A PRACTICAL METHODOLOGY FOR PROCESS NON-CONFORMANCE DETECTION

Submitted by
Sean Thompson BIS(hons)

A thesis submitted in total fulfilment

of the requirements for the degree of

Doctor of Philosophy

School of Engineering and Mathematical Sciences

Faculty of Science, Technology and Engineering

La Trobe University
Bundoora, Victoria 3086
Australia

November 2012

Acknowledgements

Foremost I would like to thank my supervisor, Torab Torabi, for providing the assistance and encouragement needed to persevere with this thesis for the five years it took to research and write it. This includes weekly catch ups, help writing publications, encouragement when the research went awry or when publications were not accepted, reading over many drafts amongst the help he provided in other ways too numerous to list here.

In the early stages of this research, I was fortunate to be coupled with a graduate student named Purva Joshi, who provided much encouragement and help with the research. She attended discussions with my supervisor and myself, helped build what became the simulation tool to test the methodology (explained in chapter 6) and co-authored the first publication that resulted from this research. Much thanks is due.

I would also like to acknowledge the assistance of La Trobe University in providing support to myself and other PhD candidates through peer research support groups, funding assistance with publications and travel expenses and through employing me as a lecturer to help keep me at university with extra time to concentrate on the research.

I would also like to thank my parents for accommodating me throughout my candidature. Without the perennial provision of lodgings and financial assistance, I simply could not have afforded to pursue a postgraduate degree.

Finally, I would like to thank the people external to my research who provided assistance in various ways such as reading papers being proofed for submission or perusing the state of the research. These people include Michael Hill, Richard Lai, Wenny Rahayu, John Grundy and Jun Han.

Abstract

Process non-conformance detection is a research area that has received relatively little attention compared to other initiatives in process improvement. There has been some off-and-on research conducted since the mid 1990s and it appeared to pick up toward the late 2000s but overall the research domain that is process non-conformance detection is yet to mature.

The main problem currently facing process non-conformance detection is the lack of practical, usable approaches. The few solutions presented to date are not practically appealing, because they have severe limitations. These limitations include being applicable to only a narrow range of processes, excessive implementation and maintenance effort and cost, inflexibility and scalability issues. There is also a lack of consideration in supporting decision making after detection, or how information relating to detected non-conformance instances should be presented to support this decision-making. These issues hinder opportunities for process improvement and diminish usability for existing proposed solutions.

Process improvement is an ongoing challenge. People will perpetually seek more efficient and effective ways of achieving their objectives. Non-conformance detection is an initiative in process improvement. A solution that can capably detect non-conformance in process enactments offers opportunities for the process to improve as a whole. However, an approach offered to detect non-conformance must be practical and appealing if it is to be used. The knowledge benefit from detecting non-conformance must significantly outweigh the effort, cost, interference and maintenance issues in implementing it.

The purpose of this thesis is to present GARDEN (Generic Application of Rule sets to DEtect Non-conformance), a practical solution for detecting non-conformance instances between a process specification and its enactments. The solution is designed to detect non-conformance while addressing the shortcomings and issues of this research field currently in the literature.

This investigation is performed in four steps. First, an analysis of existing research is presented, from general process improvement down to the specific detection of non-conformance. The issues are identified, justified as to why they are issues, and the

benefits of resolving these issues are explored. Second, a unique solution designed to address these issues and shortcomings apparent from the literature, such as keeping it widely applicable, customizable, flexible, scalable, minimizing cost and effort and minimizing time to detection is described in detail. Third, implementations of the proposed solution on a variety of different processes in different environments and settings is presented, including case study testing and objective assessment of this approach. Fourth, an evaluation of the findings is conducted, comparing the findings from GARDEN to other approaches and analysing how well GARDEN addresses the issues and shortcomings currently present in the literature.

The findings from this research show that a generic and customisable solution to process non-conformance detection is a viable and practical solution. In fact, given the wide range of process types and the innumerable environments in which they may run, a customisable approach may effectively be inevitable.

The flexible nature of GARDEN shown in the case study implementations show that it is capable of detecting non-conformance, given proper application, whilst successfully addressing the aforementioned issues concerned with this research area.

Furthermore, the implementation and evaluation of GARDEN empirically shows how the information garnered from non-conformance detection can be used to make measurable improvements to a process. The relevant data is included in this thesis to show this.

The results of the GARDEN research highlights how further research in this area can progress, how the current issues and challenges can be faced, and how the decision-making process after non-conformance detection can be supported all in a practical and useable way.

List of Publications

[1] S. Thompson, T. Torabi, P. Joshi, A Framework to Detect Deviations During Process Enactment, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

[2] S. Thompson, T. Torabi, A Process Improvement Approach to Improve Web Form Design and Usability, The 3rd Ubiquitous Web Systems and Intelligence Workshop (UWSI 2007) Co-located with DEXA 2007, Regensburg, Germany, 3-7 September 2007.

[3] S. Thompson, T. Torabi, A Formal Methodology to Detect Resource Based Non Conformance in Processes, 2nd IEEE International Conference on Digital Ecosystems and Technologies IEEE-DEST, Phitsanulok, Thailand, February, 2008.

[4] S. Thompson, T. Torabi, Towards Formalizing Resource Based Non-Conformance in Business Processes, 19th Australian Software Engineering Conference ASWEC, Perth, Australia, March 2008.

[5] S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

[6] S. Thompson, T. Torabi, An Observational Approach to Practical Process Non-Conformance Detection, Proceedings of The Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2009), London, United Kingdom 4-6 August, 2009.

[7] S. Thompson, T. Torabi, An Observational Approach to Practical Process Non-Conformance Detection, Journal of Information Technology Review, ISSN 0976-2922, 2010.

[8] S. Thompson, T. Torabi, Practical Process Non-Conformance Detection, Advances in Computational Sciences and Technology, ISSN 0973-6107, September 2012.

Table of Contents

Table of Figures

Table of Tables

Chapter 1 – Introduction

1.1 Proposal and Motivation for GARDEN

The purpose of this thesis is to present GARDEN (Generic Application of Rule-sets to DEtect Non-conformance), a process non-conformance detection methodology. This introductory chapter is intended to give an overview of what GARDEN is, and also what the reader can expect in the coming chapters regarding the purpose, design, implementation and evaluation of the GARDEN methodology, along with comparisons with other process non-conformance detection methodologies.

Non-conformance detection is an initiative in process improvement. Process improvement is a perpetual pursuit as we continually seek to improve the ways we set about conducting tasks in order to achieve a defined objective. In the context of this work, a process specification is a detailed and ordered description of the activities that must be carried out to achieve a specific outcome. We seek to accurately enhance this specification so that we can detect where the real enactment of a process does not "conform" to its original specification. Maximising the efficiency and effectiveness of how non-conformance between the enactment of a process and its specification can be detected contributes to improving the process overall, and this thesis will show how this can be achieved.

Specifically in this case, there has not been much research conducted in the field of non-conformance detection, and there are many areas to improve and address considering the current state of the literature. Through the presentation of GARDEN in this thesis, the proposal is to present a methodology to successfully detect process non-conformance in a practical and useful way, to address the shortcomings in the current state of the research, and to also provide a foundation as to the way forward. Specifically, answers are provided to questions such as what should be done once non-conformance is detected, and how can we detect and record non-conformance instances in such a way that the next step becomes easier to achieve?

Aside from actually detecting non-conformance successfully, a detailed account of the associated objectives for GARDEN in achieving this is provided in chapter 3. This thesis will show how GARDEN achieves non-conformance detection along with its associated objectives by providing:

1. a comprehensive literature review on the existing research and the shortcomings of the current state of the research;
2. a list of objectives GARDEN should achieve along with the successful detection of non-conformance between an enacted process and its associated process specification;
3. a detailed design of the proposed solution;
4. case studies depicting the solution being applied to actual processes and the results of these case studies;
5. an evaluation on how GARDEN managed to achieve the objectives set, the issues involved with its design and implementation and the results from the analysis of the data yielded from the case studies.

These are presented in the following this chapter in this thesis.

1.2 State of the Existing Research

This section will provide a brief overview of what exists in the form of process non-conformance detection solutions in this research domain. The content of this section is covered in much greater detail in the literature review in chapter 2, however here the different styles and methods that exist and the approaches that use them are referenced as a precursor to chapter 2.

1.2.1 Process Support Systems

As non-conformance is and was considered to be a negative phenomenon by researchers in this area and something that should be avoided [4] [5], solutions were conceived in an attempt to prevent non-conformance from occurring. Since the prevention of non-conformance actually conflicts with the goal of the research presented in this thesis, GARDEN does not employ any non-conformance preventative measures. Some work has also been conducted to offer a "hybrid" solution, such as the work presented by Cugola et al. [4]. This work is related to GARDEN in that it attempts to determine whether detected non-conformance is potentially dangerous to the process outcome and will "tolerate" and allow certain deviations as long as the process can still safely continue. This hybrid technology was presented in 1995 and other more observation-based methodologies are now being worked on and are covered in more detail in chapter 2.

1.2.2 Narrow Focus

The published research currently available also seems to be directed at a certain type of process. There is a lot of research devoted to software process improvement along with manufacturing processes and business processes for example, but not a lot of attempts to cross those boundaries and provide a solution that is applicable to processes in general. Specifically in the realm of process non-conformance detection, the literature clearly points to the software process [2][5] as the desired target for improvement. This has begun to change slowly, with the first publication resulting from this research aimed at keeping the approach generic in Thompson et al. [7], and also Rozinat and van der Aalst [8] claim their approach is deliberately kept generic for this reason. To increase its practical appeal and applicability, GARDEN aims to be applicable to processes generically and this is made evident in this thesis and the resulting publications from this research.

1.2.3 Methods Employed

There have been several methods employed in order to detect non-conformance instances. The most prevalent recreate reference models of the process to compare enactments against. Huo et al. [2] achieve this by "discovering" a reference model from enactment data mined from previously recorded process enactments. This is an evolutionary product of the process discovery work presented in Cook and Wolf [6]. Cîmpan and Oquendo [3] however recreate their reference model via simulation, where they simulate a flawless execution to compare enactments against to spot the flaws. Rozinat and van der Aalst [8] run recorded process event data back through Petri Nets that model the process in order to detect non-conformance.

Another approach, and the approach employed by GARDEN is to implement Boolean rules that can be applied to the enactment data. Rule-based methodologies were introduced first in an early phase of this research presented by Thompson et al. [7] and later the same year in 2007 by Kabbaj et al. [5]. Although these approaches both employ the use of rules, the application is quite different, as Kabbaj et al. attempt to detect specific instances of non-conformance from a non-conformance "list" using rules upon the enactment data, whereas GARDEN uses rules as more of a complementary mechanism to the process specification. As shown in this thesis, the rules used by GARDEN help to detect all kinds of non-conformance, not just the types that are expected and listed.

1.2.4 Cost Benefit Ratios

Although the existing approaches may succeed in actually detecting non-conformance, there is a common issue amongst them that degrades their appeal as a practical solution. This issue is the total cost of implementing and maintaining the solution. If the total cost of detecting and recording non-conformance in a process on an ongoing basis supersedes the potential benefit the process itself brings, then realistically the solution will never be used. These high costs appear in different ways in the literature depending on the approach.

The effort involved in creating a process reference model from existing enactment data can be overwhelming, as Huo et al. point out in [2]. Also, it requires the process to be repeatedly enacted before application of the approach is possible because data is needed from those initial enactments. Then, what happens when the process changes and evolves? The reference model needs to be discovered again. This is also an issue for Cîmpan and Oquendo [3]. If a flawless reference model needs to be simulated then whenever an improvement is made, presumably this reference model will need to be reformed. It's the same with any approach that uses a reference model that needs to be rebuilt when the process changes. There is a real need to minimise the maintenance costs of keeping a non-conformance detection approach relevant to an evolving process. This issue is explained further and addressed in several places in this thesis, however one key objective of GARDEN is to keep its implementation relatively low cost.

1.3 Problem Definition

This section explains the entire scope of the problem addressed by GARDEN in this thesis. The section begins with the problem domain, referring to the sort of processes targeted by GARDEN and the environments in which it can be applied. The following problem scope refers to the specific issues and challenges within the problem domain that GARDEN will attempt to address. These issues and challenges are then specifically mentioned along with why they are important and require addressing by a non-conformance detection approach. Finally, the issues and challenges that have so far not been addressed in the literature, but which GARDEN will attempt to address, are given in section 1.3.4.

1.3.1 Problem Domain

One of the key objectives of GARDEN is to make it applicable to as wide a range of processes as possible, independent of their origin, purpose, or environment. However, in order for GARDEN to provide satisfactory value to a process or the organisation in which the process runs, certain parameters must exist in order to make GARDEN worth the time and effort to implement.

GARDEN can theoretically be applied to any process at all, but like all non-conformance detection methodologies, it inherently relies upon the availability of process enactment data in order for it to operate effectively and produce useful results. Though process enactment data is critical for any non-conformance detection approach to work, its acquisition is a separate problem that is distinct from this research. The problem that non-conformance detection approaches are trying to solve is the actual detection of non-conformance, not how best to monitor, record and store process enactment data. Since each process may produce its enactment data in different ways, which affects its acquisition, we resolve this problem by assuming that if a process domain expert chooses to implement GARDEN, the process enactment data will be sufficiently structured, identifiable and accessible. This assumption is similar to Rozinat and van der Aalst's view that it is necessary to deal with this issue of process enactment data availability [8].

As will be shown in section 2.5.2 of this thesis, processes that produce structured and consistent enactment data will benefit the most from GARDEN integration. As long as process activities are distinguishable and the enactment data is consistent and reliable, GARDEN can be integrated with the best results. Processes that rely more on human interaction and subjective decision-making may be more difficult to observe for the purposes of accurate enactment data acquisition [1]. However, if it is possible for a satisfactory amount of accurate process enactment data to be logged in some way, GARDEN can be applied successfully.

As will be shown in chapters 4 and 5, GARDEN is a versatile approach to non-conformance detection and can be deployed in any number of different operating environments. In addition to process enactment data availability, an environment must also exist in which GARDEN can be deployed. Since the integration of GARDEN involves the programmatic evaluation of process enactment data using pre-defined rules, the environment where the process runs must support GARDEN interaction with the process enactment data. This includes not only the analysis of whether non-conformance has occurred, but also the logging of non-conformance instances in such a way that they can be later reported together with meaningful relevant information.

Once it has been established that GARDEN *can* be applied, an assessment should be made as to whether it is worth applying. This is the decision of the process domain expert. The domain expert will make a judgement on the perceived benefits of employing a process non-conformance detection approach versus the costs associated with its implementation. Lowering these costs will therefore increase the net benefit to having the process improved and also increase the likelihood that the domain expert will choose to implement the approach. This challenge faces all non-conformance detection approaches. Therefore lowering the cost and effort of implementation is a major objective for this research.

For any process where improvement is considered important, and if that process satisfies the constraints described in this section, then the process is an excellent candidate for implementation of the GARDEN process non-conformance detection approach.

1.3.2 Problem Scope

GARDEN is a process non-conformance detection methodology. Alongside the actual detection of non-conformance, there are issues and challenges regarding the best way to achieve this. The purpose of this section is to provide an overview of the specific issues and challenges facing process non-conformance detection that a good solution should address.

If we assume that a process specification exists to prescribe how a process should be practically enacted, then non-conformance detection is comparing actual enactments of the process against what the specification prescribes to determine if and where the enactment did not conform. This is the primary objective of both GARDEN and of the other non-conformance detection approaches referenced in this thesis.

However there are other important issues and objectives concerning non-conformance detection that other proposed solutions do not fully address. These objectives are described in detail in chapter 3. The main objectives and scope of GARDEN relating to non-conformance detection, which is covered in this section can be summarized as follows:

1. to successfully detect non-conformance (both expected and unexpected) instances between process enactments and their associated specifications;

2. to have a methodology that can be applied to as many different processes in as many different domains and settings as possible;

3. to provide flexibility and scalability in implementation and maintenance to the degree required to keep up with an evolving process;

4. to minimise "time to detection" as much as possible in any given process operating environment, so that non-conformance instances are known about as soon as, or soon after they occur;

5. to minimize the "cost" – both in resources, money and time such that the cost of implementing and maintaining the process non-conformance detection solution does not outweigh the benefits of it being implemented.

1.3.2.1 Generic Application

Since there are many different types of processes, it is advantageous for a non-conformance detection approach to be applicable to as wide a range of them as possible. It is hoped that the theories and structures presented in this research can be used to improve many different processes, so that application of the solution is not constricted to one type of process or processes restricted to a specific domain.

1.3.2.2 Flexibility and Scalability

Processes inevitably evolve and change over time as they mature and improvements are made. A good non-conformance detection solution needs to keep up with these changes and maintain its level of effectiveness. The approach should not place an undue burden on the process domain expert to maintain the solution when the process is undergoing changes.

1.3.2.3 Time to Detection

It is also important that non-conformance is detected in a timely manner. The longer the period of time between occurrence and detection of non-conformance, the more damaging the consequences may be. A good non-conformance detection approach should be capable of detecting non-conformance as quickly as possible, without wasting valuable time to set up a comparison environment or to mine data etc. The only constraints upon timely detection will be due to the operating environment the process provides for the implemented solution and the accessibility of process enactment data.

1.3.2.4 Implementation and Maintenance Costs

The ease at which a non-conformance detection approach can be applied to a process is also an important issue. Ideally, this should be as fast and as simple as possible. There should not be an undue amount of time and effort to set up or apply the approach. This also goes with maintenance. Making changes to the approach should be as simple and as quick as possible. If applying the approach is complicated, it increases the likelihood that it will be implemented incorrectly and if it takes too long, the motivation for applying it at all is decreased. An effective non-conformance detection approach should also have minimal trade-offs when it comes to its implementation and maintenance.

1.3.3 Other Important Issues

In addition to the problem scope and the objective summary described in section 1.3.2, there are also a number of other issues and challenges concerned with the research area of process non-conformance detection for which GARDEN will cater. These are not issues that are directly related to detecting non-conformance, but rather further improvements and implementation goals that are set for reasons that will be made clearer in chapter 3.

1.3.3.1 Terminology Consolidation

First, as with any relatively new research area, the language used to describe certain terms and concepts has not had a chance to become standardized. This will be touched upon in more detail in chapter 2, however researchers in this area tend to use the same terms for different things. This means that a non-conformance detection approach needs to explicitly define each term used in order to avoid confusion with other related concepts. If this can be done clearly and with a logical justification, then this will help standardise the terminology used in this research area in future.

1.3.3.2 Interference Minimisation

A characteristic that adds great value to a non-conformance detection approach is the ability to apply and utilise it without interfering with the existing process. It should be possible to apply a non-conformance detection solution to any process without hindering or changing the way people accomplish their tasks, or placing any undue burden on those already responsible for the process. The resources used to enact the process should not be diverted to implement the solution or to aid in detecting non-conformance. It is important that the process specification and its enactments remain untouched, where the approach simply detects the non-conformance instances between the two, unnoticed by those that enact the process.

1.3.3.3 Considerations Post-Detection

One facet of non-conformance detection that has been unaddressed thus far in the literature is consideration of what action to take once non-conformance has been detected. This issue is not directly related to the problem of actually detecting the non-conformance instances in the process, but some consideration needs to be given as to what to do after. The whole purpose of detecting non-conformance is so the information derived from it can be used to help improve the process. Therefore, it is important that this information be as useful and as meaningful as possible. There is an opportunity to structure the recording and logging of non-conformance so that it can aid in deciding any necessary remedial action.

1.3.3.4 The Attitude to Non-Conformance

Prior to the investigation into non-conformance in this thesis, and almost uniformly in past publications, non-conformance is universally viewed as a negative phenomenon, and something that should be prevented or avoided. This however is not always true.

It is important that we consider the possibility that people may not conform for a reason. People may figure out a better way of doing things and find improvements on their own. A non-conformance detection engine should simply detect that non-conformance has occurred and record as much data relating to the incident as possible. The consequences can then be analysed, the reason for the non-conformance analysed and if in fact it caused a benefit then perhaps this improvement can be implemented into the process specification so that the benefits can be reaped for future process enactments.

1.4 Thesis Organisation

This thesis is comprised of nine separate chapters, each with a distinct purpose in the presentation of this research.

First, this chapter being the introduction provides an overview of what the purpose of the research is. A brief rundown of what non-conformance detection is, why it is important, how it is being addressed and why addressing it in this way is beneficial and different is shown.

The second chapter focuses on the existing literature in this research domain. A great deal of related work, beginning with process improvement in general and funnelling down to non-conformance detection specifically, is covered along with all the approaches related to the goals of GARDEN. The literature is analysed and criticized with the intention of conveying that this research is worthwhile and there is availability in the state of the current research for a solution such as GARDEN.

Chapter 3 gives a specific overview of the solution along with its specific objectives. As previously stated in this chapter, these objectives do not just refer to the specific detection of non-conformance but also cover the more important and pressing issues related to it designed to maximize process improvement. Each objective is described in detail along with a justification of why it is an issue and the potential benefits that may arise from having it resolved.

Chapter 4 provides a very detailed description of the design of GARDEN and how designing the solution in this way will both succeed in detecting non-conformance whilst achieving the objective laid out in chapter 3. This chapter provides a very detailed description of every concept involved in what is considered non-conformance by GARDEN, every concept related to GARDEN and how it all fits in. The entire process from conceptual viewpoint to applying the logical constructs to a process and detecting non-conformance is explained.

Chapter 5 is similar to chapter four in that its purpose is to further describe the GARDEN solution but it does so from a different point of view. Instead, in chapter five the viewpoint of actually taking the solution described in chapter four and applying it to a process is taken into account. The step-by-step stages of process analysis, setting up of GARDEN, working out how enactment data will be retrieved, stored and analysed, and finally how non-conformance will be detected and recorded is explained in detail.

Chapter 6 is the first of two implementation case study chapters. It involves the description and high-level design of a simulation tool that was built to simulate the enactment of different processes and the application of the GARDEN non-conformance rules to the enactment data as it was acquired from the simulation. This chapter explains how the simulation tool was built, the environment it runs in and provides a detailed description of one of the processes used to test GARDEN that was run in the simulation tool.

Chapter 7 provides a real world process case study in two stages. In a progression from chapter 6 which provided a proof of concept of GARDEN and also a platform on which to thoroughly test it, chapter 7 introduces GARDEN to the real world. An online process was invented and was implemented in both an initial stage and an improvement stage for the purpose of measuring the improvements GARDEN could yield. An online implementation was chosen because it provided a suitable platform to aggregate as much data as possible and also to allow many executions of the process by literally hundreds of different people. In the first stage, the participants enacted the process and GARDEN was applied to the enactment data as normal. Then, using only the non-conformance data yielded from GARDEN, some subtle improvements were made to the process, but only improvements that were apparent from GARDEN's analysis. The process was then enacted again by a large number of different participants and the data analysed to see if the second group of people could complete the process more accurately and quickly. Chapter seven provides complete details from this experiment.

Chapter 8 is intended to be an evaluation of GARDEN. The way GARDEN was designed and whether or not it achieved its objectives stated in chapter three are evaluated. The positive and negative experiences relating to GARDEN are discussed and these are compared to other approaches aimed at process non-conformance detection. The performance of GARDEN throughout the case studies used for testing the approach is reviewed and examined. During testing, the case studies highlighted some further unexpected issues and benefits beyond the scope of the motivations covered in chapter 3. These issues and benefits are also evaluated in chapter 8 along with future opportunities for improvement that were discovered.

Chapter 9 provides the conclusion to this thesis by reviewing every chapter specified in this introductory chapter. A rundown of what was presented in this thesis is given in the form of a summary of each chapter, along with the contributions it has provided to this research area. A summary of the contributions GARDEN has made to this research area is given along with an overview of the future of GARDEN, future research in this area and how GARDEN can be improved and developed to better contribute to process improvement.

1.5 Conclusion

This chapter has provided an introduction to the process non-conformance detection solution "GARDEN". The concept and purpose of GARDEN have been described along with its objectives, why GARDEN is a different and applicable solution, and how it will contribute to this research area. A thesis plan has also been provided, which gives an indication of what content can be expected in each coming chapter and how each chapter contributes to the overall thesis. The purpose of this chapter was to provide an overview of what this research is about and how the rest of this thesis will present the details on the existing research, the existing problems to be resolved, the nature of the solution, and the testing and evaluation of its effectiveness.

The following chapter, chapter 2, presents a literature survey of the previously conducted research related to GARDEN. This chapter will investigate the body of existing research that relates to process non-conformance detection and will analyse and evaluate the existing approaches in order to provide a comprehensive view of what has been done, what the current issues are, and the direction this research field is headed.

References

[1] J.E. Cook, A.L. Wolf, Software process validation: quantitatively measuring the correspondence of a process to a model, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 8 Issue 2, ACM Press, April 1999.

[2] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[3] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[4] G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[5] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[6] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 7 Issue 3, July 1998.

[7] S. Thompson, T. Torabi, P. Joshi, A Framework to Detect Deviations During Process Enactment, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

[8] A. Rozinat, W.M.P. van der Aalst, Conformance Checking of Processes Based on Monitoring Real Behaviour, Journal of Information Systems, Volume 33, Issue 1, Elsevier Science Ltd., Oxford, UK, March 2008.

Chapter 2 – Literature Survey

2.1 Introduction

Many organizations around the world have enjoyed the benefits of process improvement since the surge in research in this area particularly from the 1980s onward [1] [2], which saw the emergence and success of business process re-engineering [3], the conception of the CMM by Watts Humphries [4] plus the founding of the W*f*MC [5] in 1993 and the subsequent successful workflow products [3] [6]. Improving the process by way of non-conformance detection, however, was not touched upon much until the mid 1990s where several techniques and methodologies were proposed using process support systems, observational models and process discovery models. It is this facet of process improvement that is the focus of this thesis.

The introduction of process improvement based concepts to other domains has started to become more prevalent with researchers introducing the more successful aspects of manufacturing processes into software processes such as the application of SPC in software from manufacturing (albeit with presently limited success) [14]. This also works the other way with the work presented in Ambriola et al. [15] evaluating and providing a collection of feedback on the results of process-centred software engineering so far (up to 1997) and its possible immersion into industrial settings.

In general, a process can be improved to either hasten the speed or improve the quality of a service. Improvement measures may be applied to processes as a whole or fragments of the process [16]. Improvement initiatives should be taken considering that the process itself is what is important, and the expense or investment in implementing an approach to improve a process should never overshadow the process itself.

Some of the literature based on non-conformance detection seems to forget this a little. While some of the related approaches technically meet the challenge of detecting non-conformance, they forget that the purpose is to improve the process and the impractical and high cost methodology they employ is unlikely to meet any of the motivational objectives for process improvement.

Though the literature in this chapter is broad in scope, the actual research area of process non-conformance detection has relatively few contributions outside the field of technology. Therefore, many of the references in this chapter are tied to technology solutions, often aimed at the software process, however the concepts and non-conformance detection solution discussed in this thesis can be applied to processes generically and independent of their respective domains.

In this chapter, the scope of the research is first defined as to what is meant by a process, process activities and non-conformance. Issues that are related to all aspects of process improvement are discussed with respect to non-conformance detection. Once the scope has been narrowed to deal specifically with the detection of process non-conformance, an analysis of directly related existing approaches using different methodologies is given. These methodologies include process support systems as well as observational systems, process model discovery, monitoring systems, and rule-based systems.

Issues and challenges relating specifically to non-conformance detection are then discussed with a focus on what needs to be resolved to progress the research in this area. Chapter 3 will follow with the specific issues taken from these, which will be addressed in detail in this thesis.

2.2 Scope

In any emerging field of research, it usually takes some time and much academic discussion before terms and concepts become standardised [65]. This is currently true of the terms and concepts used in the literature in the area of non-conformance detection. Some of the more fundamental concepts are defined in this section in relation to the body of research that is presented in this thesis.

2.2.1 Defining a "process" and its related concepts

A "process" as a general concept is defined by the Merriam-Webster dictionary as being "a series of actions or operations conducing to an end" [22]. In the business domain, Davenport [12] defines a business process as being "a set of logically related tasks formed to achieve a defined business outcome". From a software perspective, Cook and Wolf [13] provide a definition of the software process as "a set of activities applied to artefacts leading to the design, development or maintenance of a software system”.

Figure 1 depicts a very basic process example and shows how a process is comprised of activities, which are described further in section 2.2.1.3.

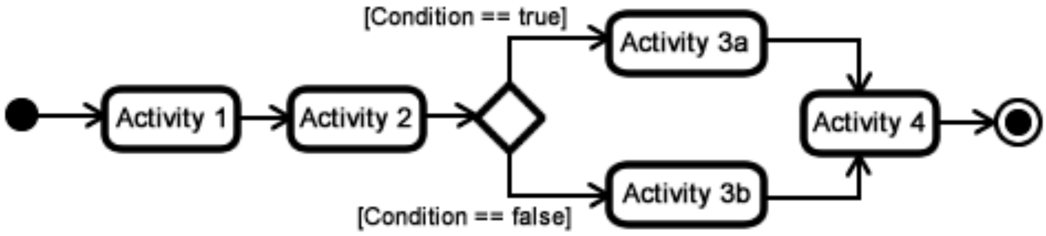


Figure 1: Basic Process Concepts

A process will be viewed in different ways depending on its domain and purpose. Business processes are typically viewed conceptually from the customer’s perspective [9][10] whereas in software, the conceptual viewpoint of a process is geared toward meeting some type of *user need* through the provision (or maintenance) of a software product [11].

There are many slightly different ways of expressing what a “process” is considered to be, depending upon the domain and viewpoint of the person defining the process. Generically however, the intrinsic concept is always the same, which is adopted in the context of the research presented in this thesis:

There exists a defined objective and we have a structured set of actions that are performed for the purpose of achieving that objective.

The actions associated with the process are also explicitly defined and can be regarded as individual elements called *activities* that make up the process. These activities will be enacted by *actors* who are burdened with the responsibility of ensuring activities are carried out according to their *specification*.

Aside from these basic notions of what constitutes a process, a process may also include machines, methods, rules, organizational structures, sub-procedures and computerized tools, as set in the description of a process in Florac and Carleton [21] and Blyth [16]. Resources such as machines, computerized tools or other exhaustive and non-exhaustive entities may also be used when enacting the process to achieve the process objective.

2.2.1.1 Process Specification

A “specification” as defined by the Merriam-Webster dictionary is “a detailed precise presentation of something or of a plan or proposal for something” [23]. Typically, a process will have a structured specification that acts as a set of instructions as to how it should be performed. The rigidity or flexibility of the specification will vary depending on the process.

Spoken from a software engineering perspective, a good and/or formal process specification is important to aid in process understanding, communication, analysis, execution guidance, visibility, coordination and improvement support [24] [25]. A poor or informal process specification forces actors to make their own presumptions as to how to carry out the process [24], which can lead to undesirable consequences.

The term “process specification” may be interchangeable with the term “process definition” in some of the literature, however “specification” is used in this context throughout this thesis.

2.2.1.2 Process Enactment

A process “enactment” refers to a single instance where a process specification has been physically carried out with the purpose of achieving its associated objective. Depending on the individual situation concerning the process in question, the inputs and outputs will be different from enactment to enactment, as Lantzy points out in [8] in relation to the software process. Therefore, multiple enactments of the same process specification will almost always have a different set of event data relating to each enactment.

One of the key challenges is the accurate observation and recording of process enactment data so that it may be used in process improvement endeavours. The comparison of enactment data to its associated process specification to detect non-conformance is the primary objective of this research.

2.2.1.3 Process Activities

Activities are the separate individual tasks that when performed by *actors* [17] that make up the process. Researchers in this domain generally accept a similar notion of the term "activity". Occasionally, the term "task" and the term "activity" may be differentiated [50], however for the most part in the literature "activities" can be used interchangeably with terms such as "steps" or "tasks".

Breaking the process into a set of related activities makes the process more modular and easy to describe and understand. It also aids in determining where (and why) non-conformance occurred in a process if we can associate the instance of non-conformance with a specific process activity.

There should also be some specification of how activities should be sequenced within the process. Enacting the sequencing of activities may also not necessarily be specified in a linear fashion, but also concurrently, simultaneously, overlapping or in parallel [19] [20]. Enacting the activity set in a way that does not conform to the sequencing specification also constitutes non-conformance. Sequence based non-conformance is referred to in this chapter as "deviation" and is regarded as distinct from other non-conformance for reasons discussed further in section 2.2.2 of this chapter.

When specifying process activities, it is prudent to try to keep each activity as modular, concise, explicit and simple as possible. For some informally defined processes, activities can be quite broad in scope and even encompass sub-processes within themselves. For example, Sommerville [27] provides a rather broad definition of what encompasses software process activities as specification, development, validation and evolution. In the context of this research, a process activity is a much more modular and preferably singular task.

2.2.1.4 Process Actors

An actor is an entity whose responsibility is the enactment of one or more activities within a process. An actor need not necessarily be human, but can also be an external entity such as an information system or a production machine [28]. There are trade-offs between human and non-human actors.

If the actor is a person, the competency of the actor to perform the activity required of them will be a vital influence on the outcome of the activity. The classifying of skills held by actors and their competency and assignment to complete different tasks traditionally involve human judgment [26] and the issues pertaining to this are not covered in this thesis.

If the actor is not human, different issues may arise, such as the likelihood of fault or failure or even the fact that if there is no human actor associated with a process activity then it can be difficult to attribute accountability to an actual person if there are none involved in the activity's enactment.

2.2.2 What is "Non-Conformance"?

Non-conformance in the context of this research refers to an instance where a process has been enacted in a way that is inconsistent with its specification. Detection of non-conformance is therefore achieved by comparing process enactment data to the process specification with the purpose of finding instances where the enactment has not conformed to the specification.

This definition of "non-conformance" is interchangeable with the term "deviation" throughout much of the literature in this area. Some of the publications do not explicitly define these terms at all such as Huo et al. [19], but the publications that do invariably give a definition of "deviation" that is synonymous with our own. Cîmpan et al. [30] describe a deviation as non-conformance between a performed process and its instantiated process model, whereas Reese and Leveson [31] say a deviation "is the difference between the actual value of a system variable and the expected (or 'correct') value".

Also apparent in the literature is the use of the term "inconsistency". Kabbaj et al. [29] refer to both "deviations" and "inconsistencies" but not the term "non-conformance". In this case, the authors similarly define a "deviation" as being an "action performed that is not described in the predefined process or that violates some of the constraints expressed in the process". However, "inconsistency" is defined as being a state that has occurred as a result of the deviation occurring. It is important to note here that the absence of a required action or activity can also constitute non-conformance, a distinction that is not suggested in the definition of "deviation" provided by Kabbaj et al.

The term “inconsistency” is incorporated into our own concept of “non-conformance” but in quite a different context. Some work has been conducted in the formalisation of these terms and concepts, including mathematical modelling and how this may be applied [18]. However, the work conducted by Cugola et al. [32] on the formalisation of deviations and inconsistencies provides the basis for the definition of terms adopted in this thesis.

Cugola et al. [32] describe the term “inconsistency” as relating to the state itself, where the “state” is the state of the process at the time the non-conformance occurred. “Deviations” are differentiated as a concept relating to *transition* between states in a process. In keeping with these concepts, the definition of non-conformance in the scope of this thesis is extended to being one of either an “inconsistency” type or “deviation” type.

2.2.2.1 Inconsistencies

Inconsistencies are non-conformance instances that relate to the attributes associated with the process or specifically with one of its specified activities.

2.2.2.2 Deviations

Deviations are non-conformance instances that relate to the transition from one process activity to the next, where that transition is considered illegal according to the process specification.

The definition of non-conformance is depicted graphically in Figure 2.

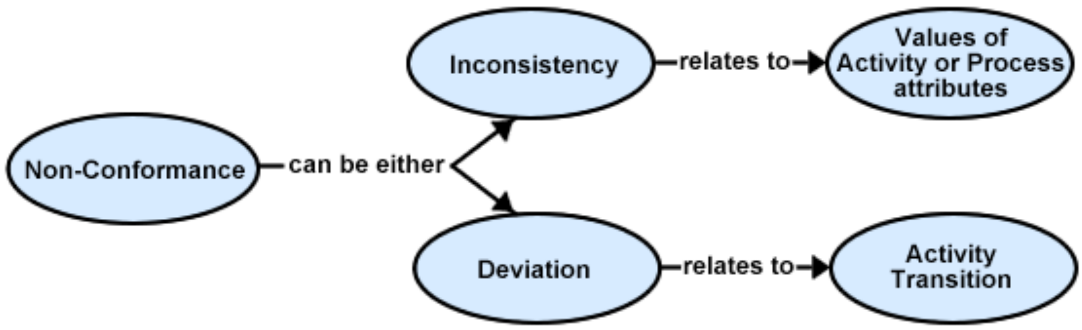


Figure 2: Non-Conformance Definition

2.3 Process Improvement Issues

The detection of non-conformance in processes is quintessentially a process improvement initiative. One of the research aims in this thesis is also to keep the approach generic and applicable to different processes generically. Since the nature and scope of processes can vary quite significantly from one process to the next, there are certain considerations we should first address that will affect the implementation of any process improvement approach, including that of non-conformance detection. We cover the issues pertaining to these considerations and their potential effect on the objective in this section.

These considerations are:

- The kind of process are we dealing with i.e. manufacturing, business, software process
- The purpose of the process
- The importance of the process, and the limit on cost and effort required to implement a process improvement solution
- How tightly or loosely the process is defined
- The current state of the process
- The extent to which the process carried out manually and/or automatically
- Risks of interfering with the process
- The point where we stop improving the process

2.3.1 Process type

The type of process being dealt with will provide us with some preliminary indication of its nature. Manufacturing processes tend to be repetitive in nature, the same process enacted over and over to produce a product. Therefore many process improvement initiatives in manufacturing are simulation based, which focus on how to speed up production times, evaluate a production line or prevent product defects [36]. The focus on business processes tends to be on how to deliver the best service to the customer [9][10]. If it is a software process we are dealing with, we can reasonably assume that it will be mainly people and action based, bereft of things such as raw materials that may be important in other domains like manufacturing. Also, in certain software processes, we may be building a complicated system only the one time, so the actual process may be enacted just the once, which will have a bearing on any improvement initiative we may wish to implement.

2.3.2 Process purpose

If we understand the goal that the process was implemented to achieve, we will attain a greater understanding of the process itself. Processes may be implemented for the following reasons:

- to produce a product;
- to deliver a service;
- to test or evaluate something;
- to increase the performance of something;
- to minimize faults.

Furthermore, is this purpose internal/organizational or external/customer based? Gaining an initial understanding of the process and the hierarchy of relative priorities pertaining to it is also paramount as a basis for the feasibility of attempting to improve it.

2.3.3 Process importance

This follows on from the ascertained purpose of the process, because then a greater understanding of the process objective and the importance of that objective is gained. Determining how critical the process we seek to improve is to the organization that has employed it also relates to feasibility.

Once we have an understanding of the importance of the process, we can begin to estimate what upper limits there may be on the cost of implementing an improvement approach. Sometimes, we find that the amount of work involved in setting up a non-conformance detection system may not be in proportion to the cost and effort of performing the process itself. A good example of an approach where this is the case is shown in Cugola et al. [38].

When a reasonable estimate of the time, effort and financial cost of a process improvement approach is available, this should be weighed against a reasonable estimate of how much of a positive difference the improvement would make, along with a comparison of the cost, effort and time involved in performing the process itself.

2.3.4 Flexibility of the process specification

How loosely or tightly a process specification is structured will greatly affect how it is performed, and therefore affect the implementation of any improvement approach. From a non-conformance detection perspective, the preference would be to have the specification as detailed as possible, but this is not always the case.

Sometimes, a process model may intentionally be implemented in an indistinct manner such as the approach proposed by Bogia and Kaplan in [39]. In this approach, the process model is deliberately vague, choosing only to define obligations, which must be met by guiding actors toward a goal without strictly enforcing any particular set of actions. Other models such as workflow systems aimed at the automation and support of business processes [40] enforce a very specific set of actions en route to a very specific goal.

The two extremes have different trade-offs. Vague specifications will afford actors a great deal of flexibility and creativity to deliver innovative solutions but at the risk of having them make costly errors and mistakes, or simply running a very inefficient course of action.

Process models that are too generically structured have also been criticised for their inability to convey specific and important detail about the process. These can be things like triggering and terminating conditions of activities, inputs and outputs of an activity with the sources and destinations of the data, how parallel and sequential process steps are supported, communication amongst humans, data flow between activities and steps required to resolve customer-reported software problems [37].

A rigid and inflexible model will mitigate this risk, but it will also severely dampen any prospect for improvement, or for the actors to discover better ways of achieving goals as well as making the process quite tedious to work on [41]. Clearly, a balance must be struck commensurate to the situation.

2.3.4.1 Size of organisation

Dybå [7] argues that the likelihood of a process specification being loosely or tightly structured is greatly influenced by the size of the organisation. The smaller an organization is, the more likely it will be geared toward a more flexible, practice-oriented way of process enactment whereas larger organisations will prefer more formally specified process models. This discrepancy is to be expected, as developing a formal model for ongoing and complex processes can be costly, difficult and error prone [33]. Also, it is usually important in large organisations in particular that process and practice remain consistent. Formal definition can help to facilitate this.

2.3.5 Current state of the process

The present maturity of the process in question will beg further sub-questions pertaining to its current state:

2.3.5.1 How old is the process?

The degree of familiarity of the process stakeholders with the process will vary depending on how long the process has been performed in its current form. If the length of time is significant, some of the stakeholders may be averse to any changes, even if those changes result in increased efficiency. Older processes may also have undergone various improvements and changes over time that may need to be considered before implementing anything new.

Newer processes may be underperforming because the actors are still becoming accustomed to it, and conversely there may be more of an opportunity to improve a newer and relatively untested process for a greater gain than an older one.

2.3.5.2 What is the "size" of the process?

The size and complexity of the process can be quantitatively measured. Considering aspects of the process such as the number of people involved and the amount of time the process is expected to take, we can gauge an idea of how significant the undertaking might be. Larger and more complex processes will affect more people and make the gathering of required information more difficult [45]. This can therefore make the proposed approach more costly and compound the amount of risk associated with its implementation. Also, there has been research to suggest that larger process models tend to have more formal flaws than smaller process models [62].

2.3.5.3 What kind of support does the process presently receive?

The amount of attention and support the process is currently receiving in its environment is also of concern. There may be a reason as to why a particular process receives more or less support. Consideration must be taken as to whether the implementation of our process improvement initiative is going to require more support and organizational resources (and therefore increased cost), and take support or resources away from some other part of the organization, or whether the amount of support required for the process in question will actually be reduced – freeing up resources to be distributed over other parts of the organization.

2.3.6 How the process is carried out

The extent to which humans are involved in the carrying out of activities will have a very significant impact on the process and any improvement approach being considered. A human is much more likely to cause non-conformance than an automated actor and human behaviour is inherently much more difficult and expensive to monitor and record [44] (especially for the purposes of non-conformance detection) than an external system. Curtis et al. [42] claim that process modelling in computer science is distinguished from other areas due to the predominant human involvement in enactments compared to machines.

Cugola et al. [32] describe processes that they consider to have a significant level of human involvement to be *human centred systems* where the influence of the people involved in a process is especially high. Software processes are considered human centred, requiring some degree of flexibility to facilitate creativity as well as the management of changing requirements, technologies or work environment [43]. In human centred systems, many tasks may be performed away from the computer making it a) difficult and expensive to monitor and record and b) impossible to guarantee conformity from the process specification to its enactment [44].

2.3.7 Risks of interfering with the process

Whenever a change is proposed for a process, there is an element of risk associated with that change, as change implies interference with the process itself. Along with the obvious care that should be taken to avoid compromising the effectiveness of the process being changed, consideration must be taken that other processes that may be coupled with the target process are not adversely affected.

There also exists the possibility that other improvement initiatives may be in effect at the time. Care should also be taken that the implementation of a new process improvement initiative does not interfere with the operations of those that have already been employed.

Research has been presented in the literature on process dependencies, however most of this relates to task dependency within a process itself, or issues regarding inter-process dependencies such as in Grossmann et al. [48]. In order to minimize risk, a key objective for the research presented in this thesis was to avoid interfering with the process or its environment as much as possible.

2.3.8 Concluding the process improvement initiative

There will eventually come a point where the cost of further improving a process does not justify the benefit gained from doing so. The issues examined in section 2.3.3 of this chapter regarding the relative importance of the process against the benefits from improvement and the cost, effort and time investment to implement those benefits must be constantly re-examined as the improving of the process evolves.

Assuming that the implementation of a non-conformance detection mechanism has been decided as feasible for a particular process, in the beginning we can expect there to be a high relative number of non-conformance instances detected. As the process specification is subsequently updated and improved, less non-conformance instances should occur over time. Eventually, it will become prudent to suspend the mechanism until such time as the process evolves from its current form, or some other change warrants its re-implementation.

A parallel can be drawn between a non-conformance type of process improvement initiative and defect removal. “Six-Sigma” is a process improvement strategy aimed at identifying and removing defects in a business process, originally developed by Motorola [47]. In a generic, unspecified process, Six-Sigma asserts that 3.4 defects for every 1,000,000 opportunities where an opportunity is a chance for a defect to occur [9][46] – is the point at which improving a process will never be cost effective [46].

Of course, this is an upper boundary that is intended to govern all processes. The actual point at which to cease the improvement of a specific process will be dependent upon the situation and process itself and is a decision to be evaluated as the process improvement initiative matures with the process.

2.4 Analysis of Related Approaches

Non-conformance detection is a research area that has evolved from process support systems and the demand for increased flexibility, through to observational systems with a view to maximising process improvement. The purpose of this section is to analyse the approaches directly related to process non-conformance detection. These approaches vary in the technologies they use, their chosen methodologies, the reasoning they take and the goals they set for themselves. These non-conformance detection approaches are presented in chronological order, and the benefits along with the issues associated with each approach are discussed. The issues associated with these existing approaches, along with the issues presented in the sections following concerning non-conformance detection make up the goals set for this thesis, which are discussed further in chapter 3.

2.4.1 Scope of the Analysis of Related Approaches

The aim of this section is to present and analyse the range of different existing techniques and approaches directly aimed at the detection of process non-conformance.

The related work cited in this section and the approaches with which GARDEN are compared are solutions directly related to the detection of non-conformance between a process enactment and its specification. These approaches are chosen because they represent a variety of different methodologies and techniques to approach resolving the problem of non-conformance detection. These different techniques have their own strengths and weaknesses and are therefore useful at highlighting the different ways in which this research area has been tackled previously and which techniques show success and which do not.

Some of the research analysed in this section has branched off and evolved as the research was pursued. In some cases, such as with Rozinat and van der Aalst [71], a non-conformance detection technique evolved from research that was not initially based on trying to detect process non-conformance. However the instances of the approaches presented here are the incarnations that best represent the technique employed in order to detect process non-conformance.

2.4.2 LATIN/SENTINEL – Support Based

The early stages of this research domain began to be explored with the introduction of more flexible process models and process modelling languages. An early example of this was the approach proposed by Cugola et al. in 1995 [38]. In this approach, the authors presented the process modelling language LATIN, which ran in a prototype environment named SENTINEL.

Cugola et al. argued that there was a clear distinction between a process specification and the actions of the humans who carry it out, and that the enactment and the specification will inevitably diverge. The proposition was still a support-based system but one which was becoming a lot more flexible, and effort was concentrated on reconciling the enactment and the specification when these process "divergences" occurred.

The intention of this support system was not explicitly to detect non-conformance, but to accommodate flexibility, evolution of the process and change without forcing the enactment to follow the prescribed model. When the enactment diverges from the process specification, then any inconsistencies that arise from the divergence should be handled by this approach.

The proposed approach implements state machines in which the state of the process could be described by variables relating to it. Transition between states is defined by preconditions and safe-states defined by invariant assertions. The authors argue that if the preconditions fail for a state transition, a *tolerable deviation* is noted and allowed as long as the invariant assertions that define the safe state still hold true. If an unsafe state is entered the process is blocked and no further progress can continue. They also argue that an illegal transition may return *polluted* data, and that the onus of recognizing this is on the user. Since all event data is recorded in a knowledge base, some of the data will be polluted and have to be removed at the end of the process. A set of pollution rules is defined for all possible cases where events may potentially be incorrect – meaning inconsistent with the process description.

Although this approach is leaning toward the provision of flexibility, it is still in the process support category and has a high degree of interference with the way the process is enacted. This interference is not necessarily a bad thing (especially if it results in an improvement), but it does change the process and the way it is enacted, and therefore changes the way further process improvement initiatives are to be considered. Non-conformance is always less likely to be detected if it is prevented due to interference in the enacting of the process.

As this approach has a recording of data and diversion handling mechanism associated with it, then "diversions" can be picked up on the fly to an extent. If an unsafe state is entered according to the approach, then the process will be prevented from continuing, so this constitutes on-the-fly detection. Tolerated diversions however will not be detected until the polluted data are analysed at the conclusion of the process enactment.

There also seems to be an inordinate amount of work required in the implementation of this kind of approach. This includes not only the effort involved in setting up and monitoring, but after the process conclusion also the identification and removal of all the polluted data. It is important to reduce this burden because a high cost vs. benefit will make such an approach unattractive in a practical or commercial setting.

In this paper, the word "diversion" is taken to have a similar meaning to "non-conformance". It is implied that a diversion is an instance where a human actor acts in a way that is not consistent with the process specification. The terminology in this paper is therefore also not consistent with most of the literature relating to non-conformance detection, but this is likely because this was written at a time when the research area was only starting to develop and the approach is not explicitly aimed at the detection of non-conformance.

2.4.3 OMEGA – Fuzzy Logic

The approach in 2000 presented by Cîmpan and Oquendo was intended specifically for the detection of non-conformance or "deviations" being the term the authors use [30]. The body of work presented was intended for a particular use in software processes. They named this approach "OMEGA", an acronym for Online Monitoring Environment: General and Adaptable.

The idea was to compare an enacted process to its associated specification (or "model") using fuzzy logic. Certain metrics from the process are selected and via the use of fuzzy sets theory, a process enactment may be compared to its expected behaviour. The level of conformance this comparison yields is a measure of similarity between actual and expected behaviour.

One possible benefit not mentioned by the authors of this paper is the heightened ability to detect *unanticipated* non-conformance, which is an issue hindering other approaches in this domain. Since a comparison is made as a measure of similarity between a generated reference model from a process specification and its enactments, then the different possible non-conformance types do not need to be anticipated and catered for as they do in approaches like Kabbaj et al. [29] or Zazworka et al. [69].

The "expected behaviour" of the process is modelled by simulating what is considered to be an ideal, flawless enactment of it. The authors here acknowledge that such an enactment cannot really be "expected", as an actual perfect enactment is too unlikely to occur in practice.

Cîmpan and Oquendo raise the concept of *tolerable deviations* in this paper. From the acknowledgement that their modelled expected behaviour was in fact extremely unlikely to ever occur, a range of tolerated behaviour must be allowed in a scope that does not jeopardize the process outcome. It is possible that some minor and predictable discrepancies may have occured during process enactment that we may wish to simply concede, especially if these discrepancies do not compromise the process objective. This became the trigger that inspired the notion of process "exceptions" used in the proposed approach for this thesis, which is explained later in chapter 4, section 4.3.1.6.

Although not explicitly mentioned as an issue of importance in this paper, severity in this approach can be theoretically deduced by measuring the conformity in the fuzzy sets theory described. The authors talk about tolerance levels in which the values of tolerated deviations may lie. Since conformance is a measure of similarity in this approach, it allows for severity of non-conformance instances to be determined. The real value of severity however, will be a measure of how far the enactment has strayed and does not reflect values such as the relative importance of the process or the importance of the part of the process where the non-conformance occurred.

Since this is an earlier approach, the terms used are different in this paper. The authors refer to "deviations" and "divergence" instead of "non-conformance". Despite this expected discrepancy, it is not clear how much work is required to implement this kind of strategy. In the act of creating the model of expected behaviour through simulating flawless execution, it is unclear how the approach might deal with non-linear execution paths. If a process can legally substitute one activity for another, or have different activities being enacted concurrently, it is unclear how this kind of approach might handle that and also what additional effort and cost may be required to support it. It is also not clear what the level of interference of this approach has on the process enactment.

Another point of interest with this approach is that it seems the use of the OMEGA environment supports on-the-fly detection, which would enhance its usefulness.

2.4.4 Process Enactment Data Mining

Huo et al. [19] introduce a retrospective approach aimed at the software process which is an evolution of the data discovery methodology presented by Cook and Wolf in [13]. This approach involves the performance of process discovery, defined by Cook and Wolf [13] as where a process is enacted and monitored, then the data is recorded and mined and the process model is derived from mining the data. As with the previously highlighted inconsistent terminology in publications in this research area, this publication from 2006 also refers to non-conformance as "deviations".

The process begins as most processes do, with a pre-defined process model or specification that is enacted. The process enactment data is then collected and prepared for mining. When the process data is mined, the goal is to diagnose the process enactment pattern and effectively discover the process model based on the process data. This discovered model may then be compared with the predefined model and non-conformance instances detected from the parts of the enacted model that do not conform to the defined model.

The first positive to come from this paper is the authors' acknowledgement of the burden of implementing the approach. They define two distinct goals in their approach: the first is to minimize the burden of data collection when collecting process instance data. The second is to address a problem they say is associated with process discovery where the discovered model is on a much lower level than the defined model and therefore, comparing the two proves a very difficult task.

Whilst process discovery enables us to see how the process is "actually" enacted, the problem, as the authors acknowledge, is the amount of work involved in gathering the data, mining it and then comparing it to gauge non-conformance. The necessity to collect all process data and then mine it before occurrences of non-conformance may be detected makes detection on-the-fly impossible and inhibits the opportunity for timely detection. Detecting non-conformance is really only half the battle in this kind of research. It needs to be achieved in an efficient manner that can provide some practical value to a process.

Another issue with this approach is the way the predefined model is initially instantiated. In general, it is prudent to use real process data from enactments where possible in order to ensure the highest quality of the defined model. It is not clear in this paper if this is the case, though the authors do state that the comparisons drawn between the predefined model and the enactments are used to improve the predefined model. If the predefined model is not instantiated in an appropriate fashion, the quality of the predefined model is uncertain, and therefore so is the real significance of any non-conformance instances that are detected by drawing a comparison to it.

2.4.5 Non-Conformance Detection and Tolerance

The 2007 approach by Kabbaj et al. [29], also presented in the context of the software process, is a solution to support the evolution of the process while being tolerant of non-conformance. The approach has some similar characteristics to the others listed here, but seeks to evolve past simple detection with some useful mechanisms to grade non-conformance instance depending on their relative severity, though the term "severity" is not used.

The objective of this research is to detect and manage process non-conformance, referred to here also as "deviations". Non-conformance is inevitable, the authors claim, and also is the need for process models to constantly change, adapt and evolve. This approach is designed to complement such process evolution.

A number of different systems and models are used in this approach. The authors describe a "Monitoring System" tasked with the monitoring and recording of process enactment data. Two other models are also directly involved in the non-conformance detection mechanism. They are the "Observed Process Model", which is derived from data recorded by the monitoring system and the "Enacting Process Model", which like the OMEGA [30] approach, represents an ideal process enactment.

As the monitoring system records process enactment data, this data is used to form the observed process model and every time some new data is recorded, the observed model is compared to the enacting model to detect non-conformance. It seems as though the authors have compiled a list of possibilities for non-conformance and comparisons are then checked with the list, of which the authors claim to have compiled around thirty different types of possible non-conformance.

Once non-conformance is detected, it can be managed using one of three strategies, which were first introduced in Cugola's work on the SENTINEL approach to process enactment evolution [67]. These three strategies, paraphrased using terminology consistent with the research presented so far are: a) do nothing; b) modify the process specification to tolerate the corresponding non-conformance type; or c) create an exception to handle that type of non-conformance without actually altering the process specification.

This enhancement is positive because the authors are thinking about what should be done after non-conformance has been detected, and what consequences it may have for the process. In order to decide which of the three measures to take with the instance of non-conformance that has been detected, they subject it to "deviation tolerance rules" from which a "deviation tolerance index" is derived. This deviation tolerance index is a value placed upon the non-conformance instance ranging from 0 to 1, 0 being there is no room to tolerate non-conformance at all and 1 being that non-conformance can be tolerated completely.

The work presented by Kabbaj et al. in this paper seems as though it has yet to mature. The paper is short and the authors state they are still working on a prototype implementation and a case study to evaluate their approach. The requirements of the monitoring system along with the other constructs that need to be in place make this approach seem difficult to apply to other processes especially processes in different domains. Since there are a few systems and models involved in this approach, it is unclear how much effort is required in its application to a process. As with many other approaches in this area, some of the terminology in this work is still inconsistent with the rest of the literature. For example, the authors define an inconsistency as a state resulting from deviations/non-conformance occurring rather than being distinct from "deviations".

It is also worth noting that the authors state, "in an ideal world, deviations never occur". This implies that they view all instances of non-conformance in a negative light and have not considered the possibility that non-conformance may, in fact, have a positive effect on the process. The preconception that non-conformance is always bad is not unique to this body of work. One of the challenges facing this research domain is to embrace the possibility that non-conformance could, in fact, be positive and that by detecting non-conformance and understanding it, we are able to improve processes.

2.4.6 Conformance Checker in ProM – Petri Nets

In 2008, Rozinat and van der Aalst presented their own approach to process conformance checking using a mathematical model [71]. The authors had previously published similar work on "choreography conformance checking", which was based on checking the conformance behaviour of services using the BPEL and Petri Net technologies, alongside the Process Mining Framework (ProM) [72]. This extension however is more closely related to specific process non-conformance detection, a more mature version of the research, and more directly comparable to the research presented in this thesis.

Rozinat and van der Aalst present a technique aimed at detecting the level of conformity between a process reference model and the event data from a corresponding enactment of the process. This technique involves first modelling the process using Petri Nets. In order to resolve the issue of process enactment data acquisition, the authors state the assumption that the process will automatically record its enactment data to an event log. As long as the event log contains a) an identifier of the process enactment for each entry; b) distinctly distinguishable process activities; and c) being completely ordered, then this approach can be applied.

Instances of non-conformance are detected using Rozinat and van der Aalst's approach by running the event data back through the process model (in the form of a Petri Net) using a tool they have developed, named the "Conformance Checker", which was built on the Process Mining Framework (ProM). The authors have made an effort to keep their approach generic and useable by a wide range of processes and state this intention explicitly.

Rozinat and van der Aalst describe conformance as a function of two dimensions: "fitness" and "appropriateness". According to the authors, *fitness* represents how well a process enactment has conformed to its reference model, whereas *appropriateness* represents how well the reference model describes what process enactments should be doing. The authors state that "fitness" is indeed the dominant requirement for conformance, but that high fitness does not actually imply conformance as the process model appropriateness could be of insufficient quality. It is interesting that the authors argue that a strong process model is required for conformance, as Zazworka et al. [73] argue that "a successful approach should be able to handle different definitions of process, ranging from informal ones… to formal ones expressed through a process modelling language".

It appears that as with other publications in this research area, the terms used in this paper have different meanings in others. The authors mention "inconsistencies" and "deviations" at various points, however these terms are not explicitly defined. The authors also refer frequently to the term "conformance" as opposed to "non-conformance", in their evaluation of "fitness" and "appropriateness" levels of certain parts of the process event data. This is different to most other approaches, which usually present the context of their research on a "non-conformance" basis. This is not a criticism, rather just an observation on how different researchers view the concept differently. The idea is not black-and-white as in other approaches where the enactment data either conforms or not but rather is a grey area measuring how well the data fits the model and how well the model describes what the data should look like.

This paper appears to deal mainly with what would be considered as violations to activity sequencing within the process. The examples given and illustrated using this technique show legal transition through the petri-net representation of the process, how the sequence of these transitions took place in the event log, and finally how these two representations marry-up to detect whether non-conformance has occurred.

Although this approach seems to be effective at detecting process activity sequencing violations and representing them well, there are other possible non-conformities that may occur that are difficult to realise using a model like the one described. These could include examples such as the wrong actor enacting an activity, time duration violations, or misallocation of process resources. It is unclear how this approach could detect and/or represent non-conformance types that are unrelated to sequencing.

Similar to Huo et al. [19] and Cimpan and Oquendo [30], this approach uses a process reference model to compare with the enactment data. An issue common with process model approaches is the blur between whether the reference model is used as a process specification, or as a means to detect non-conformance. Combining non-conformance detection attributes to a reference model intended as a process specification risks convoluting the model due to the coupling of the models objectives. There should already be a process specification in place. Creating a new reference model should be concerned solely with non-conformance detection, not with providing an additional process specification.

Arguably, the inclusion of more information and constraints into a reference model increases its "appropriateness", to use a term coined by Rozinat and van der Aalst. There are, however, some drawbacks. The first being that there are all sorts of constraints and boundaries we may wish to test for non-conformance that do not easily fit into a reference model, such as resource misallocation, actor types or even time constraints. Second, there may be issues with timely detection, although with this particular approach, it seems theoretically possible to detect non-conformance as soon as the event data becomes available, though the authors do not claim this. Third, whenever the process changes due to evolution, the reference model will likely need to be rebuilt from scratch, which can cause cost and maintenance concerns.

Finally, as with the other approaches, we need a good basis for non-conformance instance logging that provides as much information to the analyst as possible so that post-detection decisions may be supported. Rozinat and van der Aalst's model clearly shows a process analyst where non-conformance has occurred. The next step is to provide concise, informative data on the non-conformance that can aid decision making on what to do when non-conformance has occurred, depending on factors such as past instances, severity, organizational and process importance, and identifying other affected areas.

2.4.7 CodeVizard

The "CodeVizard" tool is an approach presented by Zazworka et al. in October 2009 [69]. This paper focuses on the software process and is explicitly aimed at the detection of "non-conformance" in software processes. The authors define process conformance as a measure of how much an "executed" process complies with a "defined" process. The terminology used in this paper is somewhat consistent with previously coined terminology. They mention that processes are made up of activities (however, there is no mention of actors). The terms "process execution" and "process definition" in this paper are what would be referred to as "process enactment" and "process specification" respectively throughout this thesis.

In order to detect non-conformance, the authors define a "partial set of non-conformities", which they explain is a non-exhaustive list of non-conformance types that they are looking for in the enactment data of a specific process. This list is comprised of non-conformities that the authors have identified as worth checking for since they are a) possible; and b) pose a level of risk to the process objective as judged by a preliminary risk analysis when the non-conformance type is identified.

The approach also does not make use of a process specification. The authors describe a list of non-conformities based on manual interaction with process stakeholders and make use of a "process conformance template", which is filled out manually. The tool described in this paper, named "CodeVizard", is then used to check process enactment data that has become available through some form of process monitoring or observation.

CodeVizard was specifically developed to detect non-conformance in software-based processes. The application is capable of working with tools such as CVS or Subversion in order to analyse data that is likely to be specifically applicable to software engineering in order to detect non-conformance.

It appears as though this research is still in the early-to-mid stages of development, at least at the time of publication, because much of the work is still manual. The CodeVizard tool is capable of assisting in analysing the risk of non-conformities, however this still requires subjective judgment. Compiling a list of non-conformities to check for is manual work and when this occurs, the authors suggest an interview process with the relevant process stakeholders in order to uncover the cause of non-conformance and how to use this information to improve the process. This is not necessarily a bad idea, but if enough quality data is collected as part of the detection process, then information like cause and improvement areas should become self-evident.

The authors state in this paper that timely detection is of critical practical importance when detecting non-conformance, however the detection mentioned in the paper is made overnight or on weekends. In the 'Future Work' section of the paper, the authors state that they are working on making the non-conformance detections more quickly in order to make it more practical to typical work practices, suggesting this is a work-in-progress.

Additionally, this approach is likely to face the same shortcomings as the approach by Kabbaj et al. [29] by using predefined lists of non-conformities for which to check. This places the burden squarely on the person responsible for compiling this list to foresee all types of non-conformance and to conceive ways in which to command the CodeVizard tool to detect them.

2.4.8 Summary of Existing Directly Related Approaches

Feature	SENTINEL 1995	OMEGA 2000	Huo et al. 2006	Kabbaj et al. 2007	Conformance Checker 2008	CodeVizard 2009
Adequate Enactment Flexibility	✗	✓	✓	✓	✓	✓
Observational	✗	✓	✓	✓	✓	✓
Clear and easy to produce reference model	N/A	✓	✓	✗	✓	✗
Distinction between levels of severity	✓	✓	✗	✓	✓	✗
Supports non-linear execution paths	N/A	✗	✓	✓	✓	✓
Provides a practically useful measure of severity	N/A	✗	✗	✗	✗	✗
Quick and easy to integrate and modify	✗	✗	✗	✓	✓	✓
Acknowledges "positive" non-conformance	✗	✗	✗	✗	✗	✗
Applicable to broad range of processes	✗	✗	✓	✓	✓	✗
Can detect a wide range of non-conformities	✗	✓	✓	✗	✗	✗
Support for post-detection decision making	✗	✗	✗	✓	✗	✗

Table 1: Summary of Existing Related Approaches

2.5 Support vs. Observation based approaches

2.5.1 Approach Type Overviews

Non-conformance detection methodologies have some inherent commonality to non-conformance prevention and defect removal methodologies. In the literature, these related approaches tend to fall into two distinct categories: *support* based and *observation* based. There are trade-offs between the two types, which are discussed in this section and summarized in Table 2.

The purpose of a process support system is to ensure that the process enactment conforms to the schema (or specification) [51]. Due to the nature of support systems, they tend to have a focus on controlling a process and have a direct impact on how actors can go about performing their activities.

Support systems, such as workflow products, have been criticized (especially early on) for being too rigid in forcing the users to adhere to a specified process. These systems had insufficient mechanisms for their specifications to handle exceptions [56] along with focusing too much on control [52] [53] rather than providing a flexible tool to users.

People do not like to be stringently controlled and will become malcontent with a system that inhibits their flexibility. Even in processes that are very loosely structured, Gallardo-Valencia and Sim argue that (at least in software processes) actors will often improvise from the specification [54]. In regard to software processes, Engels et al. assert that hardly any “developer wants to use highly integrated and process enforcing environments” [55]. The authors add, “most supporting tools tend to overburden the early activities in the software process with guidelines, rules and syntactical support”. Actors will then resort to going behind the system’s back, which can make the system more of a liability than an asset [40].

Flexibility has subsequently grown to become a major research topic in the area of workflow management [41]. A high degree of unstructured practice can become unmanageable, but inflexibility results in the “loss of creativity needed for sustained innovation” [34]. Ideally, processes should combine the need for rigor and discipline with the need for flexibility and creativity, but that balance is hard to achieve [35] and will be situation dependent.

However, flexibility is not only concerned with the trade-off between actors’ creative freedom and freedom to partake in risky activity. The more controlled a process is, the harder it is to improve.

Process observation systems however are more concerned with observing, monitoring and recording what takes place, and so should not have any direct influence on the enactment of the process. This approach is best used where the complete flexibility afforded to actors and the inability to prevent defects or non-conformance from occurring is not an issue. Observation-based approaches are therefore very suited to non-conformance detection approaches.

2.5.2 Which type is more appropriate?

When seeking to implement a process improvement approach, consideration should be taken of the trade-offs between support and observation-based methodologies. Depending on the goal, the desirable balance for the approach will be different. Will inhibiting flexibility create value for the approach? Table 2 summarises the considerations between support and observation-based approaches.

Consideration	**Process Support**	**Process Observation**
Flexibility	Lower	Higher
Risk	Lower	Higher
Capacity for process improvement	Lower	Higher
Provision for creativity	Lower	Higher
Level of interference with the enactment	Higher	Lower
Negative effect on actor contentedness	Higher	Lower

Table 2 Process Support vs. Process Observation

The key difference between a defect (or fault) and non-conformance here is that non-conformance may not necessarily be a negative thing. It is entirely possible that non-conformance can be a positive influence on the effectiveness or efficiency of the process. For this reason, non-conformance or defect prevention is, in fact, an undesirable mechanism for an approach aimed at detecting non-conformance.

2.6 Non-Conformance Based Issues

For non-conformance detection approaches, besides the challenge of actually detecting as many non-conformance instances as possible, there will be another set of considerations and issues that should be acknowledged. These considerations are covered in this section and can be summarised as:

- the importance of timely detection
- the availability of process enactment data
- the work involved in data discovery including data needing to be rediscovered once the process has been changed/improved
- the reasons for non-conformance occurring
- the severity and consequences
- the guidance and recommendations on actions to take after non-conformance is detected

2.6.1 Timely detection

The collection and organisation of process event data can be time consuming, delaying the detection of non-conformance instances. Some approaches in the literature that require event data to be mined such as Huo et al. [19] make on-the-fly detection impossible, however timely detection can have some desirable benefits.

If the process is enacted repetitively, fast detection can help ensure any consequences do not carry over into future enactments.

As with defect detection, early detection of non-conformance is an important step towards the timely remediation of any consequences associated with it [61] [74], thus reducing the degree of risk to the organisation.

The quicker a non-conformance instance can be detected (even if it is not on-the-fly), the quicker improvements can be identified and implemented.

The cause of non-conformance is much easier to identify if the instance of non-conformance is detected as quickly as possible. Retrospective identification of the cause becomes more difficult the longer the amount of time waited before detecting it has occurred.

On-the-fly detection is most desirable because it allows us to take reactive measures as soon as non-conformance is detected [74]. Late detection can dampen the effect of any reactive measures taken in addition to exacerbating any potential consequences.

In addition to process support mechanisms with the purpose of preventing non-conformance, and approaches where on-the-fly detection is possible such as in

Cîmpan and Oquendo [30], there has been little presented in the literature stressing the importance of optimising the timely detection of non-conformance.

2.6.2 Process enactment data availability

In order to detect non-conformance, process enactment data must be available in some form for checking and comparison. The way this data is recorded and stored will be different for every process and will significantly affect the success of any non-conformance detection approach. As Sørumgård argues in his thesis on non-conformance, the ability to measure or observe the process enactments is of critical importance in non-conformance detection [68].

The scope of this research is confined to non-conformance detection, not monitoring process enactments, so although the acquisition of process enactment data is distinct from the research objective, it is still an important requirement for a non-conformance detection approach to succeed. Rozinat and van der Aalst resolve this problem in their research on non-conformance detection by stating the assumption that all process event data is automatically written to event logs, with which they can use to apply their non-conformance detection technique [71].

The collection of data can be a time consuming and expensive endeavour, a factor contributing to why the use of simulation has become prominent in testing different research methodologies where the use of real world environments is too problematic [64].

If the process in question is highly automated or a large proportion of it is carried out using computers or machinery, then the process enactment data may be more easily recorded. Human tasks are more difficult, especially human tasks carried out away from the computer, which are especially difficult and expensive to monitor [44].

If the system in question is critical and the processes involved are highly human oriented, there are techniques and research methodologies available to aid in the accurate collection of data for use in process improvement. A good example of this is the use of radio frequency identification technology to capture emergency room data in a hospital [66].

There is also the question of when the enactment data will be made available for the purposes of non-conformance detection. If the data is not available until the process has concluded, then this makes fast detection more difficult and on-the-fly detection impossible, regardless of the approach used.

Another concern that may have to be dealt with is the possibility that there may not be a complete set of useful enactment data with which to work. The non-conformance detection approach, whatever it might be, may require data in a form or in an abundance that is simply not available. We must then make the most of the data provided.

Having considered these issues, this research is concerned with non-conformance detection, so a necessary assumption is that we have access to at least some process enactment data in some form. However the availability of process enactment data will have a significant effect on any approach being considered, regardless of what the approach is.

2.6.3 Why non-conformance occurs

An understanding of the source of non-conformance and why it occurs is critical if we hope to detect occurrences. It may be argued that non-conformance does not just "occur"; rather, it is people who fail to conform for various reasons. These reasons can be summarized in a list, which is provided by Perry and Wolf in [49] along with our own two additions at the end provided in Thompson et al. [57]:

- the process definitions omit or do not allow for relevant project contingencies;
- sometimes risks are taken;
- some process definitions are more amenable to non-conformance;
- people have good ideas, some of which are better than the defined processes;
- the process does not make sense because of individual differences or because of lack of training (also confirmed by Rozinat and van der Aalst [70]);
- a lack of commitment or interest;
- sometimes, an otherwise competent user may simply make a mistake;
- malicious people may sabotage or deliberately try to cause problems.

The essential cause of non-conformance can nonetheless always be attributed to human behaviour. As mentioned in section 2.3.6, the more automated a process or process activity is, the less likely non-conformance is to occur (despite the other issues and considerations that come with over-automating and enforcing a process). Therefore, the accurate monitoring and recording of data in process activities with a high level of human involvement is one of the key challenges to be faced regarding non-conformance detection.

2.6.4 Effort required

Although minimising the effort and cost involved should be a factor for any kind of process improvement, the area of non-conformance detection has seen some fairly highly involved approaches presented in the literature. The issue is that a non-conformance mechanism is only useful to an organisation if the effort involved in employing it does not offset the cost of performing the process itself. It is important to show organizations evidence of the possible benefits of any process improvement initiatives before they will agree to implement them [75]. The method of process discovery is a prime example to use in this case.

Process discovery is the practice of mining the process enactment data for the purposes of "discovering" a process model from it. The motivation for doing so is twofold. Initially, it affords us a means to observe how processes are *actually enacted* and secondly, it provides actual data, which may be compared to the defined model to detect instances of non-conformance.

The technique is a promising one, and was the basis of the approach presented by Huo et al. [19]. The major issue with it is the immense amount of data required to discover a process model. This is also the major point of criticism in the similar approach proposed by Cook and Wolf [13]. If the process is not executed often, this could be problematic to obtain.

This issue can cause a second problem, common to all effort intensive approaches, not just process discovery-based ones. The issue is once the process has been improved as a result of the effort applied, the process specification is now different – improved but still different. So, all the effort put in to discover the process model or apply the approach is now voided because it applied to an earlier and different version of the process. That effort may now need to be repeated if continuous improvement is desired.

Huge set-up efforts will also extend the amount of time it takes to improve the process along with evaluating the effects of the improvement. In fact, the consequences of the detected deviation or inconsistency may have already been resolved in the time it took to detect it, if an inordinate amount of time is required to gather the process data and mine it for detection to take place.

This is not to say collecting process data for mining is necessarily a negative thing. Apart from the positives noted by Cîmpan and Oquendo [30], Huo et al. [19], and Cook and Wolf [13], mining process data can also provide us with potentially very useful data and statistics on how the process behaves, which provides potential for further improvements to the process.

Thus far, it seems as though a lot of the approaches focus on the challenge of comprehensive detection. The secondary goals of providing a flexible, unobtrusive, easy-to-use and fast-to-implement tool for practical process improvement will likely get more attention as the research in this area starts to mature.

2.6.5 Severity

Once non-conformance has been successfully detected, a metric on the degree of difference between the expected value and the actual value would be a useful piece of data. For the purposes of this research, the term "severity" is coined to encompass this meaning. Again, this is an area likely to be given more attention once the research in this area matures.

Establishing a consistent degree of difference from one non-conformance instance to the next is challenging, because not all instances will return numeric values and different processes are more critical than others. Furthermore, different activities within the same process can have a greater priority than others. Non-conformance severity will therefore not only be a measure of how far an actual value has strayed, but may also depend on where in the process it occurred, what time it occurred and the perceived criticality of the process itself.

For non-conformance instances that return numeric data, there are advanced techniques such as Statistical Process Control (hereafter SPC). Using SPC helps to determine appropriate boundary values based on past data that detect real deviations as well as reducing the risk of false alarms [21] [58] [59]. It also provides a mechanism to determine a severity value by giving how many standard deviations actual enactment data values fall outside the control range.

However, not all process event data is numeric. There is currently an opportunity to develop a methodology capable of taking different data types as input and deriving an appropriate non-conformance severity value, based on the expected value, the actual value and other relevant process attributes such as process and activity priorities. An account of how numeric and non-numeric data can be given consistent non-conformant severity values was presented in Thompson and Torabi [63].

2.6.6 Post-detection Guidance

If we assume that non-conformance instances have been detected and appropriate severity ratings have been applied to each instance, the next step is to fathom a way of using this information to create some kind of value for the process and the organisation.

Depending on the relative severity rating of a non-conformance instance, certain people may have to be notified that it has occurred. Certain actions may also need to be taken to rectify any problems the non-conformance might have caused. The challenge here is to conceive a way of determining who should be informed and what should be done depending on the non-conformance instance that was detected.

Currently, the literature on the research in this field has yet to explore this tangent of non-conformance detection. However, there are mechanisms that could be applicable that have been developed, such as recommender systems where choices are filtered and the best ones presented to the user to cope with information overload [60].

If certain remedial actions are preliminarily stored and logged when they are executed as a result of non-conformance occurring, then a non-conformance instance history log coupled with an action and effectiveness log could provide helpful data to a recommender system to help with action advice.

Although applying this kind of value to a process first depends upon non-conformance being consistently and comprehensively detected and that each instance can be given a consistent and appropriate severity rating, it is important to keep in mind this eventuality when considering possible non-conformance detection approaches.

2.7 Conclusion

From the literature it is clear there is a lot of opportunity for future work in this research field. The key goal moving forward from the existing literature is making the clear distinction between defect prevention and non-conformance detection. If we accept that non-conformance is not necessarily a defective or a negative thing, then the real research objective is simply to detect it. This implies that an ideal approach should be as unobtrusive as possible.

From the issues and approaches examined, the next step besides the actual detection of non-conformance is to provide a practical approach; one that is observatory in nature, does not interfere with the existing process or its operating environment, and minimises the effort and cost required to implement it.

Forethought should be given to what must be done after non-conformance is detected. What is its severity and what might the consequences be? Is there any way to indicate appropriate remedial action after the fact? Chapter 3 focuses on the opportunities that continue to present themselves to researchers in this area and how to formalize a practical approach, one that is attractive to organisations in a commercial setting, that can be easily implemented with these issues in mind.

References

[1] V.R. Basili, F.E. McGarry, R. Pajerski, M.V. Zelkowitz, Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory, Proceedings of the 24th International Conference on Software Engineering, ACM Press, May 2002.

[2] A. Fuggetta, Software Process: A Roadmap, Proceedings of the Conference on the Future of Software Engineering, ACM Press, May 2000.

[3] T. Barothy, M. Peterhans, K. Bauknecht, Business process reengineering: emergence of a new research field, ACM SIGOIS Bulletin, Volume 16 Issue 1, August 1995.

[4] B. Curtis, Software Process Improvement: Methods and Lessons Learned, Proceedings of the 19th international conference on Software engineering, ACM Press, May 1997.

[5] "Workflow Management Coalition," http://www.wfmc.org.

[6] D. Avrilionis, N. Belkhatir, P.Y. Cunin, A unified framework for software process enactment and improvement, 4th International Conference on the Software Process, Brighton, England, 3-5 December, 1996.

[7] T. Dybå, Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context, ACM SIGSOFT Software Engineering Notes, Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-11, Volume 28 Issue 5, ACM Press, September 2003.

[8] M.A. Lantzy, Application of statistical process control to the software process, Proceedings of the ninth Washington Ada symposium on Ada: Empowering software users and developers, ACM Press, July 1992.

[9] D.M. Ferrin, M.J. Miller, D. Muthler, Lean Sigma and simulation, so what's the correlation? V2, Winter Simulation Conference, 2005 Proceedings of the, Bus. Prototyping Inc., Naperville, IL, USA, 4-7 Dec. 2005.

[10] M. Hammer, J. Champy, Reengineering the Cooperation, Harper Business, New York, 1993.

[11] S. Acuna, A. De Antonio, X. Ferré, M. López, L. Maté, The Software Process: Modelling, Evaluation, and Improvement, Handbook of Software Engineering and Knowledge Engineering, World Scientific Publishing Company, 2000.

[12] T. Davenport, Process Innovation: Re-engineering Work through Information Technology, Harvard Business School Press, Boston MA, 1993.

[13] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 7 Issue 3, July 1998.

[14] M. Komuro, Far east experience papers: software process: Experiences of applying SPC techniques to software development processes, Proceeding of the 28th international conference on Software engineering ICSE '06, ACM Press, May 2006.

[15] V. Ambriola, R. Conradi, A. Fuggetta, Assessing process-centered software engineering environments, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 6 Issue 3, ACM Press, July 1997.

[16] A. Blyth, Business process re-engineering: What is it?, ACM SIGGROUP Bulletin, Volume 18 Issue 1, April 1997.

[17] M. Dowson, B. Nejmeh, W. Riddle, Concepts for Process Definition and Support, Proceedings of the 6th International Software Process Workshop, IEEE Computer Society Press, October 28-31 1990, Hakodate, Japan.

[18] P. Struss, Deviation Models Revisited, Working Papers of the 15th International Workshop on Principles of Diagnosis (DX-04), June 2004.

[19] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[20] Y. Rezgui, F. Marir, G. Cooper, J. Yip, P. Brandon, A Case-Based Approach to Construction Process Activity Specification, Intelligent Information Systems, 8-10 Dec, 1997, pp. 293 – 297.

[21] W.A. Florac, A.D. Carleton, Measuring the Software Process: Statistical Process Control for Process Improvement, Addison-Wesley, 1999.

[22] "Process" definition, Merriam-Webster, http://www.merriam-webster.com/dictionary/process, May, 2010.

[23] "Specification" definition, Merriam-Webster, http://www.merriam-webster.com/dictionary/specification, May, 2010.

[24] C.D. Klingler, A Case Study in Process Definition, Proceedings of the conference on TRI-Ada, ACM, October 1993.

[25] L. Osterweil, Automated Support for the Enactment of Rigorously Described Software Processes, 4th international software process workshop on Representing and enacting the software process, ACM, April, 1988.

[26] A. Caetano, J. Tribolet, Modeling organizational actors and business processes, ACM symposium on Applied computing, ACM, April, 2006.

[27] I. Sommerville, Software Engineering, 8th edition, Addison Wesley, 2007, pp. 74.

[28] A. Caetano, A.R. Silva, J. Tribolet, Using roles and business objects to model and understand business processes, ACM symposium on Applied computing, ACM, March, 2005.

[29] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[30] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[31] J.D. Reese, N.G. Leveson, Software deviation analysis, Proceedings of the 19th international conference on Software engineering, ACM Press, May 1997.

[32] G. Cugola, E. Di Nitto, A. Fuggetta, C. Ghezzi, A framework for formalizing inconsistencies and deviations in human-centered systems, ACM Transactions on

Software Engineering and Methodology (TOSEM), Volume 5 Issue 3, ACM Press, July 1996.

[33] J.E. Cook, A.L. Wolf, Automating process discovery through event-data analysis, Proceedings of the 17th international conference on Software engineering ICSE '95, ACM Press, April 1995.

[34] J.S. Brown and P. Duguid, Balancing Act: How to Capture Knowledge without Killing It, *Harvard Business Review*, Vol. 78, No. 3, 2000, pp. 73-80.

[35] R.L. Glass, *Software Creativity*, Englewood Cliffs, New Jersey, Prentice-Hall, 1995.

[36] R. Harper, M.J. O'Loughlin, Manufacturing process analysis—tools and applications, Proceedings of the 19th conference on Winter Simulation, ACM, 1987.

[37] N.H. Madhavji, K. Toubache, E. Lynch, The IBM-McGill project on software process, conference of the Centre for Advanced Studies on Collaborative research, IBM Press, October 1991.

[38] G.Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[39] D. Bogia, S. Kaplan, Flexibility and Control for Dynamic Workflows in the wOrlds Environment, Proceedings of the Conference on Organisational Computing Systems, ACM Press, Milpitas, CA, November 1995.

[40] W.M.P Van der Aalst, S. Jablonski, Dealing with workflow change: identification of issues and solutions, International Journal of Computer Systems Science and Engineering, pp 267-276, September 2000.

[41] P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, M. Teschke, A comprehensive approach to flexibility in workflow management systems, ACM SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99, Volume 24 Issue 2, ACM Press, March 1999.

[42] B. Curtis, M.I. Kellner, J. Over, Process Modeling, Communications of the ACM, Volume 35 Issue 9, ACM Press, September, 1992.

[43] X. Zhao, K. Chan, M. Li, Applying agent technology to software process modelling and process-centered software engineering environment, Proceedings of the 2005 ACM symposium on Applied computing SAC '05, ACM Press, March 2005.

[44] J.E. Cook, A.L. Wolf, Software process validation: quantitatively measuring the correspondence of a process to a model, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 8 Issue 2, ACM Press, April 1999.

[45] M.M. Fathee, R. Redd, D. Gorgas, B. Modarres, The effects of complexity on business processes reengineering: values and limitations of modeling and simulation technologies, 30th conference on Winter Simulation, IEEE Computer Society Press, December 1998.

[46] M. Van Hilst, P.K. Garg, C. Lo, Repository mining and Six Sigma for process improvement, ACM SIGSOFT Software Engineering Notes, Proceedings of the 2005 international workshop on Mining software repositories MSR '05, Volume 30 Issue 4, ACM Press, May 2005.

[47] B.C. Boynton, Identification of process improvement methodologies with application in information security, 4th annual conference on Information security curriculum development, ACM, September 2007.

[48] G. Grossmann, M. Schrefl, M. Stumptner, Modelling inter-process dependencies with high-level business process modelling languages, fifth on Asia-Pacific conference on conceptual modelling, Australian Computer Society Inc, Volume 79, January, 2008.

[49] D.E. Perry, A.L. Wolf, Session 1: People, Processes, and Practice, Proceedings of the 9th International Software Process Workshop, IEEE Computer Society Press, October 5-7 1994, Airlie, Virginia, USA.

[50] B.M. Carter, J.Y.C. Lin, M.E. Orlowska, Customizing internal activity behaviour for flexible process enforcement, 15th Australasian database conference, Australian Computer Society Inc, Volume 27, January 2004.

[51] A. Borgida, T. Murata, Tolerating exceptions in workflows: a unified framework for data and processes, ACM SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99, Volume 24 Issue 2, ACM Press, March 1999.

[52] M. Ramage, Engineering a smooth flow? The links between workflow and business process reengineering, MSc thesis 1994, University of Sussex, England.

[53] L. Aversano, G. Canfora, Introducing eservices in business process models, Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02, ACM Press, July 2002.

[54] R.E. Gallardo-Valencia, S.E. Sim, Planning and improvisation in software processes, ACM, Volume 14, Issue 1, September, 2007.

[55] G. Engels, W. Schäfer, R. Balzer, V. Gruhn, Process-centered software engineering environments: academic and industrial perspectives, Proceedings of the 23rd International Conference on Software Engineering ICSE '01, IEEE Computer Society, July 2001.

[56] J. Eder, W. Liebhart, The Workflow Activity Model (WAMO), 3rd international conference on Cooperative Information Systems (CoopIs), 1995.

[57] S. Thompson, T. Torabi, P. Joshi, A Framework to Detect Deviations During Process Enactment, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

[58] W.A. Florac, A.D. Carleton, J.R. Barnard, Statistical Process Control: Analyzing a Space Shuttle Onboard Software Process, IEEE Software, vol. 17, no. 4, July/Aug. 2000, pp. 97–106.

[59] P. Jalote, A. Saxena, Optimum control limits for employing statistical process control in software process, IEEE Transactions on Software Engineering, Volume 28, Issue 12, Dec. 2002 Page(s): 1126 – 1134.

[60] Y.Z. Wei, L. Moreau, N.R. Jennings, A market-based approach to recommender systems, ACM Transactions on Information Systems (TOIS), Volume 23 Issue 3, ACM Press, July 2005.

[61] M. Sherriff, N. Nagappan, L. Williams, M. Vouk, Early estimation of defect density using an in-process Haskell metrics model, 1st international workshop on Advances in model-based testing (A-MOST), ACM, July, 2005.

[62] J. Mendling, G. Neumann, W. van der Aalst, On the correlation between process model metrics and errors", 26th International Conference on Conceptual Modeling, Volume 83, Australian Computer Society Inc, November, 2007.

[63] S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

[64] N.H. Robertson, T. Perera, General manufacturing applications: feasibility for automatic data collection, 33rd Conference on Winter Simulation, IEEE Computer Society, December, 2001.

[65] J.H. Kroeze, M.C. Matthee, T.J.D. Bothma, Differentiating data- and text-mining terminology, conference of the South African institute of computer scientists and information technologists on Enablement through technology, South African Institute for Computer Scientists and Information Technologists, September, 2003.

[66] M.J. Miller, D.M. Ferrin, T. Flynn, M. Ashby, K.P. White, Jr., M.G. Mauer, Using RFID technologies to capture simulation data in a hospital emergency department, 38th conference on Winter simulation, Winter Simulation Conference, December, 2006.

[67] G. Cugola, Tolerating Deviations in Process Support Systems VI Flexible Enactment of Process Models, IEEE, Transactions on Software Engineering, vol. 24, No 11, 1998.

[68] S. Sørumgård, Verification of Process Conformance in Empirical Studies of Software Development. Ph.D. thesis, Norwegian University of Science and Technology, 1997.

[69] N. Zazworka, V.R. Basili, F. Shull, Tool Supported Detection and Judgment of Nonconformance in Process Execution, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, October 2009.

[70] L.F.S. Silva, G.H. Travassos, Tool-Supported Unobtrusive Evaluation of Software Engineering Process Conformance, International Symposium on Empirical Software Engineering, ISESE '04, ISBN:0-7695-2165-7, September 2004.

[71] A. Rozinat, W.M.P. van der Aalst, Conformance Checking of Processes Based on Monitoring Real Behaviour, Journal of Information Systems, Volume 33, Issue 1, Elsevier Science Ltd., Oxford, UK, March 2008.

[72] W.M.P. van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, H.M.W. Verbeek, Choreography Conformance Checking: An Approach based on BPEL and Petri Nets, Proceedings of the Seminar on The Role of Business Processes in Service Oriented Architectures, Dagstuhl, Germany, July 2006.

[73] N. Zazworka, K. Stapel, E. Knauss, F. Shull, V.R. Basili, K. Schneider, Are Developers Complying with the Process: An XP Study, Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Bolzano, Italy, September 2010.

[74] M.A.A da Silva, R. Bendraou, X. Blanc, and M.P. Gervais, Early Deviation Detection in Modeling Activities of MDE Processes, Proceedings of the 13th international conference on Model driven engineering languages and systems: Part II, Oslo, Norway, October 2010.

[75] S.U. Hani, Impact of process improvement on software development predictions, for measuring software development project's performance benefits, Proceedings of the 7th International Conference on Frontiers of Information Technology, Abbottabad, Pakistan, December 2009.

Chapter 3 – Problem Statement

This chapter will provide an overview of the problem domain and present the specific issues that the proposed solution, covered in chapter 4 and 5, will attempt to resolve. The key issues facing the generic detection of non-conformance are discussed along with the importance of why non-conformance detection approaches need to address them.

The high level objective of this research is to provide a medium for process improvement through non-conformance detection. The methodology used to detect non-conformance in the best possible way is the focus of this thesis. The issues presented in this chapter relate to the gathering of knowledge of the process, and of non-conformance instances detected and then using that knowledge to improve the process. However there are challenges that need to be addressed in order to successfully achieve this goal.

For each issue facing researchers in this line of work, an analysis will be provided to consider why the issue is important, how the issue will challenge people seeking to detect non-conformance, and also how each issue has affected some of the approaches and the work already presented in this research domain. Additionally, the potential benefits of addressing these issues are also discussed from the viewpoint of process improvement but also from an organisational point of view.

A plan for the rest of the thesis will then be presented. This plan will begin with a brief overview of how the proposed approach will function, followed by a description as to how it will be presented throughout the rest of the thesis, focussing on how this solution will be implemented, tested and evaluated.

At the conclusion of this chapter, the important aspects related to this research that the reader will have covered can be summarised as:

- the main issues facing non-conformance detection;
- why these issues are important;
- how these issues are not being appropriately addressed by alternate approaches presented in the literature; and

- the benefits that can be contributed to process non-conformance by resolving these issues.

3.1 Key Issues Facing Non-Conformance Detection

In this section, the issues facing process non-conformance detection are covered with an analysis on why it is important that non-conformance detection approaches address these issues.

The issues covered in this section as ones of importance that require addressing are:

- adequately defining non-conformance, its scope and the terms associated with it;
- minimising the level of interference a non-conformance detection approach will have on its target process;
- minimising the cost and effort required to implement the non-conformance detection approach with the process;
- keeping the non-conformance detection approach flexible and scalable to maximise its capability to evolve with the process;
- maintaining an approach generic enough to be applied to many types of processes across different domains;
- the use of observational approaches to maximise potential for detection, not prevention of non-conformance;
- structuring the approach in a way that is mindful of future actions after non-conformance detection, such as determining severity and taking remedial action;
- minimising the time between non-conformance occurrence and non-conformance detection; and
- ensuring that enough process enactment data exists in a consistent and useable structure such that a non-conformance approach can be properly applied.

3.1.1 Inconsistent Use of Concepts and Terminology

As discussed throughout section 2.2, there has not been much consistency both in the terms used to describe non-conformance and other terms related to it, but also in the consistency with which the concepts are presented. Cugola et al. presented an attempt at formalizing some of these concepts as far back as 1996 [11], however these concepts and definitions have not seemed to have found their way consistently into the vernacular of researchers in this domain.

Inconsistent terminology is often a common characteristic of young research areas, although process non-conformance detection is not altogether so young any more and there is an opportunity to standardise the concepts and terminology in this field.

One of the challenges facing a non-conformance approach therefore, is to adequately define the scope of non-conformance, to clearly state what constitutes non-conformance, the attributes and sub-concepts associated with it and to define and use consistent and clear terminology regarding non-conformance and its detection.

Non-conformance has been referred to in many different ways (as shown in chapter 2, section 2.2.2) and many different concepts have been presented and redefined in this domain. As such, it can be difficult to determine exactly what is meant when a certain term is used in the literature unless the author explicitly defines its meaning and context such as the definitions of *inconsistency* and *deviation* in Cugola et al. [11], which incidentally are different to the definition of terms given in papers such as Kabbaj et al. [6]. The explicit definition of what is meant when using certain terms in this research area will likely have to continue for the time being in the interests of clarity, however the formalization and standardization of concepts inherent to this research would benefit researchers' understanding when working in this area.

3.1.2 Interference with the Existing Process

This follows from the observation approach against the support approach dichotomy discussed in chapter 2, section 4. Process support systems are intended to ensure that the enacting process conforms to its specification [1], and so they have a way of interfering with the process enactment when they are implemented. This is not to say they are not beneficial, but they reduce flexibility and decrease the chances of detecting non-conformance for the perceived benefits of preventing it.

A high level of interference with a process can have a negative impact on the benefits we are trying to achieve with a non-conformance detection approach. Interfering with a process enactment means that actors now have added considerations to worry about and resources that would otherwise have been utilised for the process enactment may be redirected into implementing the non-conformance detection approach.

If an approach can be integrated with the target process without interfering with it, this can have follow-on benefits such as the reduction of risk and a reduction in the amount of resources and effort required to implement it.

A non-conformance detection approach should only need access to the enactment data in order to achieve its objective. Interference for the purposes of non-conformance prevention or process control is not necessary if the goal is simply to detect the occurrences of non-conformance. As long as there is adequate enactment data available there should be no legitimate need to interfere with the process enactment.

3.1.3 Effort and Cost Required

The greater the amount of work required to implement a non-conformance detection approach, the greater the risk that this will offset the benefits yielded by either the intended improvement or the process itself. In other words, if the cost involved in implementing an improvement is disproportionate to the surmised benefit from doing so, then it is not worth doing regardless of its effectiveness. Conversely, if the implementation costs are low and the potential benefit is high then this makes the approach all the more attractive. There is high demand in industry to know the impact and benefits of process improvement initiatives before organizations invest in them [16].

This important consideration has been somewhat overlooked in the literature, with researchers seemingly preferring to perfect the most effective way of detecting non-conformance rather than focussing on commercial practicalities. The overburdening of effort has been acknowledged in Cook and Wolf [2] and Huo et al. [3] with Huo et al. [3] making a deliberate attempt to curtail the level of effort required in the process discovery for their approach. The later approach by Zazworka et al. [14] also specifically mentions the need to worry about the "cost" of measuring an enacting process for the purposes of detecting non-conformance.

There is also a problem regarding the level of effort required to successfully define all the different things a process should not do. This is relevant to approaches that attempt to list all possible non-conformance types and then check to see if they have occurred, such as approaches like Kabbaj et al. [6] or Zazworka et al. [14]. Aside from attempting to anticipate everything that process should not do being an extremely difficult task [7], it can also be an arduous and time consuming one.

Since low cost, high benefit process improvement initiative are practically more desirable than those that are not, and the detection of non-conformance is essentially a process improvement initiative, then the minimisation of effort in implementation is an important consideration for any non-conformance detection approach.

3.1.4 Flexibility and Scalability

Different processes will have different sizes and different numbers of modular, separate tasks. At times, we may only want to concentrate on a particular aspect of the process for improvement. Therefore, there are benefits to be enjoyed by formulating process improvement approaches that can be applied to different and specific parts of the process at a time.

Conversely, as the process evolves, the approach should be scaled with the evolving process. Some parts of the methodology may not be applicable any longer to process sections that have changed over time, so the approach should be able to easily accommodate only these changes that are required.

It is unclear how process discovery non-conformance approaches such as Huo et al. [3] would handle evolution, given that the process model derived from the enactment data would likely have to be rediscovered when the process changes. If there is a large amount of work in doing this, it can be problematic. It is also unclear how the "ideal enactment" benchmark from Cîmpan and Oquendo [4] would need to be changed along with the changes in a process specification.

In any case, a non-conformance approach that affords flexibility and scalability when dealing with an evolving process can be highly beneficial. A secondary objective in the proposed approach is to facilitate this, such that when a process changes, only the affected parts of the implementation need be addressed; it should not require an entire reimplementation.

3.1.5 Limited Applicability

In the literature, approaches seem to be geared for processes in specific domains, such as the software process, the business process or the manufacturing process. If we obtain a lot of knowledge about the domain and the types of processes we are dealing with, then dealing with them becomes much easier.

However, the solutions and improvement approaches implemented for a specific purpose become less adaptable to different processes in different situations. Since non-conformance is a problem associated with all process types in all different domains, a non-conformance detection approach that can be applied generically to any process could be greatly beneficial.

It is therefore a challenge for any non-conformance detection approach to keep the solution generic enough that it can be applied to a wide range of different processes, but also effective enough that it can consistently and adequately detect non-conformance when it is implemented.

3.1.6 Capitalising on Possible Benefits

This issue is related to the minimising of interference issue. A non-conformance detection approach can be implemented in two ways, as discussed in section 2.5. In the first way, if the intention is to detect and prevent, is to use a process support system is used which can help prevent the potential consequences of non-conformance by stopping them from occurring. The second way is to simply observe the process being enacted, and to analyse the resulting data to detect any non-conformance instances. This kind of approach will not prevent the non-conformance from occurring but it should have little to no interference on the process to which it is applied.

Process support systems by nature tend to interfere with the process more than observational approaches, which is something we are trying to minimise. One of the main purposes of a process support system is to ensure the enactment conforms to its specification, something that conflicts with the whole purpose of non-conformance detection [12].

Other characteristics common to process support systems such as process control and non-conformance prevention are not required when attempting to detect non-conformance and in fact, these characteristics can be completely undesirable as they can interfere with the detection mechanism.

The primary objective of non-conformance detection approaches should be simply to detect that non-conformance has occurred and gather as much knowledge about the non-conformance as possible. An observational approach only requires access to the enactment data to perform its primary task, which negates any need to implement a controlling and interfering mechanism like a process support system.

Approaches of this kind do not need to prevent non-conformance from taking place, especially considering that non-conformance has the potential to be a beneficial change, something that has been overlooked in certain other approaches such as Kabbaj et al. [6].

As we desire to keep interference to a minimum, and only detect non-conformance instead of preventing it, an observational system is a much better option for a non-conformance detection approach than process support.

3.1.7 Supporting What Happens After Detection

As discussed in chapter 2, little research has been conducted in this area concerning approaches aiding the steps to be taken after non-conformance has been detected, aside from the publication highlighting this problem and offering a rudimentary solution in Thompson and Torabi [8]. After an instance of non-conformance has been detected, it is hoped that this can be used in some way to help improve the process. Researchers in this area must consider the potential ways in which this might be done.

One step likely to be of use after non-conformance has been detected is ascertaining some kind of value indicative of how *severe* a particular instance of non-conformance is relative to the rest of the process. Another step is deciding what should be done to address any consequences the non-conformance might have caused.

Different people will have different approaches in addressing after-detection considerations such as these. The challenge then, is to structure the approach and the knowledge base of non-conformance such that these future steps can be more easily achieved. Detection of non-conformance should be approached in such a way that considerations such as determination of a severity rating and recommendation of appropriate remedial action would be the logical next step in the path to process improvement.

Furthermore, as much information as possible about the non-conformance instance must be provided. The more data available, the better informed a process administrator will be when making decisions on a course of action following a non-conformance detection. At present, no existing approaches offer any kind of assistance in this regard.

3.1.8 Time from Non-Conformance Occurrence to Detection

There are numerous benefits associated with the fast detection of non-conformance. Ideally, we should aim to detect non-conformance on-the-fly (as it happens), but this is not possible in some cases. Still, decreasing time to detection is a worthwhile pursuit but is a research area that does not receive much attention in the literature. It is certainly worthwhile investing some effort into minimising the amount of time it takes to detect non-conformance after it occurs as much as possible.

This issue is not one of concern in most process support systems such as Cugola et al. [5] and even Cîmpan and Oquendo [4] because if the goal is to prevent then this occurs on-the-fly by default. Approaches where a lot of process data needs to be mined such as in Huo et al. [3] can slow down detection although Huo et al. [3] make an effort to mine process enactment data at the same level as the specification in order to reduce this.

It is normally easier to deduce the cause of non-conformance if it is detected quickly. Since the surrounding environment will be largely similar to what it was when non-conformance was detected, examination of the process environment can yield clues as to why a process did not conform to its specification if it is performed quickly.

Similarly, any consequences that may arise due to non-conformance occurring can be mitigated better if it is dealt with quickly. Zazworka et al. [14] argue that early detection of non-conformance can help to minimise impact to schedule and quality, though they state this is a consideration for their future work in non-conformance detection rather than a specific feature of the current generation of their approach.

The sooner non-conformance is detected, the sooner initiatives may begin to improve the process. It is for these reasons that any approach being considered should work as expediently as possible in its operating environment.

3.1.9 Acquiring Process Enactment Data

As previously discussed in chapter 2, the availability of process enactment data is a pre-requisite for the application any kind of non-conformance detection approach. Sørumgård argues that a critical requirement for successful non-conformance detection is satisfactory observation of the process enactments [15]. The successful application of the proposed approach is also dependent on an adequate availability of process enactment data. Simply put, if the process enactment data cannot be captured in such a way that it exists in satisfactory abundance and quality, then non-conformance detection becomes impossible regardless of the approach used.

However, different processes can be specified and enacted in very different ways, especially processes from different operating environments. The ways in which process enactment data is captured and recorded will therefore be different for each different process. For example, the acquisition of enactment data from an online process will be performed in an online capacity. The capturing of enactment data for a process that is enacted online may be vastly different from the way data is captured for a manufacturing process. This makes the challenge of acquiring process enactment data different for each process, especially given that a goal of the proposed approach is that it may be generically applied to any suitable process.

The problem with process enactment data availability is that its acquisition is separate from non-conformance detection. Researchers who are studying ways to detect non-conformance need to assume that enactment data will be made available to them in a structured and consistent format, without worrying about developing alternate solutions focussing on how to collect it.

Therefore, the important issue to be considered when implementing a process non-conformance detection approach is that there is an adequate availability of consistently structured process enactment data before beginning to implement the approach itself.

3.2 Overview and Scope of the Proposed Solution

The main focus of the proposed approach, aside from actually detecting non-conformance, is to keep things as simple as possible. A generic, structured methodology is desired that can be applied to as many processes as easily as possible. The approach has to be flexible to change without affecting things around it and have the capacity to evolve with the process itself. It also must bear a relatively small burden on the person whose responsibility it is to integrate it. Such an approach should yield benefits of the kind not yet presented in the literature by addressing the aforementioned issues in section 3.1.

From these goals, an approach was formalized based on *rule specification,* which is covered in greater detail in chapter 4. The rules are defined separately from the process specification and act as a medium for comparing it to the process enactments. These rules complement the specification by defining conditions by which the enactments should hold and conditions that should not hold.

For example, a process specification may require an activity to be enacted only after another different activity has concluded, and also within a given time duration. We can define a set of rules to test that enactment data conforms to this specification such as:

- activity X must be complete;
- activity time duration must be greater than or equal to Y seconds;
- activity time duration must be less than Z seconds.

Rules such as these may be defined to complement this while keeping the rules distinct from the original specification. Once the process enactment reaches this stage and enactment data becomes available, we can apply these rules to the enactment data to test whether or not the enactment has "conformed" to its specification.

There is, of course, a great deal more to this approach than what has been briefly described, however this is the basis for the proposed approach. The way the proposed approach addresses the aforementioned issues is the focus of this section.

3.3 Plan to resolve the problem and associated issues

This section presents the plan by which the problem of process non-conformance detection will be addressed, tested and evaluated with respect to the issues covered in this chapter.

3.3.1 Proposed Solution

In chapter 4, the proposed solution to detecting non-conformance will be presented in detail, along with how approaching the solution in this way will address the issues raised in this chapter. The proposed solution will begin with a clear definition of the scope of non-conformance including what concepts are a part of this scope and what terms will be used throughout the thesis.

The non-conformance scope will then be broken down into a format that can be worked with, a structure that can be maintained such that knowledge about non-conformance instances can be practically derived. These breakdowns are presented such as the initial scope of non-conformance in a concept map that describes the application of rule sets used in detection and the specific detected non-conformance instance.

Non-conformance detection concepts will then be formalised generically in the form of a UML diagram that brings together the concepts and attributes covered in the chapter into a clear and applicable methodology. Techniques for applying this methodology will then be discussed, presenting the ideal way in which the proposed approach can practically be applied to a target process to detect non-conformance.

3.3.2 Implementation and testing of the proposed approach

The effectiveness of the proposed approach to address these issues was will be illustrated in a couple of different scenarios. First, a detailed description of the design and workings of a simulation environment developed to test the proposed approach will be given. This simulation environment grants the user the capability to define the processes to be simulated and then the rules that could be applied to the process to test the conformance of the enactment data. Second, an implementation involving a real world online process environment will be presented. This process has two stages that will be covered, which will show how the use of the proposed approach can be used to make practical and measureable improvements to the target process. Stage one is a preliminary non-conformance detection stage, and afterwards a secondary stage when improvements resulting from the detected non-conformance have been implemented into the process are covered.

3.3.2.1 Simulation Tool

The initial testing of the proposed approach was conducted via a tool implemented in C# .NET. The approach and its implementation are presented in Thompson and Torabi (2007) [9]. The tool facilitated the definition and simulation of different types of processes, which could have rules specified and applied to the simulated enactments. The application presented an interface with which rules could be specified that were translated into SQL queries that checked the enactment data stored in a relational database. The detail of this implementation is covered in greater depth in chapter 5.

The main benefit of this testing was that it confirmed that the approach could be applied to any process, as long as the enactment data could be structured and stored in a consistent way. The approach also facilitated on-the-fly detection, as the simulation environment enabled rule checking as enactment data was simulated and recorded.

The simulation tool also confirmed that the amount of effort to define, apply, remove and modify rules to and from the process could be a fast and straightforward task. This is especially true if a tool such as the one constructed existed for the process, but the application of rules in a manual sense is also not too burdensome, as discovered in the second implementation covered in chapter 6.

The focus of this simulation tool was primarily to confirm that the proposed approach was capable of detecting non-conformance instances in a practical way, and could do so while addressing some of the aforementioned issues. But since this was the primary focus of the first implementation, issues such as the structuring of non-conformance data to facilitate severity and recommendations or minimise interference was also not a consideration in this implementation.

3.3.2.2 Online implementation

Once the approach had been confirmed as viable after the first round of implementation and evaluation, a real world process was required to further test the practical viability of the proposed approach. This will be covered in chapter 6.

Since non-conformance detection is essentially a process improvement initiative, another objective of this implementation was to confirm that the process could actually be improved by evaluating any improvements that were made to the process as a direct result of any non-conformance that was detected.

It is important therefore, that the burden of process enactment data acquisition be as small as possible in this implementation. The focus should be on the non-conformance detection, not the collection of enactment data.

Chapter 6 will present the details of this online real world implementation and how this confirms that the issues presented in this chapter can be resolved using the proposed approach to detect non-conformance. Importantly, also shown in this implementation is how the process can be measurably improved by making improvement changes only based upon the non-conformance instances that were actually detected. The improvement potential of the proposed approach will be shown by presenting the findings from two phases of the implementation: the initial standard process phase and the second improved process phase.

The findings from this implementation and their relevance to the proposed approach have been presented previously in the proceedings listed in both Thompson and Torabi (2010) [10] and also Thompson and Torabi (2009) [13].

3.3.3. Evaluation of the approach

A chapter on evaluating the proposed approach in regard to its effectiveness as perceived by the implementation chapters is then presented. Here, the success of the proposed approach in detecting non-conformance will be discussed as well as an analysis on how well the approach was able to address the issues through the testing and validation discussed in the implementation chapters. The advantages and disadvantages of using the proposed approach will then be discussed, followed by a concluding chapter that will summarise all the work presented in this thesis.

3.4 Conclusion

In this chapter, we have explored the major issues currently facing the challenge of process non-conformance detection, why these issues are important and the potential benefits that could be gained by resolving them. A brief overview of the proposed solution was given, which will be covered in detail in the following chapter 4. Finally a plan for presenting the proposed approach was delivered, with an overview on how this approach would be tested and evaluated using the simulation implementation to be presented in chapter 6, the online process case study in chapter 7 and the approach evaluation in chapter 8. In chapters 4 and 5, the approach will be described in detail. This will include exactly how the approach works and how it resolves the issues covered in this chapter, followed by a guide on how this approach can be practically applied to a process alongside a logical presentation of why this approach works and the benefits it yields.

References

[1] A. Borgida, T. Murata, Tolerating exceptions in workflows: a unified framework for data and processes, ACM SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99, Volume 24 Issue 2, ACM Press, March 1999.

[2] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM)**,** Volume 7 Issue 3, July 1998.

[3] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[4] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[5] G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[6] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[7] I. Sommerville, “Software Engineering”, 8th edition, Addison Wesley, 2007, pp. 25 - 27.

[8] S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

[9] S. Thompson, T. Torabi, P. Joshi, A Framework to Detect Deviations During Process Enactment, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

[10] S. Thompson, T. Torabi, An Observational Approach to Practical Process Non-Conformance Detection, Journal of Information Technology Review, ISSN 0976-2922, 2010.

[11] G. Cugola, E. Di Nitto, A. Fuggetta, C. Ghezzi, A framework for formalizing inconsistencies and deviations in human-centered systems, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 5 Issue 3, ACM Press, July 1996.

[12] A. Borgida, T. Murata, Tolerating exceptions in workflows: a unified framework for data and processes, ACM SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99, Volume 24 Issue 2, ACM Press, March 1999.

[13] S. Thompson and T. Torabi, An Observational Approach to Practical Process Non-Conformance Detection, The Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), London, England, August 4 – 6, 2009.

[14] N. Zazworka, V.R. Basili, F. Shull, Tool Supported Detection and Judgment of Nonconformance in Process Execution, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, October 2009.

[15] S. Sørumgård. Verification of Process Conformance in Empirical Studies of Software Development. Ph.D. thesis, Norwegian University of Science and Technology, 1997.

[16] S.U. Hani, Impact of process improvement on software development predictions, for measuring software development project's performance benefits, Proceedings of the 7th International Conference on Frontiers of Information Technology, Abbottabad, Pakistan, December 2009.

Chapter 4 – Proposed Solution - GARDEN

In chapter 3, the main issues concerning the research in this thesis were presented. This chapter describes in detail the solution developed to address these issues with the goal of reliably detecting non-conformance instances in processes in a commercially practical way. As was covered in chapters two and three, the existing literature in the domain of process non-conformance detection seems to overlook certain issues that prevent those solutions from being practical and attractive enough to implement.

The solution presented in this chapter is designed specifically to address these issues in order to create a process non-conformance detection methodology that is as practical to adopt as possible. This solution is designed to:

- successfully detect non-conformance between a process enactment and its associated specification;
- be applicable to as wide a range of processes as possible;
- provide a methodology capable of detecting non-conformance instance types that are anticipated as well as unanticipated;
- minimise the time from non-conformance occurrence and non-conformance detection;
- minimise the cost and effort required to implement the solution;
- be flexible enough to evolve with a changing process and make changes to the implemented solution as easy and seamless as possible;
- minimise the level of interference the solution has with the existing process, including minimising the amount of resources that may need to be diverted from the process itself in order to implement the solution;
- provide a basis to assist and support the decision-making process on what remedial actions to take after non-conformance has been detected.

4.1 Introduction

The purpose of this chapter is to provide a detailed presentation of the proposed approach, focussing on how non-conformance is to be detected and how it can be detected in such a way that resolves the issues that were discussed in chapter 3. The concepts of the approach are presented and discussed and how they can be used to detect non-conformance is covered in this chapter.

The proposed approach has been given the name "GARDEN", to which it will be referred hereafter. In naming the approach, I have taken the same line a lot of researchers use when naming their work, which is to frame a sentence that describes the approach from which an acronym can awkwardly be derived. So, the acronym "GARDEN" stems from "Generic Application of Rule-sets to DEtect Non-conformance".

In presenting this approach, a brief overview of GARDEN is given first. This section provides explicit detail on the proposed approach. The scope of non-conformance is then discussed. The concepts and attributes relating to non-conformance are covered and represented in a concept map. These concepts are then broken apart into a consistent structure of rule sets and detected instances that can be used to define how non-conformance will be detected and what knowledge can be gained from this.

Following this, the loose structure of the proposed approach is then formalised in a UML representation. The concepts and attributes mentioned in the specification are covered again with a focus on how the formal approach handles them.

The methodology is then covered, describing the technique from start to finish in terms of how the proposed approach can be applied to a target process in order to detect non-conformance.

Finally, the chapter is concluded and is followed by two chapters on implementing and testing the proposed approach.

4.2 GARDEN Methodology Overview

Before we can begin to implement GARDEN and apply it to a process, there are some small prerequisites concerning the process that need to be checked. These prerequisites concern a) the basic and consistent structure of the process, i.e. whether the process can be analysed as a clear set of tasks or sections that may be represented as process activities; and b) the availability of process enactment data, i.e. a judgment as to whether there is enough quality process enactment data available for evaluation by GARDEN to make its implementation worthwhile. These prerequisites are covered in sections 4.5.1 and 4.5.2 respectively in this chapter.

The GARDEN approach is primarily concerned with the definition and application of "rule sets", which are designed to evaluate process enactment data to determine whether or not non-conformance has occurred. These rule sets are designed to complement the process specification with the purpose of checking process enactment data to verify that the process enactments are adhering to it. Rule sets are comprised of one or more "rules", which evaluate specific parts of the process enactment data and return a Boolean value to the rule set. Rules are covered in section 4.3.3 of this chapter and rule sets are covered in section 4.3.4.

When the GARDEN approach detects non-conformance, we want it to store as much information as possible about each instance that occurs. This provides a better medium in assisting process improvement, helps provide a better understanding of what non-conformance types are occurring and why, and assists in making the most informed decisions possible when considering what actions to take once non-conformance has been detected.

Essentially, the GARDEN methodology involves:

- the analysis of a process specification to gauge how a process is supposed to be enacted;
- the design and application of rules and rule sets to ensure process enactments are being carried out in accordance with the specification;
- evaluation of process enactment data using the defined rule sets to determine whether or not non-conformance has occurred;
- presentation of detected non-conformance instances in such a way that provides maximum possible knowledge and assistance in how to respond once non-conformance detection has occurred.

Additionally to these goals, the solution is designed to address the issues which have not been addressed by previous approaches and to make this approach as practically adoptable as possible. As the detailed explanations of the different facets of this approach will show, GARDEN can be safely implemented without interfering with the existing target process and the effort involved in application is designed to be completely at the discretion of the person in charge of implementation.

4.3 GARDEN Concepts and Specification

In this section, the concepts relating to the specification of GARDEN are discussed. The scope of what non-conformance is considered to be is covered along with all the attributes that are associated with it, in order to maximise knowledge gain. These attributes are then refined into two distinct entities, the rule sets used to detect the non-conformance and the instances of detected non-conformance themselves. The relationship between these entities facilitates how the necessary knowledge can be gained using the proposed approach. Individual rules themselves are covered first, before their adaptation into rule sets is presented. Finally, the specific instances are specified along with the attributes associated with them are presented at the end of this section.

4.3.1 Scope of Non-Conformance

In chapter 2 the concept of non-conformance was defined. Here, the concept of non-conformance is described in the scope of the proposed approach. The purpose of this is to be explicit about exactly what non-conformance is and the knowledge we seek to capture when it is detected. Extra care has been taken to be explicit about the meaning of the terms and concepts referred to in this solution in order to address the issue of clarity covered in chapter 3 section 3.1.1. A concept map depicts the scope of non-conformance in Figure 3.

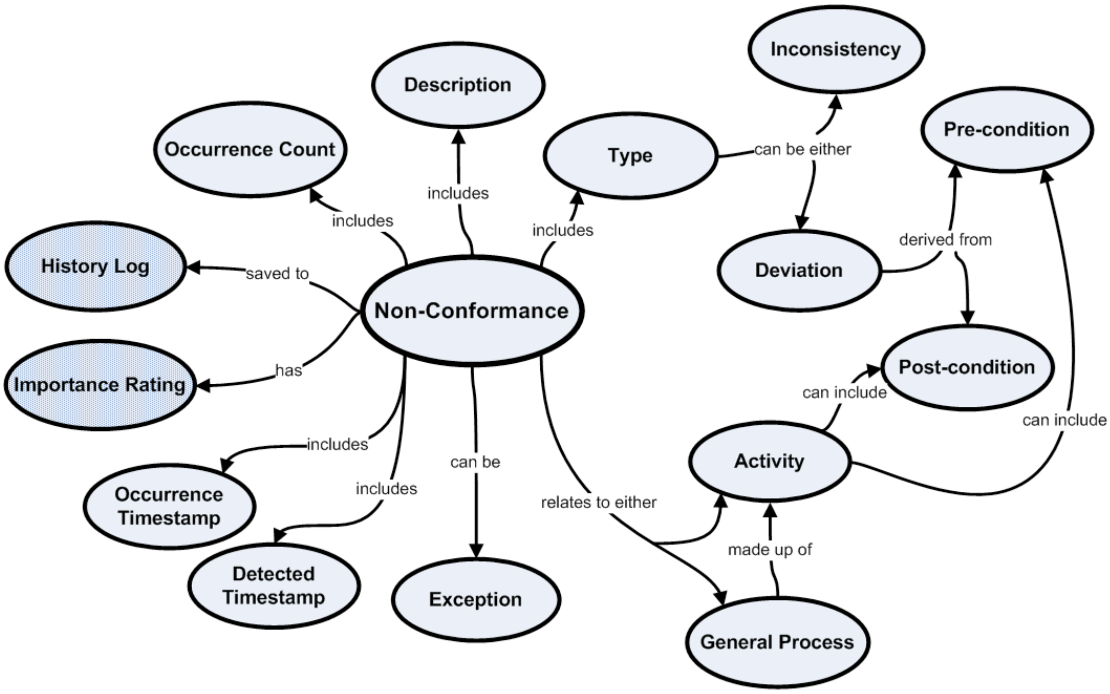


Figure 3: Non-conformance concept map

In previous work, the specifics of what constituted non-conformance, or a deviation depending on the approach were not detailed, such as in Kabbaj et al. [3], Cîmpan and Oquendo [7] and Huo et al. [15]. The closest the literature goes to specifically describing what non-conformance is, how it should be considered and what to look for when attempting to detect it, was given by Cugola et al. [6] in their paper on formalizing such concepts. It is for this reason that the specifics of non-conformance detection are presented in detail in this chapter.

The entities depicted in Figure 3 represent sections of knowledge about each instance of non-conformance that would ideally be extrapolated when it is detected. Deriving this knowledge from the detected non-conformance instances will provide the information required about the non-conformance to provide the best chance at process improvement as a result of the non-conformance being detected. Each node in the concept map is described briefly here, and in greater detail later in this section.

4.3.1.1 Description

This is a plain English description of what specifically has not conformed. This attribute is associated with the rule sets (explained in section 4.3.3) rather than any specific detected instance of non-conformance as this will be defined before the implementation is actually run against an enacted process and non-conformance instances are detected.

There is no formal structure to this attribute, but the description should mention the area of the process that was checked and exactly what the enactment data is being checked for.

For example, if the process asks for a person's age as an input, then the description might be something like "verify correct age input for activity *x*".

4.3.1.2 Detected Timestamp

The detected timestamp attribute is directly associated with a specific instance of detected non-conformance and it reflects the exact time and date the non-conformance was detected. It does not reflect the time and date of when the non-conformance actually occurred, only the time it was detected.

4.3.1.3 Occurrence Timestamp

If the process enactment data available indicates the time and date of when a process activity was carried out, then we can record this time as the timestamp indicating when the non-conformance was believed to have occurred.

4.3.1.4 Occurrence Count

Implementation of the proposed approach entails the application of one or many modular rule sets to a process. These rule sets are discussed further in section 4.3.3. Every time a rule set detects an instance of non-conformance, it needs to be logged and stored somewhere. The occurrence count attribute indicates the number of times that non-conformance has occurred and been detected from a single specific rule set.

This could be important information as it directly indicates when a certain type of non-conformance is repeating itself over time, which then implies that something requires addressing. This attribute type and its relationship with non-conformance have not been discussed in the available literature on non-conformance detection.

4.3.1.5 Type

The type attribute relates to the rule sets, and tells us that the rule set is being applied as either a pre-condition, post-condition or consistency rule set. As covered in chapter 2, non-conformance can be classified as being one of two distinct types: deviations and inconsistencies [6].

Deviations in the context of this solution relate to an instance of non-conformance that has been detected during the transition to or from an activity. Transition is governed via the use of pre-conditions and post-conditions in the proposed approach. Therefore, a deviation is a non-conformance instance derived from a rule set that has been applied to a process activity as either a pre-condition or a post-condition.

Inconsistencies refer to any non-conformance instances that were detected relating to either the general process as a whole, or in the enactment of one of the process activities. A rule set that has been labelled as type "consistency" will yield inconsistency type non-conformance instances when they are detected.

4.3.1.6 Exception

This attribute also relates to the rule set used to detect the non-conformance instances. The idea behind exception is that it is a "tolerable" version of non-conformance. It may be something that ideally should not occur, but we choose to acknowledge that it may sometimes occur and check for it nonetheless. We can label rule sets as exceptions if they are used to check the process for this kind of non-conformance. This attribute is derived from the "tolerable" types that were presented in Cugola et al. [16], where "deviations" that were considered not to force the process to enter an unsafe state were tolerated and allowed to occur.

4.3.1.7 History Log and Importance Rating

These two nodes on the concept map are shaded slightly darker than the others because in the context of this research they are considered extended attributes. The purpose of the importance rating is that there be some way of determining how severe non-conformance might be if it occurs. This attribute could aid in determining how severe any potential consequences may be as a result of the non-conformance occurring. It could also aid in prioritising remedial action addressing the non-conformance instances that have occurred. These concepts are more of an afterthought for consideration after non-conformance has already been detected and such considerations have not yet been addressed in the literature, as pointed out as an issue of concern in chapter 3, section 3.1.7.

The determination of what remedial action to take after non-conformance has occurred is also a challenge needing to be addressed in this area. This could be decided subjectively from experience, but making available some extra information to help with this decision could be very beneficial. Once non-conformance is detected, it needs to be logged so that it can be addressed at some future time. Many of the values and properties of the instance of non-conformance can be derived from the rule set it was derived from, however there will be some extra attributes relative to the specific detected instance itself that are beneficial to log. This is covered in greater detail in section 4.3.4 where the actual detected instance is discussed.

4.3.2 Why the Rule Set Approach?

The approach described in this chapter makes use of "rule sets", which are essentially groups of rules, each of which evaluate specific parts of the process enactment data to detect whether or not non-conformance has occurred. Each rule (covered further in section 4.3.3) checks a specific part of the process enactment data and returns a Boolean value indicating whether or not the enactment data satisfies the rule. The rule set uses these values to determine if non-conformance has taken place.

The use of rule sets enables us to detect non-conformance by complementing the process specification without altering it or interfering with it in any way. This is important to resolve the issue of process interference as discussed in chapter 3 section 3.1.2 and also means we glean the benefits of keeping the approach observational as discussed in 3.1.6.

To successfully implement a rule set, we need to analyse a part of the process specification to determine what the process is supposed to do at a particular juncture. Then we simply define a rule set to check the relevant part of the process enactment data to ensure that the process enactment has conformed to the specification. There is complete autonomy in how the rule sets are implemented and how they function. This helps satisfy the issues from chapter 3 focussing on maintaining a flexible approach (covered in 3.1.4) and also minimising interference (covered in 3.1.2) in the existing process. It also means that the approach can be applied more generically, an issue covered in section 3.1.5, because as long as the existing process specification can be clearly understood, we can build rule sets to complement it.

The use of rule sets is based on the principle of object orientation. In object-oriented methodologies, the aim is to keep related methods encapsulated into an object that deals with a specific part of a system, application or process. Likewise with GARDEN, it makes sense to identify a specific part of the process that needs addressing and then create a set of rules to deal with it, rather than procedurally specifying rules ad-hoc.

Since GARDEN provides the ability to apply rule sets as needed, the approach can be tailored at the integrator's discretion, which addresses the issue of implementation effort covered in section 3.1.3. If there is only a certain area of the process that needs to be addressed for the purposes of non-conformance detection, then rule sets can be defined that address only that section. This is one of the key differences to the approaches discussed in chapter 2 such as the data mining approaches presented in Cook and Wolf [17] and Huo et al. [15] as well as the process support mechanisms in Cîmpan and Oquendo [7] and Cugola et al. [16].

The grouping of rules and their application as rule sets also plays an important role in supporting process evolution and maintenance of the approach, an issue covered in 3.1.4. Since each rule set is applied to a specific part of the process, only when that part of the process changes do we need to readdress the rule set. The rule sets which apply to unchanged parts of the process need not be reconsidered and the whole approach does not need to be re-implemented from scratch, which is a problem associated with some of the other approaches, namely Huo et al. [15], Cîmpan and Oquendo [7] and Cugola et al. [16]. As only affected rule sets need to be reconsidered when the process changes, this has a positive effect on reducing maintenance effort.

This differentiates GARDEN from other approaches such as Kabbaj et al. [3] which have rules specified to detect non-conformance from a pre-defined list. It is preferable to group them so they add meaning when they are actually used. This method also helps differentiate different non-conformance types modularly, whereas it may be difficult to classify them in approaches such as Cîmpan and Oquendo [7] or Huo et al. [15], while still offering greater flexibility to adapt to different non-conformance types in different processes rather than the pre-defined list offered in Kabbaj et al. [3].

Furthermore, as will be covered in more detail in section 4.3.4, additional data can be associated with rule sets that give us valuable information about what non-conformance is occurring. This includes information that may be used to aid in aspects of process improvement considered after non-conformance detection has occurred (to aid in addressing the issue discussed in 3.1.7), such as the application of severity values and possible remedial recommendations. For these reasons, a rule set approach to non-conformance detection promises to be the best fit to successfully and practically detect anticipated and unanticipated non-conformance instances, as well as to meet the goal of resolving the current issues covered in chapter 3.

4.3.3 The Function of Rules

The specification of rules to govern legal process behaviours is not a new innovation. Zisman et al. [4] present an XML based approach for specifying consistency rules for the overlapping content of distributed web documents, an extension to their approach to detect inconsistencies among distributed documents based on pre-defined rules [5]. Kabbaj et al. [3] use a rule-based approach in a similar piece of research to our own, however their rules focus on the determination of whether or not a detected "deviation" is tolerable rather than whether or not it has occurred. To put it simply, a rule exists to check a portion of the process enactment data and will return a Boolean value that tells us whether that portion of data conformed to the rule or not. It does not matter how the rule itself is specified or defined, but it must be capable of testing whether or not the process enactment data complies with the condition it specifies, and then convey the result of this test in a Boolean form.

Rules are used to complement the process specification. For example, if the process specification stipulates that a "character level" must be input by the user at some point, we can define rules that can test the actual character level inputs to check whether they meet certain constraints we may have decided to set for this section of the process. Such a rule might be to check that this input was supplied in a numeric form:

Example rule:

Character level must be of numeric value

The rule is simple. A process actor must specify the level of the character numerically. If they do not, this rule will return false, indicating that an instance of non-conformance has been detected at the point in the process where this rule was implemented.

The actual implementation of this rule will be different depending on the environment of the process in question and the format of how the enactment data is stored. It could be coded in some application logic or even stored as a relational database query.

Table 3 represents how this rule might hypothetically be used to validate some event data from several enactments of the same process.

Enactment Input Value	Rule	Return Value
Level 13	Character level is numeric	False
#13	Character level is numeric	False
13	Character level is numeric	True

Table 3: Example Rule Application

Rules are good for checking specific conditions on very specific parts of the process enactment data. In order to check for related conditions more comprehensively whilst also maintaining modularity in the approach, rules are grouped into *rule sets*, which represent the full mechanism of checking for non-conformance instances.

Rules cannot be applied independently to the process enactment data using the proposed approach; they must be part of a defined rule set.

4.3.4 Rule Sets

Rule sets are the backbone of the GARDEN approach. The application of the rules contained within the rule set is the mechanism used to detect each instance of non-conformance. Like a rule, a rule set will also return a Boolean value. If a rule set returns true, it means that the process enactment data has complied with the rule set and no further action is taken. If a rule set returns false, the enactment data has not complied with the rule set and non-conformance has occurred, which is then logged.

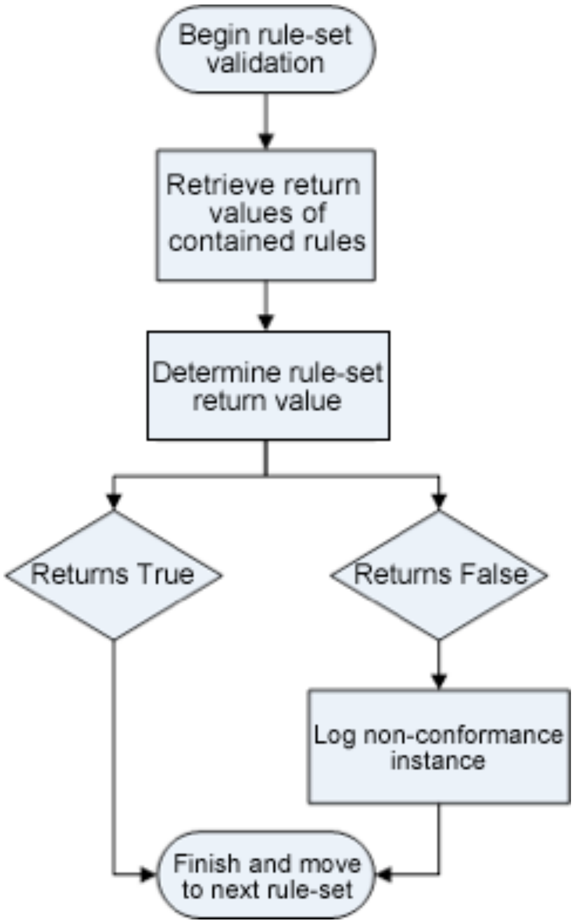


Figure 4: Rule set validation flow

The mechanism used to determine the return value of a rule set depends on the return values from the rules contained in the rule set and is explained in section 4.3.3.5.

When defining a rule set, we are not just grouping related rules together to detect non-conformance. We are also defining where in the process we are looking for non-conformance, what the importance is of this type of non-conformance is and what kind of non-conformance will be detected using the given rule set.

A concept map showing the different attributes and concepts regarding rule sets is depicted in Figure 5.

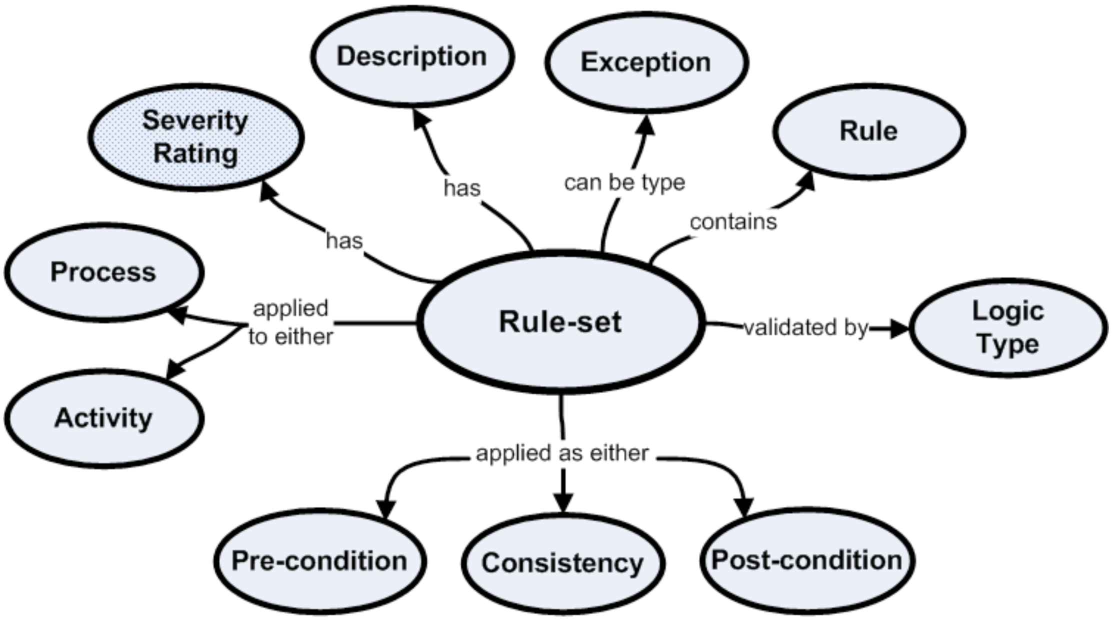


Figure 5: Rule set concept map

The rule set provides much of the information desired about non-conformance, as it is the mechanism used to detect non-conformance. These nodes in the concept map are described in greater detail in this section.

4.3.4.1 Description

This attribute is representative of the detail about non-conformance described in section 4.3.1.1. A rule set has a description that explains in plain English what part of the process is being checked for non-conformance and the type of non-conformance (if applicable) being checked for.

4.3.4.2 Application of Rule Sets

The rule set will also have an attribute specifying whether it is being applied to the process as a whole, or to a specific activity within the process and which activity this is. It is this attribute of the rule set that tells us where in the process we are looking for instances of non-conformance.

Once this has been specified, an attribute is also required that specifies how the rule set will be applied, this being either as a pre-condition, post-condition or consistency rule set. If the rule set is being applied to the general process, this value will default to "consistency". However, if the rule set is being applied to an activity, it may be applied as one of these three types. This is covered in more detail in section 4.5.6.

4.3.4.3 Severity Rating

As discussed in chapter 3 sections 3.1.7 and 3.2.2.7, an objective for this research is to define non-conformance in such a way that it facilitates the future initiatives to determine non-conformance severity and consequences. Since this attribute is not directly used in the proposed approach, it is shaded slightly differently to the other nodes in Figure 5. At present, the rule set may be given a severity rating of "minor", "average" or "major" to denote its relative importance. The default value is "average". The use of this kind of information to determine real severity values and possible recommendations is presented in Thompson and Torabi [8].

4.3.4.4 Exception

As mentioned in section 4.3.1.6, the exception attribute is used to denote when a particular instance of non-conformance is not ideal but regardless, is marked to be tolerated. The basis for the decision to include this attribute as part of the rule set make up was from Cîmpan and Oquendo's mention of "tolerable deviations" [7].

This attribute is distinct from the severity rating. By labelling a rule set as an exception, this does not mean that it inherits a minor severity rating or similar. It is used more as a mechanism to apply constraints that we may be hesitant to apply to the process as typical non-conformance for some reason, or even just non-conformant type behaviour that we may wish to acknowledge and ignore as a normal part of the process. The exception attribute enables us to achieve this.

4.3.4.5 Containment of rules

A rule set must contain at least one and may contain many rules. The rules contained within the rule set are in charge of checking the process enactment data and will return a Boolean value indicating whether or not the associated data has complied with the rule.

Most commonly in GARDEN, rule sets will have the logic-type of 'AND', meaning every rule in the rule set must return true in order for the rule set to return true. If one or more rules within the rule set return false, the rule set also returns false and non-conformance is logged. There are other types of logic-types that can be defined for the rule set which will affect the way in which the rules contained by the rule set determine the evaluation of the rule set itself. This is explained further in section 4.3.4.6.

An example of how a rule set may be set up is depicted in Table 4. Here, the rule set has been specified to validate data for a person's age as input. Several rules have been defined within the rule set to help validate the data. The logic-type of the rule set is 'AND', meaning every rule contained must return true in order for the rule set itself to return true and not log a non-conformance instance.

First, an example of how a rule set might check an enactment value that complies with the rule set is shown. Let this hypothetical value be **34**.

Rule	**Return Value**
Age input is numeric	True
Age input is >= 0	True
Age Input is < 120	True
Rule Set Return Value:	**True**

Table 4: Rule set example - check age input returning true

Now notice the outcome of changing this enactment value to an invalid **265**.

Rule	**Return Value**
Age input is numeric	True
Age input is >= 0	True
Age Input is < 120	False
Rule Set Return Value:	**False**

Table 5: Rule set example - check age input returning false

So the return value of the rule set itself is completely dependent upon the return values of the associated rules it contains. As previously mentioned in this section, a rule set that returns **true** indicates that all the rules within it have complied with the specification, so no further action is taken. A rule set that returns **false** indicates that the rules within the rule set did not validate when they were evaluated according to the logic set by the rule sets logic-type. A rule set that returns false also means that non-conformance has been detected, so it should then be logged so that it may be addressed at the appropriate time. The logic-type attribute of the rule set is covered in the next section 4.3.3.6

4.3.4.6 Logic-Type

The logic-type is an attribute of the rule set that tells us how it will be validated using the rules contained within. Every rule within the rule set will return a Boolean value. The logic-type tells the rule set how to evaluate the result of each rule within the rule set to determine whether the rule set returns true or false. The inclusion of this attribute to the rule set facilitates a variety of different ways to check for non-conformance at the discretion of the person specifying the rule sets. It becomes easier then, to specify ways in which a process should and should not be performed.

The existing process specification will define how actors *should* perform a process, not necessarily how it *should not*. Sommerville argues that predicting all the different ways things can go wrong is extremely difficult [2]. Nevertheless, it can still sometimes be important in certain instances to describe specific things that should not be done. Kabbaj et al. follow this line with their list of 30 odd non-conformance possibilities they cite [3].

There also may be different possible actions that an actor might choose to take in a given situation, each of which might be perfectly valid. In approaches such as Cugola et al. [6] and Huo et al. [15], it is not clear how these legitimate differences from enactment to enactment could be handled.

This problem is solved with the inclusion of the logic-type attribute for rule sets. The logic-type attribute will have one of four possible values that are based on digital logic:

- AND
- NOT
- OR
- XOR

4.3.3.6.1 The AND logic-type

The “AND” value is the default value for the logic-type attribute. Like its namesake in digital logic, it works in much the same way in respect to the Boolean values each rule will return to evaluate the rule set. A rule set with the logic-type attribute set to “AND” must have every rule it contains return **true** in order for the rule set itself to return true. An example of this type was given previously in Table 4, which is replicated below in Table 6 for the reader’s convenience.

Rule	**Return Value**
Age input is numeric	True
Age input is >= 0	True
Age Input is < 120	True
Rule Set Return Value:	**True**

Table 6: Rule set example with an “AND” logic-type

Again, as previously depicted in Table 5, if any of the rules within an “AND” logic-type rule set return false, then the rule set will also return false and non-conformance will be logged.

4.3.4.6.2 The NOT logic-type

If the logic-type attribute is set to the “NOT” value, then it works in the opposite way to the “AND” logic-type. Every rule in a “NOT” rule set must return **false** in order for the rule set to return true. An example similar to the one used in section 4.3.3.5 is shown in Table 7 to demonstrate this.

Rule	Return Value
Age input contains alpha characters	False
Age input is < 0	False
Age Input is > 120	False
Rule Set Return Value:	**True**

Table 7: Rule set example with a "NOT" logic-type

Rule sets that have the "NOT" logic-type are checking process enactment data for specific conditions with which they should not comply. To illustrate, the age example was used again in Table 7. In this previously used example, the enactment data had a hypothetical value of **34**, which was confirmed as having complied with the process specification by the default "should" rule set. However, if we wanted to instead apply a "NOT" logic-type rule set to check this data, we would define rules that stipulate conditions with which the input age should not comply.

We know that in an "AND" logic-type rule set that if one or more rules return false, then the rule set itself will also return false. However, the opposite behaviour occurs in a "NOT" logic-type rule set, being that if one or more rules contained within return **true**, then the rule set will return false.

4.3.4.6.3 The OR logic-type

This type is specified when at least one of the rules within the rule set should be satisfied for the rule set to return true and avoid non-conformance detection. Rule sets with the "OR" logic-type will not return false unless every rule contained within the rule set also returns false. One or more rules returning true will cause the rule set itself to also return true and for non-conformance to be avoided. An example of such a rule set is depicted in Table 8.

Rule	Return Value
Date of birth was given	False
Age value was given	True
Over 18 years old was given	True
Rule Set Return Value:	**True**

Table 8: Rule set example with a "OR" logic-type

This example is slightly different from the previous examples. Let us assume here that an actor has to input a value as part of the process that relates to age. The actor has a choice, however, from a few different but equally valid options. The actor may input a date-of-birth, an age value or simply state that he or she is over 18 years of age. To check for non-conformance, all we might care about is that at least one of these values was provided.

4.3.4.6.4 The XOR logic-type

The "XOR" value is similar to the "OR" value, but with one important difference also apparent in their digital logic counterparts. In an "XOR" logic-type rule set one and only one rule can return **true** for the rule set to also return true. An "XOR" rule set, therefore, can return false and record a non-conformance instance if either more than one rule within returns true, or if all rules within return false. An example is given in Table 9 of a possible "XOR" logic-type rule set specification.

Rule	Return Value
Date of birth was given	False
Age value was given	True
Over 18 years old was given	True
Rule Set Return Value:	**False**

Table 9: Rule set example with a "XOR" logic-type

4.3.4.6.5 Logic-Type Comparisons

From the examples and the descriptions covered in this section, a side-by-side representation of the different logic-type values available in the proposed approach is shown in Figure 6 and how the rule set would be validated to return true.

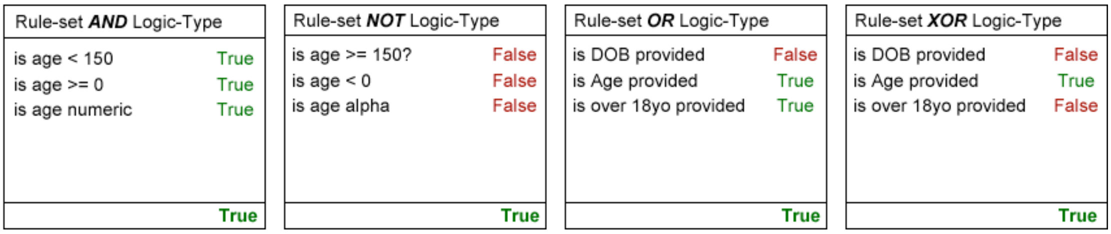


Figure 6: Different Logic-Type Rule Set Comparisons Returning True

Conversely, the same rule sets are shown in Figure 7 depicting the rules with different return values causing them to return false, which would constitute non-conformance.

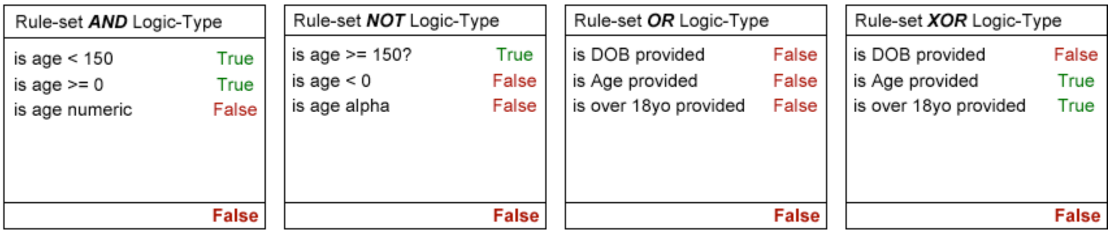


Figure 7: Different Logic-Type Rule Set Comparisons Returning False

4.3.4.7 Rule Set Review

The GARDEN approach provides the freedom to still specify ways in which the process should and also should not be carried out, without burdening them with actually predicting all the ways in which this could occur. This is something that has not been touched upon so far in the literature.

The rule set approach also facilitates the use of tolerable non-conformance in line with the idea presented in Cîmpan and Oquendo [7] in the form of exception types.

By use of a severity attribute associated with the rule sets, a mechanism for determining severity has been supplied, albeit in a preliminary capacity. This is distinct to the mechanism presented in Cîmpan and Oquendo [7] where "severity" was measured as a degree of difference of the enactment value purely from the expected value.

4.3.5 Detected Non-Conformance Instances

Once a rule set has been used to detect an instance of non-conformance, we must now think about what additional data we want to know about the specific instance itself.

The rule set provides most of the information concerning non-conformance, such as the description, the process section, the type etc… however there is additional valuable data that can be stored relating to the non-conformance instance that can be used to improve the process. The additional attributes that are logged with the non-conformance instance in GARDEN are depicted in Figure 8.

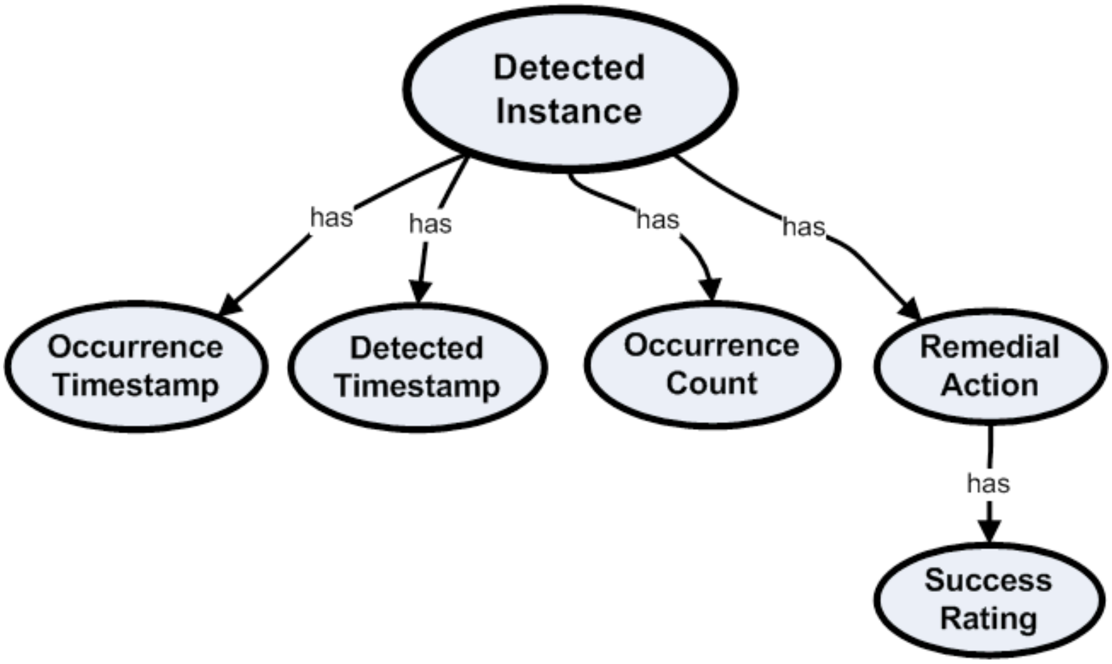


Figure 8: Specific non-conformance instance concept map

<u>4.3.5.1 Occurrence Count</u>

Since each specific instance of non-conformance is related to the defined rule set that was used to detect it, counting the occurrences of each instance of non-conformance for each defined rule set is a simple matter. It is important to do so because an occurrence count will highlight any particular rule set that is causing repeated non-conformance and can therefore indicate that the corresponding part of the process specification may need addressing. As will be shown later in the implementation chapters, the non-conformance logs over time can provide a valuable tool in addressing weak points in the process specification by indicating which types of non-conformance have a tendency to repeat themselves. The occurrence count is useful for indicating this in a plain manner, and is an attribute not included in other approaches such as Kabbaj et al. [3], Cîmpan and Oquendo [7] or Huo et al. [15].

4.3.5.2 Detection Timestamp

Recording the time and date of when the non-conformance was detected is an easy and logical step. The longer an instance of non-conformance has existed before being addressed, the greater the risk associated with its potential consequences and the smaller the potential for using the new information for improving the process. There are also added benefits, such as being able to tell the chronological order of non-conformance instances in a process.

4.3.5.3 Occurrence Timestamp

As covered in section 4.3.1.3, if the process enactment data can provide an indication of when the non-conformance actually occurred, then this data can be useful to obtain. Coupled with the detection timestamp, it can provide information on how long it has been since a) the non-conformance actually took place and b) the non-conformance was actually detected. Such information may be useful in making judgment calls as to what action to take to rectify any consequences, especially since time-to-detection is a key issue for this approach as flagged in chapter 3, section 3.1.8. It is possible that the time difference between when non-conformance actually occurred and the time that it was detected is significant. Having the detection timestamp and the occurrence timestamp helps to signify this, if it is an issue for the process.

4.3.5.4 Remedial Action Taken

This data is not logged when non-conformance occurs, but rather, optionally after any remedial action has been taken to rectify the non-conformance. The remedial action data is input simply as plain English and is not evaluated in any way other than manually. This data will have to be manually added after the non-conformance has somehow been addressed. The purpose of this attribute is to track the different kind of actions that are undertaken when different types of non-conformance occur. This data can hopefully come in useful in future if similar non-conformance instances occur as a reference to what was done in similar circumstances in the past, and contributes to the resolution of the same issue mentioned in chapter 3, section 3.1.7.

4.3.5.5 Success of Remedial Action

In addition to logging any remedial action taken for a non-conformance instance, it is also important to gauge how successful the action was in addressing the non-conformance. The input of this attribute value is also a manual task and performed optionally when the remedial action itself is input. The possible values of which this attribute could hold are presently a matter for future research on which work is currently being undertaken. An overview of the work regarding the implementation of this feature was presented in Thompson and Torabi [8], however the facility for storing some severity relevance value is provided in the GARDEN approach.

As with the remedial action, this data is only evaluated manually at this stage. The idea behind the rating attribute is that if similar non-conformance instances can be quantified with successful and unsuccessful ratings then a person involved in addressing an instance of non-conformance can deduce which remedial actions may have a high likelihood of success and which are best avoided.

4.4 Formalisation of the proposed approach

The approach described in this chapter is presented formally in a UML 2 diagram depicted in Figure 9. This diagram brings together all the concepts previously presented in the concept maps shown in this chapter along with their intended function. The formalised version of the proposed approach is discussed in this section. In the diagram, it is shown how all the previously described concepts fit together and how they are applied to the process in order to detect non-conformance.

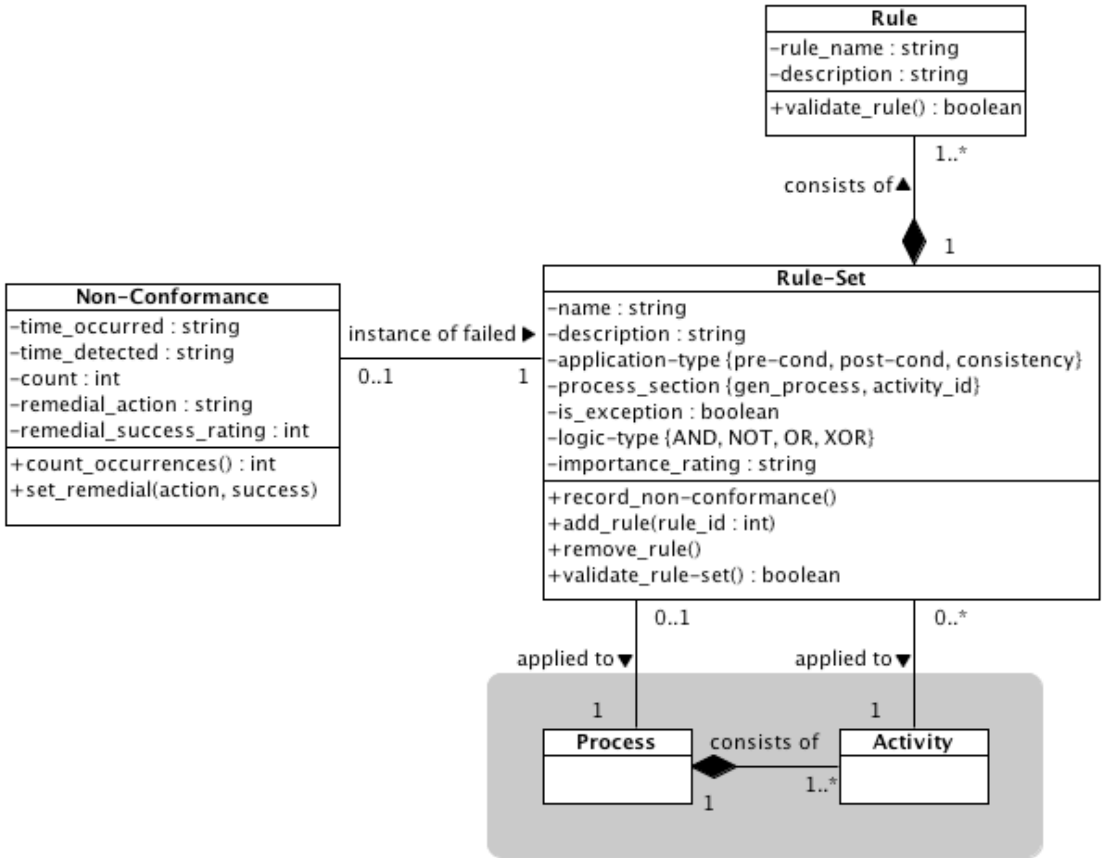


Figure 9: Formal UML representation of proposed approach

4.4.1 Model Notes

In section 4.3.1 of this chapter, all the attributes and concepts that non-conformance consists of in the scope of this research were covered. The formal model depicted in Figure 9 shows that some of these attributes have been distributed over the non-conformance and the rule set object. The reason for this is because it makes more sense to have these attributes relating to either the rule set that detected the instance or the actual instance itself.

All values relating to a specific instance of non-conformance can be derived from the relationship between the rule set and the non-conformance objects depicted in Figure 9.

4.4.2 Process Structure Objects

Before the proposed approach can be applied, we must confirm that the target process is specified with some kind of structure that includes the use of activities or entities that can be considered activities. The classes encapsulated by the grey shaded area depict this in Figure 9. Since the proposed approach involves the application of rule sets to either the general process or a specific activity, it is paramount that this kind of structure be distinguishable in a process being considered as a candidate for GARDEN.

4.4.3 Rule Object

As shown in Figure 9, the Rule class has a composite relationship to the Rule Set class, which indicates that a Rule Set object will need to exist before a Rule can be defined as being part of it.

The implementation of Rule objects will be completely dependent upon the process, the environment in which it runs and the form and structure of the available enactment data.

In all good development practice it is wise to document what has been implemented and why. Therefore, the Rule should include a name and a description of the specifics of what it intends to check, but it only has a single function: to validate some enactment data. This single piece of validation performed by the rule should be documented to show what it is for and what it is doing, which is what the name and description are used for.

The implementation of the validate_rule() function will check some process enactment data for some specific condition, depending on the intention of the rule. This function has a Boolean return value type. If the enactment data complies with the rule, then validate_rule() will return true. If it does not, then validate_rule() will return false.

4.4.4 Rule Set Object

The primary function of the Rule Set class is to validate every Rule object it contains. This is represented by the validate_rule-set() function, which returns a Boolean value and ultimately determines whether or not non-conformance has or has not occurred. This determination is however affected by some of the attributes associated with the object.

First, a *name* and *description* is included. Like the similar rule object attributes, these attributes have no functional purpose except to provide a means of communicating what kind of non-conformance is being tested for by the rule set. Only the name and description of the rule set object is used to convey the purpose of the whole rule set, not the rules it contains.

The *process_section* attribute conveys what part of the process the rule set is checking for non-conformance. This must either be the “general process”, which indicates the rule set applies to the emergent properties of the process as a whole, or uniquely identifies a process activity to which the rule set applies.

The *application-type* attribute is limited to having one of the values of pre-condition, post-condition or consistency. If the *process_section* attribute is set to general process, then the value of *application-type* is ignored and defaults to “consistency”. If the rule set is applied to a process activity, then the value of the *application-type* attribute tells us how the rule set is being applied.

- If the value of *type* is “consistency”, then the rule set is checking the process enactment data for inconsistencies relating to the general values of the activity.
- If the value of *type* is “pre-condition”, then the rule set is checking the process enactment data for deviations relating to the commencing of the associated activity.
- If the value of *type* is “post-condition”, then the rule set is checking the process enactment data for deviations relating to the conclusion of the associated activity.

The final two attributes in the Rule Set class have a Boolean value. The *is_exception* attribute denotes whether or not the rule set is checking for non-conformance that should be considered as an exception rather than a standard non-conformant instance if its value is “true”.

The *logic-type* attribute represents how the Rule Set in question should be validated. There are four possible values that the *logic-type* attribute may hold, being AND, NOT, OR and XOR with the default value being AND. The value of this attribute directly affects the means by which the rule set is validated and whether non-conformance has occurred or not. The validation methods for each *logic-type* have been covered in section 4.3.3.6.

The function validate_rule-set() determines whether or not non-conformance was detected. It accomplishes this by calling the validate_rule() function from every Rule it is comprised of and evaluating the return value from each one. If validate_rule-set() returns true, then non-conformance has not occurred as the enactment data is considered to be validated.

If validate_rule-set() returns false, then it will call the record_non-conformance() function. The purpose of the record_non-conformance() function is simply to record the instance details of the non-conformance instance in some kind of repository. These details will depend on the process and what kind of data is available, however the attributes concerning the specific non-conformance instance are discussed further in section 4.4.5.

An example of the kind of data structure appropriate for recording a non-conformance instance is depicted as a relational database entity in Figure 10. This structure is not mandatory, however it is an appropriate schema of how non-conformance instance could be stored. This is covered further in chapter 5.

Non-Conformance_Instance		
PK	**InstanceID**	INT NOT NULL AUTO_INCREMENT
	Time_Detected	DATETIME
	Time_Occurred	DATETIME
	Remedial_Action	TEXT
	Remedial_Success	ENUM(Poor, Average, Good)
FK	**Rule-SetID**	INT NOT NULL

Figure 10: Non-conformance storage structure

4.4.5 Non-Conformance Object

The Non-Conformance class relates directly to actual instances that were detected using the rule sets as depicted in Figure 9. This object manages the handling of detected and recorded non-conformance instances.

The *count* attribute is representative of how many non-conformance instances have been detected and recorded by the same rule set. This attribute is not modelled by the schema in Figure 10 because it is set in the Non-Conformance object via the count_occurrences() function. The count_occurrences() function simply counts the number of non-conformance instances which have been detected by each rule set and sets the *count* attribute for the Non-Conformance object.

There are also two timestamp attributes included in the Non-Conformance class. The *time_occurred* attribute holds the value of the date and time that the non-conformance was believed to have occurred. If this value cannot be determined, the attribute may be given a null value to represent this. The *time_detected* attribute holds the value of when the non-conformance instance was actually detected. The difference between these two values conveys the amount of time between occurrence and detection, and also the amount of elapsed time if nothing has been done to address the non-conformance since it occurred or was detected.

The final two attributes in the Non-Conformance class are the *remedial_action* and *remedial_success_rating* attributes. The values of these attributes are given after non-conformance has been addressed somehow, rather than when it occurs or is detected. These attributes respectively are intended to store what was done to address the non-conformance that was detected (if anything) and how successful that action was at mitigating any consequences. The inclusion of these attributes provides valuable information when determining what to do next after non-conformance has occurred, which is a goal for this methodology as explained in chapter 3, section 3.1.7.

4.5 Methodology

The aim of this section is to describe how GARDEN can be applied to a target process. This begins with an assessment of process suitability, including how to evaluate the target processes structure and the availability of its enactment data. Once the target process has been satisfactorily identified as one to which the proposed approach may be applied, the basic preparations for implementation such as the setting up of a data structure to store non-conformance instances or the way the rule sets will be implemented in code is then discussed.

The manner of applying the rule sets to the process itself is then presented, including how to apply rule sets to activities, the general process and how to specify them as being consistency rule sets or conditional. Finally, the handling of the correct sequencing of process activities is covered. As a general overview, this section is presented as a prelude to chapter 5, which covers the practical implementation and design of the GARDEN approach and the issues that may be faced when attempting to apply it to a process.

4.5.1 Process Structure

One of the key aims of GARDEN is to keep the solution generic, so that it may be applied to different processes in different environments. Different processes are structured differently and are enacted in different environments, so there may be a great deal of difference from one process specification to the next. Although this type of generic applicability may conceivably be available in some other approaches in this domain, most are tailored specifically to one type of process (usually the software process) and are presented as such. The strength of GARDEN is that it may be applied to any process as long as the process fits the application requirements.

In order to implement GARDEN successfully, we must make a few assumptions about the target process and the characteristics it will possess. First, there must exist a process specification and there must be a degree of structure to it. The specification should describe how the process should be enacted to a level of detail such that it is possible to observe whether or not an enactment has conformed to it.

Vague specifications such as the process model described by Bogia and Kaplan [1] would generally by unsuitable to the proposed approach, because at the conclusion of an enactment we are only really able to tell whether or not it has achieved the process goals, not if it conformed to a structured and detailed specification.

As long as there is a specification present in some form, it is also desirable that there be separate and distinguishable activities involved in making up the process. Every process should be made up of at least one activity. The modularisation of the process provides multiple benefits. It facilitates the application of the proposed approach to limited or specific parts of the process if desired, it facilitates the approach to being applied more comprehensively and it allows localizing detected non-conformance to exactly where in the process they occurred.

Finally, it is helpful if specific *actors* are attributed to each activity performed within the process. It does not matter if the actors are automated or human, but there should be a responsible actor specified for each activity in the process.

The assumed process structure is depicted in a formal UML model given in Figure 11.

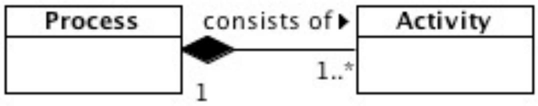


Figure 11: Basic Process Structure

4.5.2 Availability of Process Enactment Data

The issue of the availability of process enactment data has been previously discussed in chapters 2 and 3. The acquisition of process enactment data is not the focus of this research and as such it is not covered in this chapter, however it is required for GARDEN to be implemented effectively.

It is not particularly important how the process enactment data is supplied, whether it is in file logs or in a relational database somehow – each process may supply data from its enactments in different ways. The important criteria here is that the data made available for the implementation of the proposed approach is in an abundant enough quality to facilitate non-conformance detection. Also, the data should be supplied in a consistent way.

Since the application of non-conformance detecting rules will be done in a consistent way, the available data must have at least some form of consistent structure. This makes the definition and application of rules and rule sets a much easier task.

4.5.3 Preparing the Implementation

One of the goals for the proposed approach, as stated in sections 3.1.3 and 3.2.2.3, was that the effort required be as minimal as possible. Now that the process has been identified as one that is suitable for the proposed approach to be applied to it, some minimal preparations must still be made before the implementation can begin.

The first and arguably more difficult of the two decisions that must be made is how the rule sets will be functionally applied. This could be in code or via the use of a tool much like the environment presented in chapter 5. Ultimately, the most practical way to apply the rule sets will be dependent on the process.

Once the mechanism for how the enactment data will actually be checked has been decided upon, there needs to be a backend data structure to store the instances of non-conformance that are detected. Again, this can be done in various ways such as log files or a database. However, if the non-conformance instances are intended on being stored, a schema such as the one depicted in Figure 10 should be followed.

4.5.4 Application of Rule Sets to a Process

Now that the workings of rule sets and the way they will be implemented have been made clear, the ways they can be applied to a process are discussed. There are two main ways rule sets can be applied to the process, firstly to the general process as a whole and secondly to one of its specific defined activities. The application of rule sets to each is covered in this section.

4.5.5 General process and its emergent properties

Processes, like other types of systems will have *emergent properties* [9] [10] that are representative of the process as a whole. These properties and characteristics can only be understood and evaluated by regarding the process as a whole entity rather than the individual activities of which it is comprised. Unlike the other non-conformance detection approaches in this area of research, GARDEN also attempts to detect non-conformance relating to these emergent properties that reflect upon the entire process and not just the individual aspects of detail that make up the process.

Therefore, provision has been made to define rule sets that exist to check some of these properties in an attempt to confirm that the process as a whole entity has conformed to its general specification. General process rule sets will always be applied as *consistency* type rule sets. Pre-condition and post-condition type rule sets refer to transition and are applicable only to the processes' individual activities and are discussed in section 4.5.6.2. Since general process rule sets apply to the entire process, there is no transition applicable and so only consistency type rule sets are required.

As with any rule set definition, there is complete freedom to specify rules to check any kind of whole process property as required, however to illustrate the kind of rules that might be commonly applied as general process rule sets, some examples are as follows:

- time duration – did the process complete too quickly or take too long?
- time date – was the process begun or concluded at a particular time when it should not have been?
- activity count – were too many or too little activities executed during the process enactment?
- activity absences – were any required activities not executed during the process that should have been?

4.5.6 Activities and pre and post conditions

Activities are the separate individual tasks that make up the process. Rule sets can be applied to activities in one of three ways: consistency, pre-condition or post-condition type rule sets. These three ways are depicted in Figure 12.

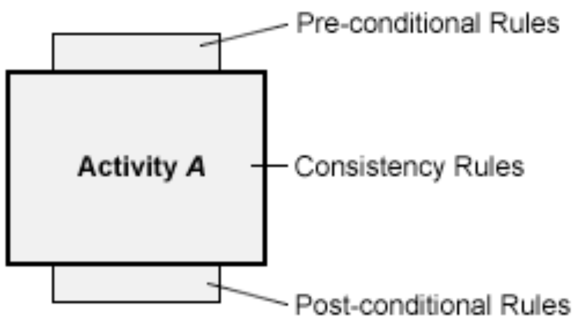


Figure 12: Assigning rule sets to process activities

<u>4.5.6.1 Consistency Rule Sets</u>

Much like the rule sets applied to the general process, consistency rule sets that check general attributes, values and properties can be applied to each activity. There is unlimited freedom afforded to the application of rule sets to activities in a consistency type form. Any part of the enactment data relating to the activity may be checked by a rule set. However, to illustrate the kinds of rule sets that may apply generically to process activities, some example rule set checks may be applied as follows:

- time duration – a rule set to define the minimum and/or maximum amount of time it should take to carry out the activity.
- time date – are there any specific dates or times when the activity should or should not be performed?
- bad actor – are there certain actors with specific skill sets who should or should not perform the activity?
- too many repetitions – has the activity been repeated too many times during the enactment of the process?
- high exception count – if exception-type rule sets are set for the activity, is there a maximum set on how many exceptions can occur before it constitutes being non-conformance?

If non-conformance is detected as a result of a consistency type rule set, then the non-conformance is referred to in this approach as an *inconsistency*. Inconsistencies are recognised as the same, whether they were derived from general process consistency rule sets or activity specific consistency rule sets.

4.5.6.2 Pre-condition and Post-condition Rule Sets

Rule sets applied to activities as pre-conditions or post-conditions indicate that they are transitional in nature. Pre-conditional rule sets are defined as transitional in the sense that the activity cannot be *initiated* without the rule set being satisfied first. Post-conditional rule sets are defined as transitional in the sense that the activity cannot be *completed* without the rule set being satisfied first.

As pre and post-conditional type rule sets are used in this approach to represent transitional constraints, this is heavily related to the way the proposed approach deals with non-conformance regarding activity sequencing, which is discussed further in section 4.5.7.

Non-conformance that is detected from rule sets applied to a process activity as pre-conditions or post-conditions are referred to in GARDEN as *deviations*.

4.5.7 Activity Sequencing

An important part of non-conformance detection is the detection of *deviations* referring to non-conformant transition [6] between process activities. This is handled in the proposed approach by the application of rule sets to activities in the form of pre-conditions and post-conditions.

The handling of sequencing in the proposed approach is concerned with defining the order in which activities should be enacted. This is distinct from other published research in activity sequencing such as the comparison of anomalies in overall sequence patterns [11] in different process enactments. Instead, the objective is to specify conditions that govern the transition of activities and check that when the activities are enacted, they conform to these conditions.

Therefore, the complicated task of activity ordering in the vein of "once this activity is complete, move to activity *x*" is avoided. This type of sequencing shown in research such as the Little-JIL [12] system where sequencing badges are used can become complicated, considering the different ways activities can legally be enacted apart from sequentially and the various conditions that permit these enactments.

Instead, a less complicated and more modular system of tying activity sequencing to pre-conditions and post-conditions was formulated. This technique was inspired by the pre-requisite approach presented by Park et al. [13] in their attempt to simplify sequencing in the SCORM eLearning Activity Control Model. The technique involves the definition of "pre-requisites", which specifies the conditions that must be met before an activity can legally begin.

An example of such a condition relating to sequencing could be the completeness of one activity being conditional for the next to begin. Zhou et al. describe this concept as "control dependency" [14]. Activity **B** is control dependent upon activity **A** if the completeness of **A** is a condition for the enabling of **B**.

These conditions can be stipulated using the proposed approach by defining them as rule sets. Specifying a rule set as a pre-condition means that the rule set must be complied with before the activity may begin, and as such, it is a transitional rule set. To achieve control dependency, we simply define a rule set for activity **B** that specifies that activity **A** must be complete. The rule set, though defined for activity B, will still check the corresponding enactment data pertaining to activity A to ensure that it is complete and that the rule set has been satisfied to test for non-conformance.

4.6 Conclusion

This chapter has presented in detail the solution and its application to a process in order to detect non-conformance. The solution presented in this chapter is intended to also address the issues covered in chapter 3 in order to detect non-conformance in a way that is generic, flexible and practical, as well as maximising the potential to use the information attained for process improvement. The implementations presented in the following chapters will show how the proposed approach achieves the claims made about it in both this chapter and chapter 3.

References

[1] D. Bogia and S. Kaplan, Flexibility and Control for Dynamic Workflows in the wOrlds Environment, Proceedings of the Conference on Organisational Computing Systems, ACM Press, Milpitas, CA, November 1995.

[2] I. Sommerville, Software Engineering, 8th edition, Addison Wesley, 2007, pp. 74.

[3] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[4] A. Zisman, W. Emmerich, A. Finkelstein, Using XML to Build Consistency Rules for Distributed Specifications, Tenth International Workshop on Software Specification and Design (IWSSD'00), IEEE, November, 2000.

[5] E. Ellmer, W. Emmerich, A. Finkelstein, D. Smolko, A. Zisman, Consistency Management of Distributed Documents using XML and Related Technologies, UCL-CS Research Note 99/94, 1999.

[6] G. Cugola, E. Di Nitto, A. Fuggetta, C. Ghezzi, A framework for formalizing inconsistencies and deviations in human-centered systems, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 5 Issue 3, ACM Press, July 1996.

[7] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[8] S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

[9] I. Sommerville, Software Engineering, 8th edition, Addison Wesley, 2007, Chapter 2, pp. 23.

[10] P. Checkland, Systems Thinking, Systems Practice, Chichester: John Wiley & Sons, Chapter 2, 1981.

[11] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, G. Coleman, Detection and explanation of anomalous activities: representing activities as bags of event n-grams, Conference on Computer Vision and Pattern Recognition, IEEE, Volume 1, 20-25 June, 2005, pp. 1031 – 1038.

[12] J.M. Cobleigh, L.A. Clarke, L.J. Osterweil, Verifying Properties of Process Definitions, international symposium on Software testing and analysis (ISSTA), ACM, August, 2000.

[13] N. Park, H. Kim, K. Kim, H. Park, J. Chun, Y. Hwang, An eLearning Activity Control Model for SCORM's Sequencing Prerequisites, Fourth International Conference on Semantics, Knowledge and Grid, IEEE, 3-5 December 2008, pp. 322 – 329.

[14] Z. Zhou, S. Bhiri, M. Hauswirth, Control and data dependencies in business processes based on semantic business activities, 10th International Conference on Information Integration and Web-based Applications & Services, ACM, November, 2008.

[15] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[16] G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[17] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 7 Issue 3, July 1998.

Chapter 5 – GARDEN Methodology

5.1 Introduction to the methodology

This chapter will provide a detailed methodological guide to applying the GARDEN non-conformance detection approach to a process. Whereas chapter 4 gave a detailed technical description of the solution and how each facet of the methodology works, this chapter concentrates on taking this solution and applying it practically to a target process from beginning to end, as well as detailing the benefits of doing it this way.

The structure of this chapter follows the same sequence as the required tasks to be completed when applying GARDEN to a target process. Before elaborating on the procedure of integrating the GARDEN approach, an overview of the integration procedure is provided in Figure 13.

Once the overview has been provided, the integration procedure begins with some basic assumptions about the target process. In order for a process to be suitable for the GARDEN approach, it must possess certain attributes for GARDEN to be effective. These attributes are listed and the reason for their importance is explained. So, we begin by assuming that the target process for GARDEN possesses these attributes.

As we have a suitable process for improvement via GARDEN integration, we commence by looking at the available process enactment data. In order to detect non-conformance in a process, we require process enactment data. This stage is concerned with analysing what enactment data is produced by the process, how accessible that data is to GARDEN and at what stage it becomes available for evaluation.

Before explaining the rule set implementation, section 5.3 is provided on how GARDEN interacts and evaluates the process enactment data. Since different processes will produce enactment data in different forms and mediums, how GARDEN interacts with this variability is explained, and how we avoid any potential incompatibility problems such as the ones noted in Huo et al. [3] and Cook and Wolf [4] is also detailed. Once the enactment data has been analysed and we know how GARDEN is going to work with it, the rule sets themselves are addressed.

The section on rule set implementation covers the full range of considerations when designing an appropriate suite of rule sets appropriate for detecting non-conformance instances in a process. This covers designing rule sets to effectively constrain the process as per the specification, the design of rule sets to cover specific non-conformance types, how to traverse through the process activity sequence effectively and apply appropriate rule sets, and finally how to change rule sets as the process evolves.

Once the rule sets have been set up, the GARDEN integration is complete and ready to detect instances of non-conformance as enactments of the process are executed. The means of detecting and storing non-conformance along with the data that is derived and logged is explained.

Lastly, before the chapter is concluded, the evaluation of non-conformance is covered. At this stage, GARDEN has been implemented and a process enactment has been run. Some non-conformance instances have presumably been detected, recorded and logged. This section details how to evaluate the non-conformance logs for the purposes of improving the process and addressing those non-conformance instances where appropriate.

The purpose of this chapter is to show how the GARDEN approach can actually be used practically, and how the concepts covered in chapter 4 can be applied to a real process. This will be achieved by providing practical examples with explanations of how the methodology works in this chapter. In order to illustrate this process before discussing it in detail in the sections of this chapter, a diagram is provided in Figure 13 to depict the high level GARDEN integration process. This figure illustrates how the GARDEN solution is implemented and works with a process to detect instances of non-conformance. The dotted lines depict the task of providing an interface for the process domain expert to manage the solution after GARDEN has been implemented.

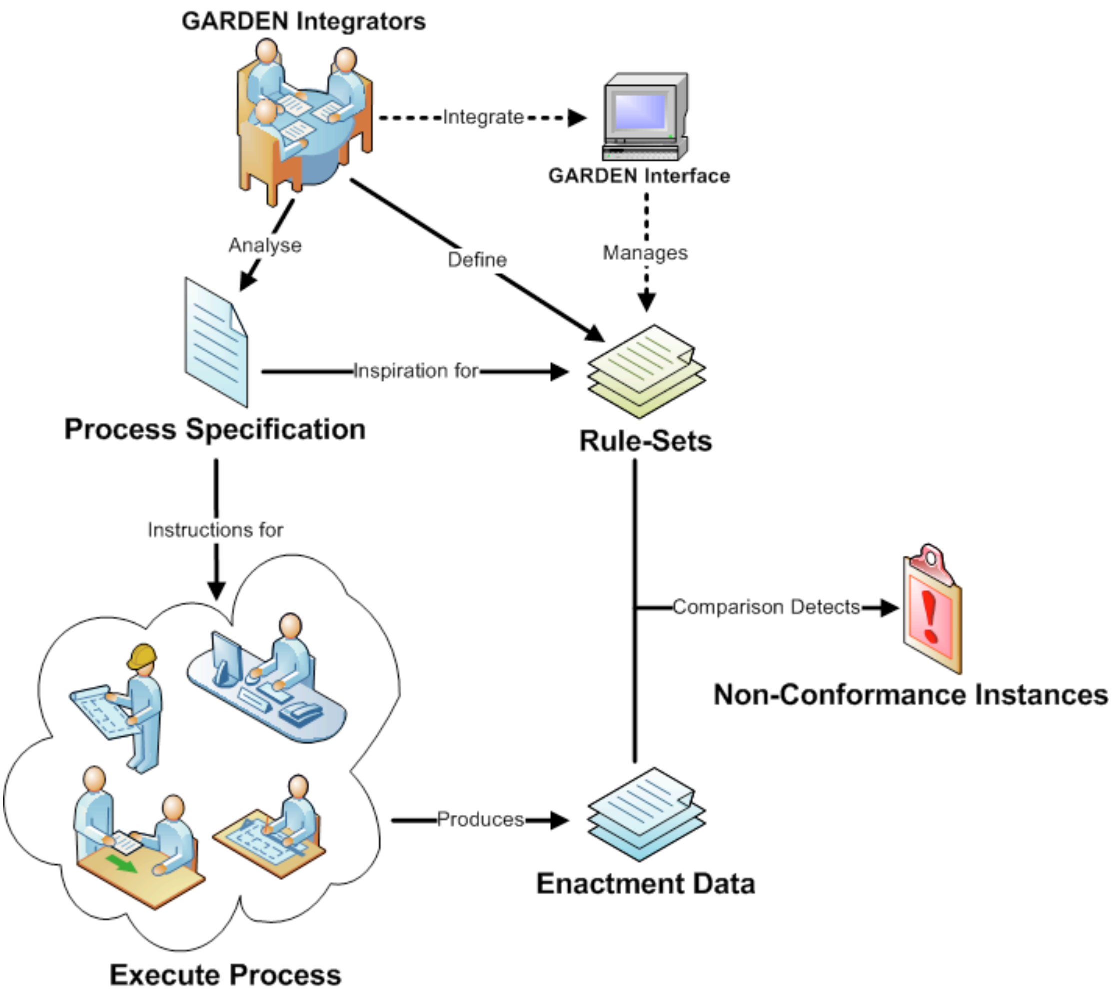


Figure 13: High Level GARDEN Integration Process

5.2 Initial Process Assumptions

Before we can begin to describe the application of the GARDEN approach to a target process, we have to make some assumptions about the nature of a process that would be suitable for this approach.

First, we assume that a clear process specification exists. This is essential because we need a clear account of exactly what the process is supposed to do in order to effectively design rule sets ensuring that the process enactments conform. A clear process specification to work from is a requirement for all non-conformance detection approaches. Process model discovery approaches like Huo et al. [3] and Cook and Wolf [4] use enactment data from previously enacted processes to create the reference specification. The main issue with this is you then compare future process enactments with what was actually done in the past, rather than what was *supposed* to have been done. Cîmpan and Oquendo [5] use the "flawless enactment" simulation to compare future enactments, but even this requires a clear understanding of what is supposed to occur in order to successfully produce the "flawless enactment" reference model.

Second, we must assume that the process produces some enactment data that is recorded or logged in some way. Since GARDEN relies on the process enactment data to make its comparisons to detect non-conformance, we must assume that enactment data exists, that it is recorded in some kind of logical and structured way, and that the data is 'queryable' by the rules GARDEN uses to evaluate it.

Third, we must assume that the process is broken up into activities, or some kind of modular form where each part is distinguishable. As long as enactment data is available, GARDEN rules can be defined to access and evaluate them, but in order for GARDEN to be a practical and useful tool, it will be much more effective if activities are identifiable within the process that rules can be applied to individually.

With these three basic assumptions about the target process satisfied, we may begin applying the GARDEN process non-conformance detection approach.

5.3 Process Enactment Data Analysis

Once a suitable process has been chosen for GARDEN integration given the assumptions listed in section 5.2, the first step is to analyse the available process enactment data. This will give the integrator an idea of what conditions can be checked for and what cannot when designing the rule sets to evaluate the process enactment data. For example: we may want to check if there are timestamps recorded. If timestamps exist in the enactment data, we can create rule sets that check duration times. If there is no discernible way to deduce from the data when things occurred, implementing such a rule set may not be possible.

Each process may have a different mechanism and recording process for storing its enactment data. This is not a problem, as long as we can create rules capable of evaluating it.

The challenge of reconciling the non-conformance detection implementation with the process enactment data is addressed in different ways for different comparable approaches. Process discovery approaches such as Huo et al. [3], and Cook and Wolf [4] build their reference model from the available data of previous enactments. This infers that the enactment data from future process enactments should be directly comparable with the discovered reference model, since they are developed on the same level. In approaches such as Cîmpan and Oquendo [5], however, where the reference model is simulated, care needs to be taken to ensure that enactment to reference model comparisons can be made readily.

Zazworka et al. [9] have the advantage of assuming the data from the software process they plan on working with will adhere to most standard practices, so their CodeVizard tool can mine it and produce data that they can work with. If the process does not conform to these standards i.e. not using SVN or a standard code repository, then it is unclear how their approach will handle enactment data that needs to be checked.

In GARDEN, we can simply tailor the functional implementation of the designed rules to fit the available enactment data to resolve this problem. If the enactment data is stored in log files, then the logic to evaluate the rules and rule sets can be implemented to extract data from those log files. If it is more convenient, we also have the option of massaging the enactment data into a form that is easier to access and query, such as a relational database (if the enactment data is not already in such a form). However, integrators should be mindful that accessing process enactment data as it is recorded is important to minimize time-to-detection, an issue covered in section 3.1.8. The ability of GARDEN to be applied at the level of data enactment is useful in realizing this objective.

Having flexibility in how the rule sets evaluate process enactment data is also advantageous to resolve the issue of process enactment data availability, as covered in section 3.1.9. This is an advantage because even if the available data is minimal, rule sets can still be specified to check it, so at least some form of the approach can still be applied. Other approaches such as the ones relying on data mining [4] [3] may find this impossible without an adequate supply of enactment data. Of course in any approach at all, less available enactment data will result in lower approach effectiveness.

An example of a typical process enactment data storage schema is provided in Figure 14, which has been borrowed from the process enactment data storage database discussed in chapter 6 of this thesis.

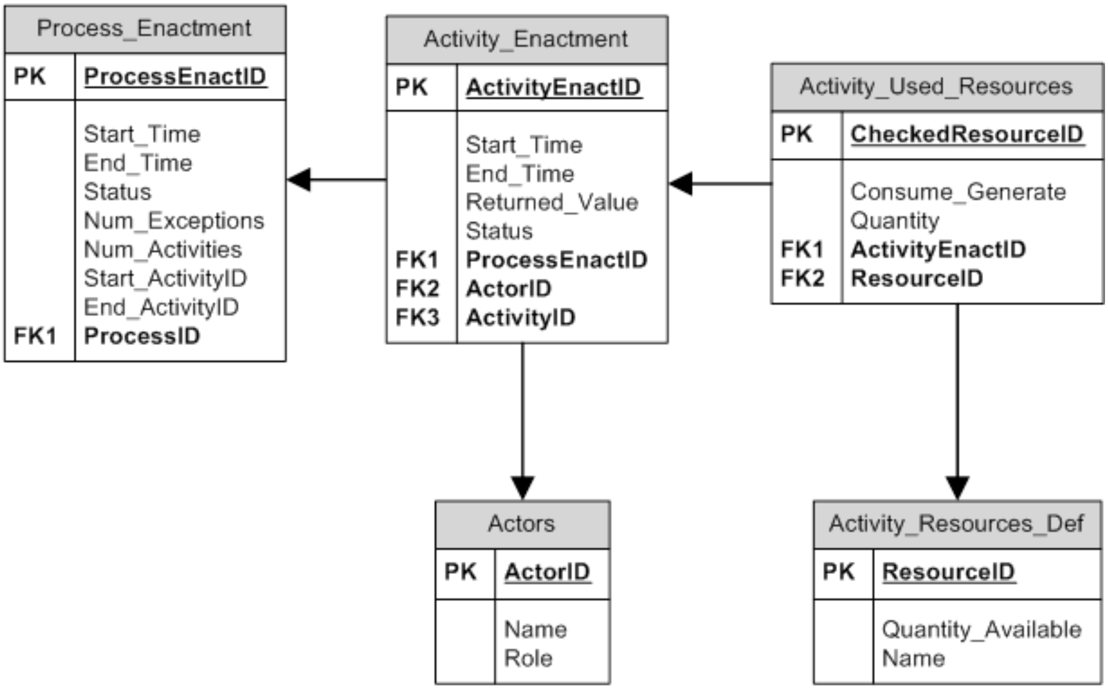


Figure 14: Enactment Database schema

The schema depicted in Figure 14 shows the data structure in which the enactment data is stored from the processes tested in chapter 6. As can be seen in the schema, process timestamps are recorded and included in the logged data, which may be used for evaluation by GARDEN rule sets. As such, the data structures and mediums used to store process enactment data will depend on each different process and the enactment data available.

If any data massaging is required to transform recorded enactment data from one medium or format to another, care should be taken to maintain as much of it as possible without compromising it. There is also a possibility that the form of the recorded process enactment data may change with the evolution of the process, as new and different data becomes available and obsolete data disappears. When this happens, only the affected rule sets in the GARDEN implementation need to be addressed, as well as the associated enactment data these rule sets use for evaluation. This ensures that during process evolution, the GARDEN approach remains flexible to evolve with the process, resulting in reduced maintenance effort and saved time.

Once the available process enactment data has been analysed, we are ready to start defining the rule sets that can be applied to it for the purposes of detecting non-conformance. This is discussed in section 5.5, however first the means the rule sets use to evaluate process enactment data is provided in section 5.4.

5.4 How Rule Sets evaluate Process Enactment Data

This section is concerned with setting up the GARDEN framework so that all the relevant process enactment data is accessible to the rule sets we plan to define, so rules can be easily defined and implemented to check for non-conformance. This framework will provide the basis from which the rule sets can work to detect non-conformance. It is therefore discussed here before a definition of the rule sets themselves is given.

Although a GARDEN implementation will provide the user with an appropriate interface for defining and implementing rule sets, here we show an example of how rule sets may interact with the process enactment data through code in order to evaluate it. This is therefore something that a GARDEN user will not have to worry about, but is included here to show how an instance of the framework might interact with the process data. To illustrate this example, we can assume that the structure of the process enactment data takes the form previously shown in Figure 14, in a relational database.

In order to show how the rules and rule sets access the process enactment data, we first need an example rule set. A good example rule set to demonstrate this is the placing of a minimum and maximum time duration limit on one of the process activities. This rule set is both common to many processes and has a simple implementation, so it fits the purpose of demonstrating process enactment data evaluation well.

We begin by defining the rule set in code that conforms to the rule set specification previously shown in chapter 4 that is replicated in part here in Figure 15 for clarity.

Rule-set	
Name:	*Name of Rule-set*
Description:	*Appropriate description*
Application-Type:	*{Pre-condition, Consistency, Post-condition}*
Process_Section:	*{General Process, Activity(ID)}*
is_Exception:	*{True, False}*
Logic-Type:	*{AND, OR, NOT, XOR}*
Rules	
......	
......	

Figure 15: Rule set Structure

When we begin to define rule sets, we do so with the intention of checking for a certain kind of non-conformance. Once we have decided what that type is, we can begin to implement the rules themselves that are contained by the rule set to achieve this task. Therefore even though the rule set exists on a higher level than the rules it contains, the rule set is defined first. In this case, we begin with a rule set intended to place a time duration restriction on a process activity. The rules within this rule set make sure that the activity both runs long enough for it to be properly completed, but not for an excessive amount of time. Such a rule set may be defined as shown in Figure 16.

Rule-set	
Name:	*Restrict_Time_Duration*
Description:	*Ensure the duration of Activity X is reasonable*
Application-Type:	*Consistency*
Process_Section:	*Activity (X)*
is_Exception:	*False*
Logic-Type:	*AND*
Rules	
time_duration is > 60 seconds	
time_duration is < 300 seconds	

Figure 16: Example Rule Set

The example in Figure 17 shows how this rule set may be defined as a class in PHP code. Each rule contained by the rule set is implemented as a function of that class.

test_ruleset_class_specific_v2.php

```php
<?php
/*Name: Activity_Time_Duration
Description:        Use to set minimum and maximum time durations for activity $ActivityID
Process Section:    Activity
Activity ID:        $ActivityID
Application Type:   Consistency
isException:        false
Logic Type:         AND*/
class Ruleset_TimeDuration
{
    private $min_allowed; //seconds
    private $max_allowed; //seconds

    //evaluates the rule-set and returns Boolean value
    public function evaluate_ruleset ($actual_time) {
        $ruleset_result = true;

        if($this->minimum_time_breach($actual_time) && isset($this->min_allowed))
            $ruleset_result = false;

        if($this->maximum_time_breach($actual_time) && isset($this->max_allowed))
            $ruleset_result = false;

        return $ruleset_result;
    }

    //this function represents a rule within the Activity_Time_Duration Rule-Set
    private function minimum_time_breach($actual_time) {
        if($this->min_allowed > $actual_time)
            return true;
        else
            return false;
    }

    //this function represents a rule within the Activity_Time_Duration Rule-Set
    private function maximum_time_breach($actual_time) {
        if($this->max_allowed < $actual_time)
            return true;
        else
            return false;
    }

    public function set_min_time($min_time) {
        $this->$min_allowed = $min_time;
    }

    public function set_max_time($max_time) {
        $this->$max_allowed = $max_time;
    }
}
?>
```

Line: 21 Column: 38 HTML Tab Size: 4 evaluate_ruleset

Figure 17: Rule Set Implementation Example

From this example code, it can be seen how a GARDEN interface may call an instantiation of this class if the user wanted to apply such a rule set to an activity. The values of the minimum and maximum duration of an activity can be set if needed by calling set_min_time() or set_max_time(), and the rule set can be evaluated by calling evaluate_ruleset().

5.5 Rule Set implementation Analysis

Once we know what enactment data is available, we can start thinking about what rule sets will be needed to properly cover the process specification. Here is where having a clear process specification is important. In order to define useful and effective rule sets that detect important non-conformance instances, it is helpful to know exactly what the process is supposed to do, what its goals are and any issues that are likely to arise during enactment.

This section discusses the method of analysing the process specification in a number of different ways in order to produce the most robust collection of rule sets possible to detect non-conformance. It also deals with rule set change, and how to effectively deal with process evolution.

5.5.1 Designing the Rule Sets

There are two major steps to formulating the rule sets that will be applied to the process and ultimately detect the non-conformance instances that occur. Although the GARDEN framework makes rule set integration easy, knowing exactly what rule sets are appropriate is still required to make the approach work as effective as possible. Therefore, the better the stakeholder knowledge of the process, the better equipped they will be to design effective rule sets. Zazworka et al. [9] achieve this step via interviews with process stakeholders, which could also be applicable when applying GARDEN if the resources are available.

The first step is to help constrain the process according to its specification. This step is concerned with complementing the process specification in constraining how the process *should* be performed. The second step is to identify any special situations where we might want to specifically stipulate occurrences that should not occur. This task is concerned with identifying ways in which the process *should not* be performed. The benefit of this is we are testing the process in two effective ways: a) using the process specification to create rule sets that complement its intended enactment path; and b) thinking about what could go wrong, both internally and externally and whether there are any other extenuating circumstances to check for. This double analysis gives a distinct advantage over other similar approaches.

Both these techniques in formulating a comprehensive array of rule sets for the process are covered in this section. At the conclusion of rule set design we will be left with a list of non-conformance types for which are testing, similar to the non-conformance lists alluded to in Kabbaj et al. [1].

5.5.2 Constraining the Process

The best way to begin specifying rule sets is to start by looking at the existing process specification and to think of ways in which rules could be defined to help constrain it in an even more positive way including placing rule sets as time duration constraints. This is best achieved by interpreting the process specification which will indicate and imply the way in which the process should be enacted. If the existing process specification is of satisfactory quality, then some of these constraints should be quite obvious. Since GARDEN is observatory and there may be no support system available to control the process as included in Cugola et al. [6], care may need to be taken to check for certain conditions that may otherwise have been prevented by a process control mechanism.

To illustrate, suppose a process specification included an activity whose purpose was to validate a CAPTCHA [7] code on a web page. This example is taken from the implementation discussed further in chapter 7. The specification may dictate something similar to the following:

> *A 6 character code will appear in red text, enter this case-sensitive code into the input field provided and click the submit button.*

Even from this simple activity specification, there are a few constraints that come to mind that could be turned into rule sets and implemented as part of the GARDEN approach. These might be:

- minimum time of 2 seconds. A person should take at least this time to read, enter and verify a CAPTCHA code.
- maximum time of 20 seconds. An arbitrary number, but anything exceeding this could indicate there is a problem.
- maximum validation attempts of 3. If a person fails to validate the correct code 3 or more times then this may also indicate a problem.

In essence, the quality of the rule sets applied to constrain the process will be dependent upon the expertise of the person responsible for their implementation. So it is important that the person in charge of rule set specification is someone who has enough experience with the target process and is also knowledgeable about the GARDEN non-conformance detection methodology.

5.5.3 Specifying Specific Non-Conformance Types

Once rule sets have been specified to help constrain the process, we should start thinking of possible ways the process may divert that we may want to specifically cater for. This, according to Sommerville [2] is a difficult task if we aim to be comprehensive, but there may be some standout instances where we want to test the process enactments to ensure they do, or do not behave in some specific way.

Since the rule sets constraining the process should already have been written at this point, the burden of effort is significantly lessened in this step. We only need to write rule sets for specific cases not covered by the "constraint" rule sets. This helps reduce the level of effort required to implement the solution, addressing the issue discussed in 3.1.3, as well as saving us from the burden of having to anticipate every single non-conformance type such as in Kabbaj et al. [1] or Zazworka et al. [9]. In any case, we have the facility to define rule sets catering for these specific instances.

To use the same process specification as before as an example:

> *A 6 character code will appear in red text, enter this case-sensitive code into the input field provided and click the submit button.*

There are also certain negatives that can be immediately deduced from the intention of the specification. These could be:

- no symbols should be submitted including HTML tags
- the code entered should not exceed 6 characters in length

A lot of these types of non-conformance can be covered by constraining the process the same way as explained in 5.4.3. For example, the no symbols or HTML tag rule set could quite easily be covered by a process constraining "only alpha-numeric characters allowed" rule set.

However for whatever reason at the time we may sometimes want to specify rule sets that specifically look for certain types of non-conformance. One such reason is that we may be particularly concerned about users attempting HTML injections, hence the rule set deliberately seeking symbols and HTML tags.

5.5.4 Traversing the Process

Whilst the task of designing the array of rule sets that will ultimately make up the GARDEN non-conformance detection approach is under way, doing so procedurally can help to produce a more comprehensive coverage of the process. Making note of the way the process should and should not be performed is a sound technique when designing rule sets, but this should be performed as the process is traversed from activity to activity.

Since we have already confirmed that the process is suitable for GARDEN integration, we can safely assume that the process activities will be known and identifiable along with some form of sequencing. Therefore, beginning at the initial activity in the process sequence (or one of the initial activities, if there is more than one) we can start with the conditionals for that particular activity.

A good way to produce a satisfactory coverage of rule sets is to begin at the initial activity in the process and ask some questions that should prompt some of the more obvious rule sets with which to commence. These questions might include:

- what is the purpose of the activity and how can rule sets be incorporated to better define and constrain this purpose?
- are there any actions or conditions that should not obviously occur when performing this activity?
- what conditions should hold before the activity begins?
- what conditions should hold after the activity completes?
- what other activities should be completed before this activity should begin?
- are there any possible behaviours that while not ideal, could be earmarked for rule setting as exceptions?

Running these questions over each activity in the process from beginning to end will help greatly in the building of a solid range of rule sets to detect non-conformance when it comes time to enact the process.

Once the activities have been individually constrained with an appropriate set of rule sets, we can turn our attention to the emergent properties of the process itself. Rule set specification should start to become easier at this point, because properties such as the maximum and minimum expected run times of the enactments, or the number of activities that should and shouldn't be performed in an enactment, for example, can usually be derived from the previously scrutinised activities and the rule sets that have been applied to them.

The relative effort in the initial stages of implementing the GARDEN approach for the first time will differ compared to other approaches presented in the literature. Approaches such as Huo et al. [3] and Cook and Wolf [4] circumvent the need to brainstorm rule sets to apply but instead have the burden of having to discover a reference model from enactment data from previously enacted processes. Using the approach presented by Cîmpan and Oquendo [5] requires the user to simulate a "perfect" execution properly to create a reference model.

Different approaches such as these and GARDEN will have varying degrees of implementation effort, but also the type of effort required will differ. The process discovery approaches [3] [4] have more of a brute force kind of initial setup, requiring the mining and structuring of a reference model from existing enactment data, whereas GARDEN is more of a creative process that may increase the level of dependence on the person in charge of its implementation. The fuzzy logic approach is arguably in the middle of these two setup types, requiring both existing guidance to build the reference model but also creative allowance in order to effectively model what a "flawless" process execution would be.

5.5.5 Changing Rule Sets

It is important that when a rule set is changed in some way, a copy of the original is archived and the new, evolved rule set is given an updated identifier. The reason for this is that the previously detected non-conformance instances recorded will have a reference to the rule set that detected it, so if the rule set is modified without any traceability then its relationship with the old non-conformance instances it detected may become corrupt.

There are several ways to prevent this corruption. The first is through the "time_detected" attribute that is shown in Figure 18. If each rule set has a similar timestamp associated with it that indicates when it began being used then timestamp matching could be used from rule set to non-conformance to show which version of the rule set detected the specific non-conformance instance. The second way is to simply archive outdated rule sets and allocate the evolved rule set a new unique identifier that is retained by the stored non-conformance instances it detects.

As with all traceability documentation in development, the purpose of the change should be documented with the new version of the rule set, whether this was to better constrain the process, or to look for new types of non-conformance or simply because the process had evolved and certain rule sets needed to be modified in order to be of use with the changed process.

If rule sets are modified intelligently with standard traceability considerations in mind, then as previously stated, this significantly reduces the required maintenance effort of the GARDEN approach from approaches such as Huo et al. [3], Cook and Wolf [4] and Cîmpan and Oquendo [5] because it is possible to modify rule sets i.e. evolve the approach with the process without having to rebuild the entire reference model for the enactment data.

5.6 Detecting and Storing Non-Conformance

This section is concerned with the detection and storage of non-conformance for reporting purposes. Now that we know what process enactment data is available, the rule sets have been defined and are capable of evaluating the enactment data, we can begin thinking about the non-conformance detections themselves, and what happens when non-conformance is detected.

As the process is enacted, process enactment data will start to become available. A practical benefit of GARDEN is that rule sets may begin to evaluate process enactment data as soon as the data becomes available to GARDEN. This allows for on-the-fly non-conformance detection, or at least fastest possible detection depending on when the process enactment data becomes available (which is process dependent and beyond the control of GARDEN).

Typically, rule sets may be used to evaluate the process enactment data as soon as all the data required by that rule set becomes available. This is dependent upon the rule sets implemented however, and what dependencies the rules have on each other. For example, if a rule set that checks the completion status of another rule set exists, then the latter rule set will need to be evaluated first regardless of the availability of the process enactment data.

When a rule set evaluates the process enactment data and determines that non-conformance has occurred, we want to gather as much data relating to that instance of non-conformance as possible. Since every instance of non-conformance is inherently tied to the rule set, we automatically gain information that can be derived from the rule set (such as which activity it was linked to, or its relative severity/importance).

However there is some additional information we can store relating to the detected non-conformance instance that is also helpful and cannot be derived from the non-conformance instances' associated rule set. The additional data we want to include relates to the time of detection (as opposed to what time the non-conformance actually occurred), which was previously discussed in chapter 4, the remedial action that was taken after detection and the relative success rating of the remedial action. Even though remedial action attributes are not directly related to the non-conformance instance occurring itself, it is still important and useful because it allows retrospective analysis on what are and are not effective countermeasures if the non-conformance ever repeats itself in future process enactments. This is one benefit GARDEN provides that other non-conformance detection approaches do not.

It is unclear from a lot of the previous research in the literature exactly how non-conformance is logged, or what information is specifically logged with the instances. This is relatively surprising, since the quality of the report on what non-conformance instances have occurred has a direct relationship with our ability to take action or make decisions that improve the process. Furthermore, the importance of the difference between time of occurrence and time of detection is not emphasised in other approaches. GARDEN defines time to detection as a key issue, so time of occurrence and time of detection are included as attributes in the non-conformance log to reflect this.

The non-conformance object has already been described in detail in chapter 4, however it is re-addressed here to illustrate some of the data that is logged when recording non-conformance in order to provide the most useful report after the process has concluded.

	Non-Conformance_Instance		Example Data
PK	**InstanceID**	INT NOT NULL AUTO_INCREMENT	1
	Time_Detected	DATETIME	2011-02-10 13:54:38
	Time_Occurred	DATETIME	2011-02-10 13:31:20
	Remedial_Action	TEXT	Reduced CAPTCHA chars to 4
	Remedial_Success	ENUM(Poor, Average, Good)	Good
FK	**Rule-SetID**	INT NOT NULL	8

Figure 18: Non-Conformance Instance Storage

5.7 Evaluating the non-conformance data

Once a process has been enacted, the repository will contain logs of each non-conformance instance that was detected along with a relationship linking it back to the rule set that detected it. Over time after repetitive enactments of the process, it will be possible to tell not only which process enactments caused which non-conformance types that were detected, but also patterns in the data that can reveal weaknesses in the process specification. This is what the extra attributes included in GARDEN being the history log and severity rating (discussed in chapter 4 section 4.3.1.7) are designed to aid with.

When the process is enacted for the first time, there will be a number of records in the non-conformance repository that indicate what non-conformance instances were detected for that particular enactment. These particular instances will need to be analysed manually for now (an example of this is also provided in chapter 7). Each non-conformance instance could indicate various things; it might be a problem with the actor performing that particular section of the process or it could indicate a problem with the process specification. Over time, as more enactments are carried out, certain non-conformance types will be detected more often than others and once the repository of enactments for a process becomes more numerous and comprehensive then the task of analysing the non-conformance instances for ways to improve the process becomes much easier. Processes that are enacted more often compared to processes that are enacted less frequently are therefore easier to improve, not just with the GARDEN approach, but with any of the non-conformance detection approaches discussed in this thesis.

As previously mentioned, there is nothing present in the literature on process non-conformance detection that specifically focuses on the remedial actions following the detection of non-conformance. However, simply logging each instance whilst maintaining traceability can show how the process specification can quite easily be improved just by simple pattern recognition in the non-conformance instances being detected. This is shown in greater detail with case study data in chapter 7.

At present, the mechanism to determine the severity of non-conformance instances and then provide recommended remedial actions is rather a manual initiative and is part of the future work related to this thesis. Some of the concepts related to how GARDEN can achieve this have begun to be explored and tested however, and these concepts have been presented in Thompson and Torabi [8].

5.8 Conclusion

The material covered in this chapter gives a detailed overview of how the GARDEN approach is actually integrated for a process, why it is integrated in this way and the benefits of doing so. The methodology of applying the GARDEN approach to a process has been discussed with some examples to show how it would work in practice. This was presented chronologically, beginning with the analysis of process enactment data, through to designing and applying rule sets and detecting and storing non-conformance instances. The accessibility of different process enactment data was also discussed, and the provision of post non-conformance detection considerations as discussed in chapter 4 was revisited.

In the following chapters, two case studies where the GARDEN approach has been used and evaluated will be presented and discussed. These implementations showcase the way GARDEN is intended to be applied to a process as shown in this chapter with a description of the processes involved and the outcome of the case studies.

References

[1] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[2] I. Sommerville, Software Engineering, 8th edition, Addison Wesley, 2007, pp. 74.

[3] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[4] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 7 Issue 3, July 1998.

[5] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[6] G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[7] Official CAPTCHA Site, http://www.captcha.net/

[8] S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

[9] N. Zazworka, V.R. Basili, F. Shull, Tool Supported Detection and Judgment of Nonconformance in Process Execution, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, October 2009.

Chapter 6 – Case Study 1: The Process Simulation Environment

6.1 Overview

This chapter covers the first phase of the GARDEN implementation and evaluation through a simulation environment. Chapter 7 will cover the second phase - a real world implementation with result process data from both before and after GARDEN was integrated.

The simulation environment presented in this chapter can be used to simulate different processes and apply defined rule sets to the process enactments in order to detect non-conformance instances. This provides an adequate platform from which to test GARDEN through the many different processes that can be simulated, and the limitless number of times a process may be simulated. As such, this tool can be used to gauge large amounts of quality data that can be used to evaluate the GARDEN approach. Simulation of the test processes was in this case a tool used as Banks and Carson asserted as a computerized representation of a real world process in order to observe its behaviour more easily as it progresses through time [2].

The plan for this chapter is to first overview this tool and the environment it runs in, and provide a high level architecture of how it works and how processes can be simulated and non-conformance detected. The data structure describing how process data is stored, how process enactment data is stored and how rules and rule sets are stored is presented and the database schemas along with their relevance are explained.

The three components that use and run on the data layer are then presented and explained in detail one by one. These components are used to define specific processes along with rule sets to constrain them (facilitating the steps described in chapter 5 section 5.5), to simulate the process enactments, and to evaluate the rule set data with the defined rule sets to detect the non-conformance instances. A detailed explanation of how each component in the tool achieves this is given.

Finally, a test-case process is explained. The details of the process are given; along with how it can be defined using the tools in this simulation environment, how it will be simulated and how any non-conformance instances would be detected. This involves describing the process itself along with the rule sets applied to it and explaining how these rule sets are appropriate and applicable.

6.1.1 Introduction

In this chapter, the simulation environment used to test the validity of the GARDEN approach and to discover any issues which may be present in the approach is presented. The environment consists of a central database that feeds three separate components that when used in conjunction, can test all aspects of the GARDEN methodology. These components are used to:

- define processes;
- specify rules and rule sets;
- apply rule sets to defined processes;
- simulate the enactment of a defined process;
- detect non-conformance by evaluating process enactment data with previously defined rule sets.

The purpose of this environment is to validate that the GARDEN approach is a viable solution to process non-conformance detection. If any issues exist with the GARDEN approach, the simulation environment described in this chapter is valuable to help discover them, as it is easy to define and simulate a variety of different processes to test it. This flexibility also helps confirm the validity of the approach in general to detect process non-conformance.

In this chapter, the environment for defining the process and simulating it is explained in detail, along with how to define and apply rule sets to detect non-conformance. There are three components that fit together to make up this tool. First, a high level architecture depicting how these components work together is presented before each is explained in detail, one by one. Also, the data structures behind the components are discussed along with how they store information relating to the process definitions, process enactments and non-conformance related data.

Although this environment was used to test GARDEN against many processes, one of these simulated processes is presented to illustrate more clearly how a process can be defined, simulated and checked for non-conformance using the environment discussed in this chapter.

6.1.2 Environment Description

This simulation environment is comprised of three separate but related components that when run in conjunction, can be used to define and simulate a variety of processes and detect any non-conformance instances occurring during the simulation.

Each component is written in C# .NET and uses a MySQL backend database management system to structure and store all the data relating to the process specification, the enactments performed and the non-conformance instances that were detected. Each component is described in detail in this chapter.

A key benefit of this tool is that it provides a structured environment where different processes can be defined and simulated and the GARDEN approach can be tested against them. This environment guarantees a consistent process structure because the process specification data has to be stored within the confines of the MySQL schema that was created for this tool.

Creating a simulation tool also eliminates the preliminary problems of first finding a suitable process on which to test the approach on and second, confirming that an adequate amount of process enactment data can be supplied to ensure a satisfactory level of testing of the proposed approach.

An early version of this preliminary implementation was presented in [1].

6.2 High Level Architecture

A diagram depicting the high level architecture of the environment set up to test the proposed approach is shown in Figure 19.

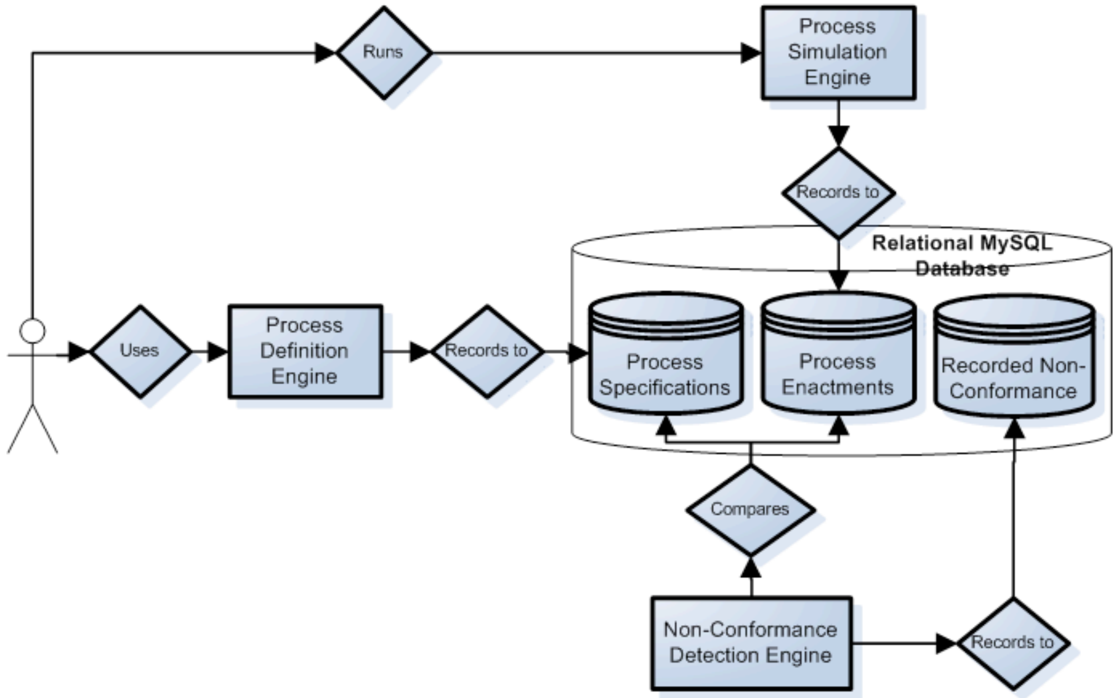


Figure 19: High Level Architecture

There are three subsystems at work in the simulation environment set up to initially test the viability of the proposed approach. Each subsystem is implemented in the operating environment as a separate application and all three applications connect to the same relational database server.

1. Process Definition Engine. This component allows the user to provide a basic specification of an existing process. Once the process and its activities have been specified, the component then facilitates the definition of rules and rule sets and their application to the defined process. This sub-system is explained in further detail in section 6.4.

2. Process Simulation Engine. This component simulates a virtual enactment of one of the existing defined processes. It provides an interface for a user to manually enter process enactment data values as the process runs. This is to provide maximum flexibility and also a mechanism to facilitate comprehensive testing of defined rule sets. The process enactment data is then stored for comparison with the rule sets that were defined using the process definition engine.

3. Non-conformance Detection Engine. The purpose of this component is to evaluate the rule sets by comparing the rules contained within them to the simulated enactment data. When an instance of non-conformance is found, it is logged and stored in the database. A report is generated at the end of the simulation that details all the non-conformance instances that were detected.

6.3 Data Structure

6.3.1 Overview

The back end of the simulation environment is set up as a relational database that was implemented in MySQL. All data pertaining to the process, enactments and non-conformance is stored in the one common database. The purpose of the simulation environment is to confirm that the proposed approach had merit by testing it using different situations. Confining all data sources to a common database guarantees the consistency of the data for comparison purposes as well as keeping the back end simple and generic.

The full database schema is complicated because it has data tables that service three different component engines all with related data. Although the schema is confined to a single common database, the tables within are still segregated into three distinct parts, identifiable by their relation to the components that write to them. They are covered in this section and each part of the schema is shown.

6.3.2 Process specification data structure

Figure 20 depicts a section of the database schema that relates to the storage of process specification and rule set data. This figure along with most other database representations in this thesis are depicted as Microsoft Visio database models, as they facilitate the presentation of table fields as well as primary and foreign keys that add greater clarity to the diagram.

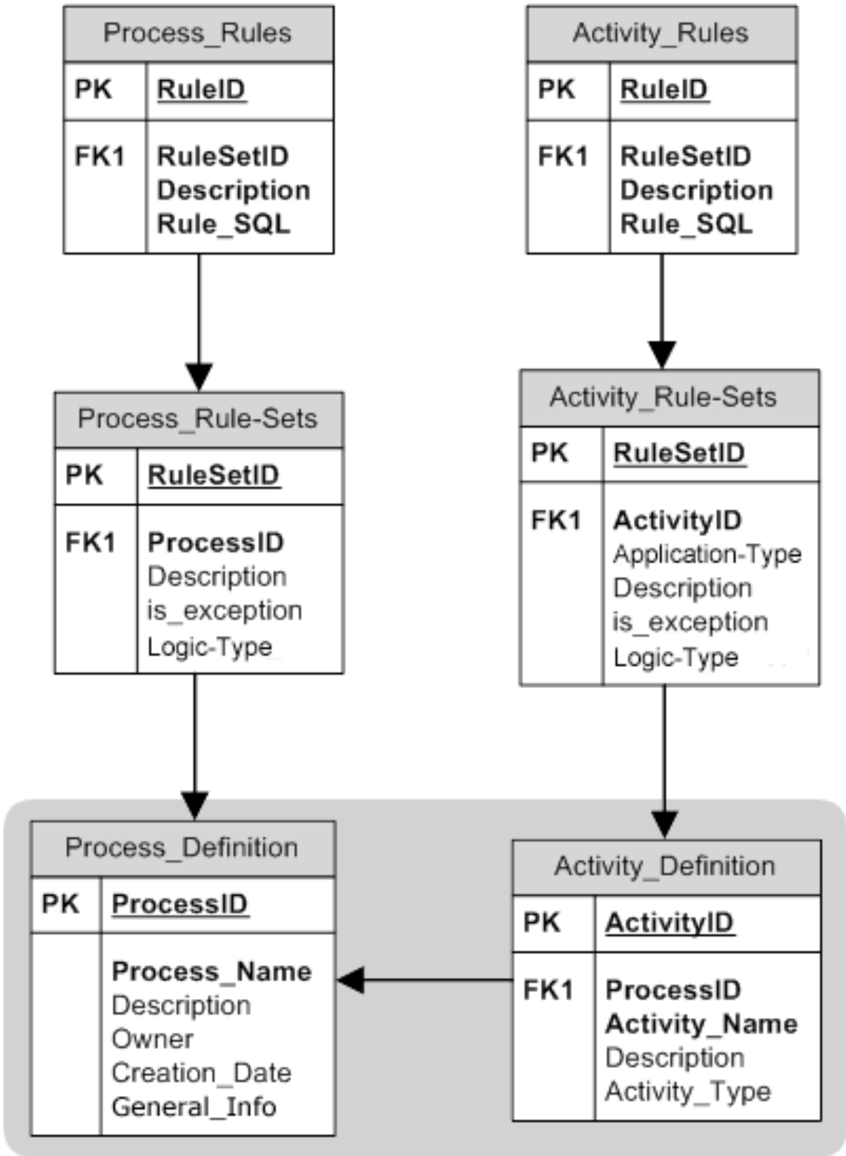


Figure 20: Process Specification Database Schema

As can be seen in the section in Figure 20 with the grey backdrop, the actual process specification is quite basic. It requires only some identification data to be input along with some activities and their associated data. Since GARDEN is aimed at enhancing the existing process specification, most of the stored data aimed at constraints for non-conformance detection are covered by the rules and enactment schema.

The reason this diagram is included in this section is to show how the actual process specification is stored, how the rules and rule sets that make up the GARDEN implementation are stored and how this implementation relates to the defined process. Using the schema represented in Figure 20, we can clearly see the relationship between the two concepts at implementation level.

The rest of the schema depicted in Figure 20 that is not over the grey backdrop shows how the rules and rule sets are stored and how they are related to the process specification. The schema is simple here; rule sets may be defined for either the activity or the process and are linked via a foreign key. The rules within the rule sets contain the actual SQL queries that are used to check the enactment data. So in practice, SQL queries are stored in the same database they are used to query. The Process Simulator engine extracts these queries and runs them, which will be discussed further in section 6.5.

6.3.3 Process Enactments Data Structure

In this section, the storage schema for process enactment data is provided. This schema has been designed to accommodate a wide range of processes and the different enactment data that may be associated with them. Each process defined and simulated using this tool must return enactment data consistent with this schema, however since we can use the tool to define and structure the process specification this is not an issue. The schema provides an adequate medium for various process enactments to store different data, to which rule sets can be applied to test the GARDEN approach.

Different processes may produce enactment data in different ways, and GARDEN will be able to handle this without a problem on a process-by-process basis. However this schema was designed in order to create a feasible enactment data structure compatible with many different processes used to test GARDEN and to keep these processes consistent with the simulation tool. This structure is used to facilitate the easy storage of enactment data from simulations that were executed for different processes in a way that would make rule set application simple. Figure 21 depicts the way the simulated process enactments were stored from the actions of the simulator engine.

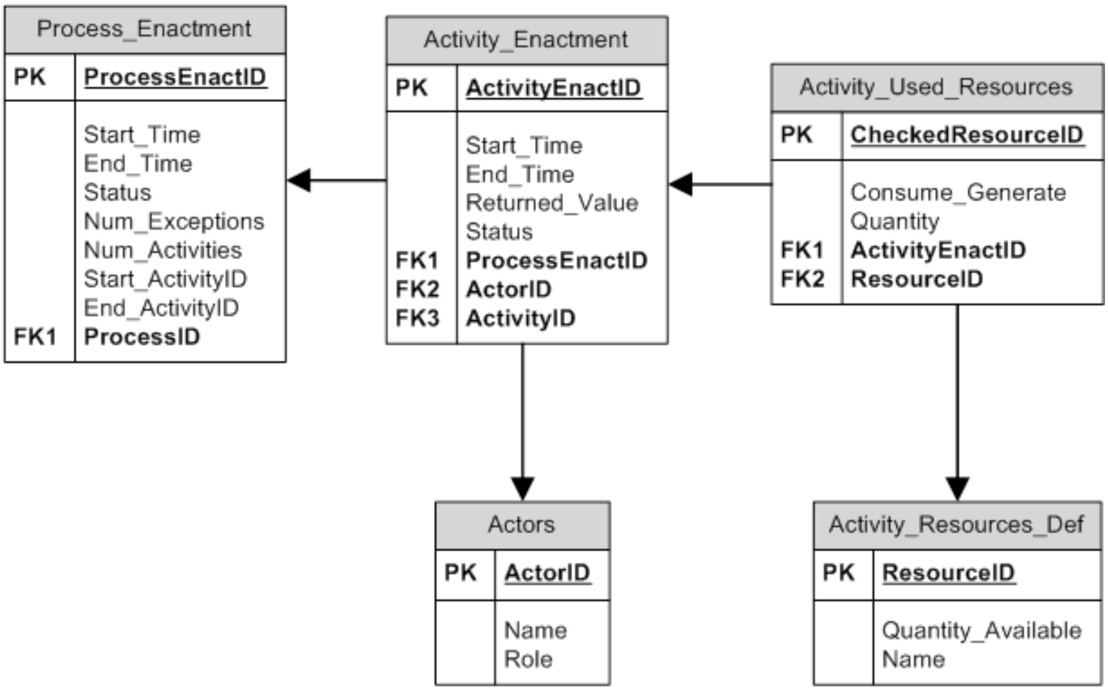


Figure 21: Enactments Database Schema

As can be seen in Figure 21, there are two entities that do not technically store "enactment" data. These are the *Actors* and the *Activity_Resources_Def* entities. The purpose of these tables is to give an enactment reference to both actors responsible for handling activities and also a resources log of available resources an activity might use while being enacted. In order to add depth to the different possible process types and what they are capable of doing, a log of available, required and used resources is provided. We can also log different actor roles and assign them responsibility for different process activities. This is all data that rule sets may be applied to, in order to show the versatility and power of the GARDEN approach.

There are also two foreign keys depicted in Figure 21 that seemingly have no reference apparent in the schema. These are the *ProcessID* key specified in the *Process_Enactment* entity and the *ActivityID* specified in the *Activity_Enactment* entity. These foreign keys actually reference the process specification entities *Process_Definition* and *Activity_Definition* respectively as shown in Figure 20 in order to show which sets of enactment data refer to which process specifications.

The real process enactment data is stored in the top three entities. Information stored by these entities that is available for the rules and rule sets to evaluate are as follows:

- activity start and end date-times that may be checked for things like duration or activity sequencing. In the *Process* entity, the start time is automatically recorded as the earliest start time and the latest end time of all of its enacted activities by the simulator component.
- an activity "return" value. This value can be anything at all. It was decided that if an activity should have some kind of value associated with it that might need to be checked, then the value could be stored here. For the purposes of this simulation environment, if an activity had more than one such value, then the activity could be further modulated and split into separate activities.
- the total number of activities and exceptions that occurred throughout the execution of the process. Rule sets may be defined to put minimum or maximum values on such properties as the number of activities that should be run.
- any resources that may have been consumed or generated during the enactment of a single activity. This includes required resources versus resources actually used or generated and the available quantity of any specific resource.
- the actor who actually enacted a specific activity. There may be Rule Sets defined that stipulate which activities should be enacted by only certain eligible people.

This array of data is not intended to be a comprehensive schema that covers all or most possible process enactment data values. Considering that GARDEN is intended to be kept generic, attempting to do so would be too broad and complicated. Instead, what is presented here is a schema for storing a set of process enactment data against which rule sets can be tested.

6.3.4 Non-Conformance Instances Data Structure

Previously in this thesis in sections 5.7, 4.3.4.4 and 4.3.4.5, post-detection considerations have been discussed along with the effect this has on what we should log regarding non-conformance data. Specifically, scope for logging both the remedial action taken post-detection and the success rating of that remedial action for historical reflection purposes were discussed.

Although this implementation of the simulation tool did not directly address how these post non-conformance detection considerations would be resolved, the structure used to store non-conformance still catered for this future consideration. The schema for storing detected non-conformance instances is depicted in Figure 22.

Non-Conformance_Instance		
PK	**InstanceID**	INT NOT NULL AUTO_INCREMENT
	Time_Detected	DATETIME
	Time_Occurred	DATETIME
	Remedial_Action	TEXT
	Remedial_Success	ENUM(Poor, Average, Good)
FK	**Rule-SetID**	INT NOT NULL

Figure 22: Non-Conformance Detected Instance Schema

Since the database schema for storing rule sets contains separate tables for rule sets relating to the process and rule sets relating to the activities, an additional attribute is included in this schema that indicates to which rule set table the non-conformance instance is related to.

6.4 Process Definition Component

There are two primary purposes of this tool:

1. to define a basic process specification to which rules and rule sets may be applied to;

2. to define rules and rule sets and apply them to a process.

6.4.1 Defining a Process

The component begins with an interface to select an existing process or to create a new one.

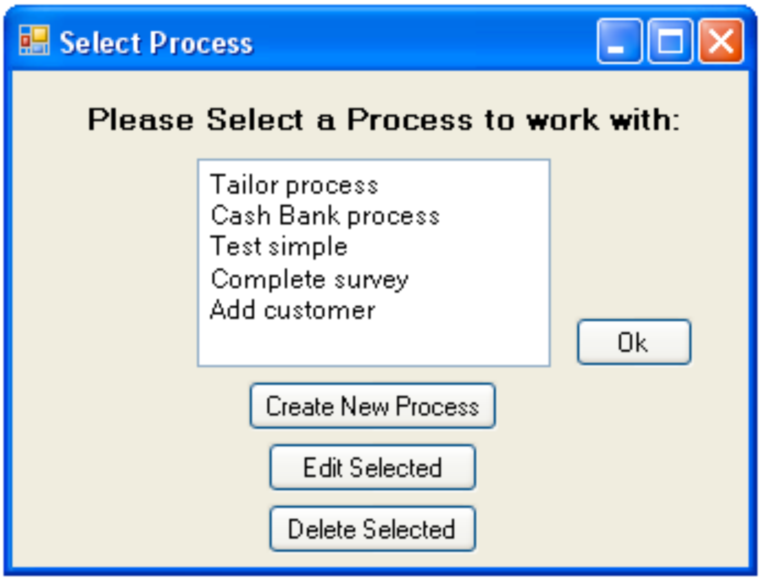


Figure 23: Select Process

When creating a new process definition (or editing the high level details of an existing one), some arbitrary information may be input in order to identify the process along with some brief information that explains what it is or what it does. Figure 24 depicts the interface used to edit the basic process data, but on this interface, there are some noticeable extra fields. These are the minimum and maximum activities and the maximum exceptions fields. The component facilitates some automatic generation of some basic rule sets, such as limiting the maximum number of activities that may be performed during the enactment of the process. If any numeric value other than zero is entered into these fields, the logic included in this component will convert the information entered by the user into SQL rules and stored in the database ready to be compared to enactment data. This helps save the user some time when they want to specify some common rules such as placing minimum and maximum boundaries on the total number of acceptable activity executions for the process.

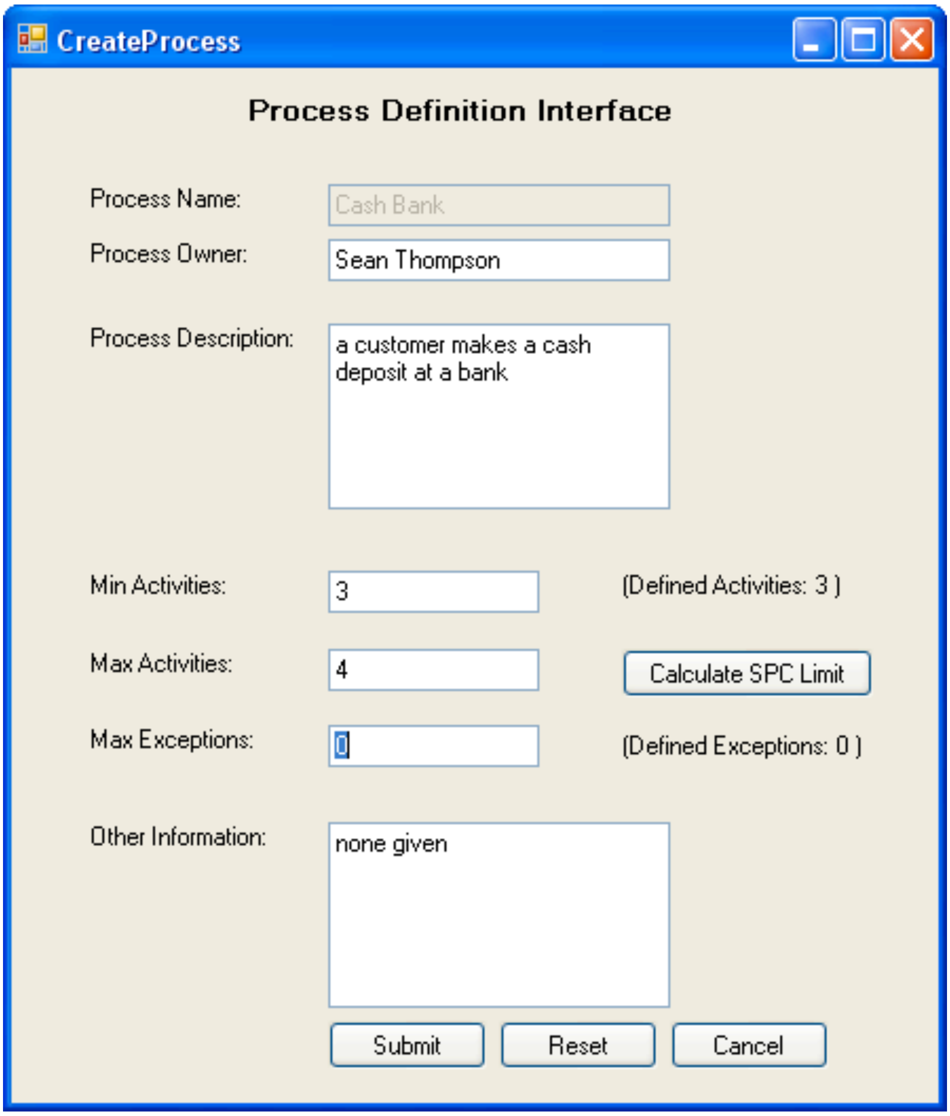


Figure 24: Process Definition Interface

Once a process has been selected or created with which to work, we can begin setting some properties and specifying activities for it. This will make up the specification of the process that will be enacted. The main screen that presents the bulk of the process and activity details is depicted in Figure 25.

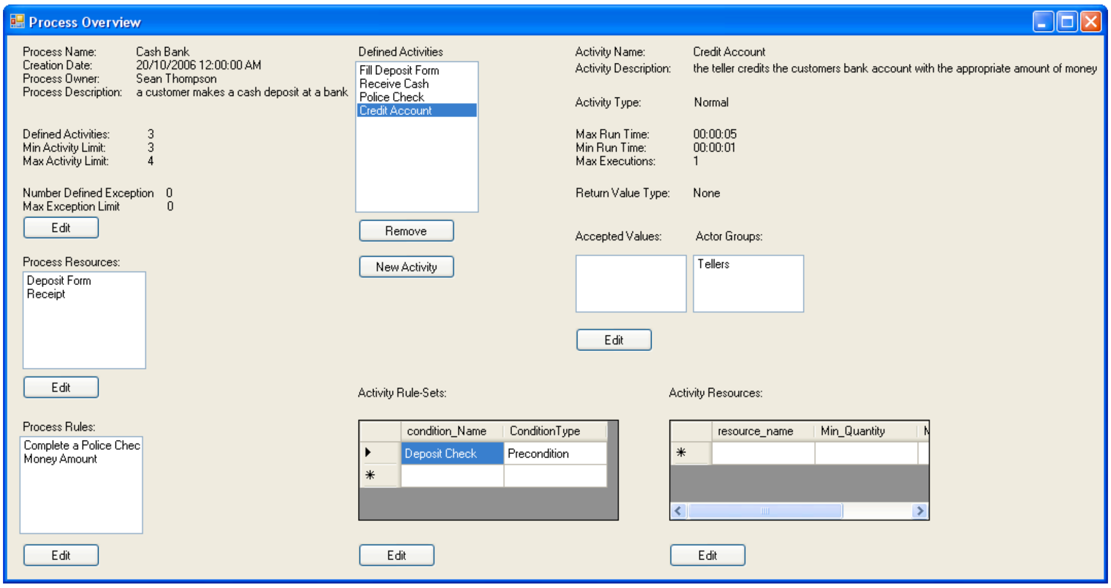


Figure 25: Main Process Definition Screen

6.4.2 Defining Activities

When activities are defined for the process, the most important aspect regarding the specification is that it knows that the activity exists and is associated with the process. We therefore only need to store the name, description and the type of activity it is in the specification, as rule sets can be defined to complement it at a later stage. However for user convenience, the activity definition interface of this tool provides the ability to easily apply some automatically generated rule sets, such as placing minimum and maximum time duration constraints or by restricting the actors and actor groups permitted to perform the activity.

The reason the activity definition interface is constructed this way is to provoke the user into really thinking about the activity they are listing for the process. By providing some simple predefined rule set types along with the identifying activity definition fields, the user thinks not only about simply listing required process activities, but also about what the activity actually does and how it can be constrained at a high level. Since there are only a few predefined rule set types included in this interface, the user is not overburdened with trying to constrain the activity completely from the start. They can simply reflect on what the process activity is for, apply some high level constraints and move on to the next one, returning to any previously defined activities as they wish. An interface for applying rule sets more thoroughly is provided later.

Figure 26 depicts the interface for creating and editing process activities. This interface is accessed either by clicking the “New Activity” button to create a new activity or by selecting an activity from the list and clicking the relevant “Edit” button, as shown in Figure 25. The rest of the fields and options on the interface in Figure 26 are concerned with automatic rule set generation options, which are discussed further in sections 6.4.3 and 6.4.4.

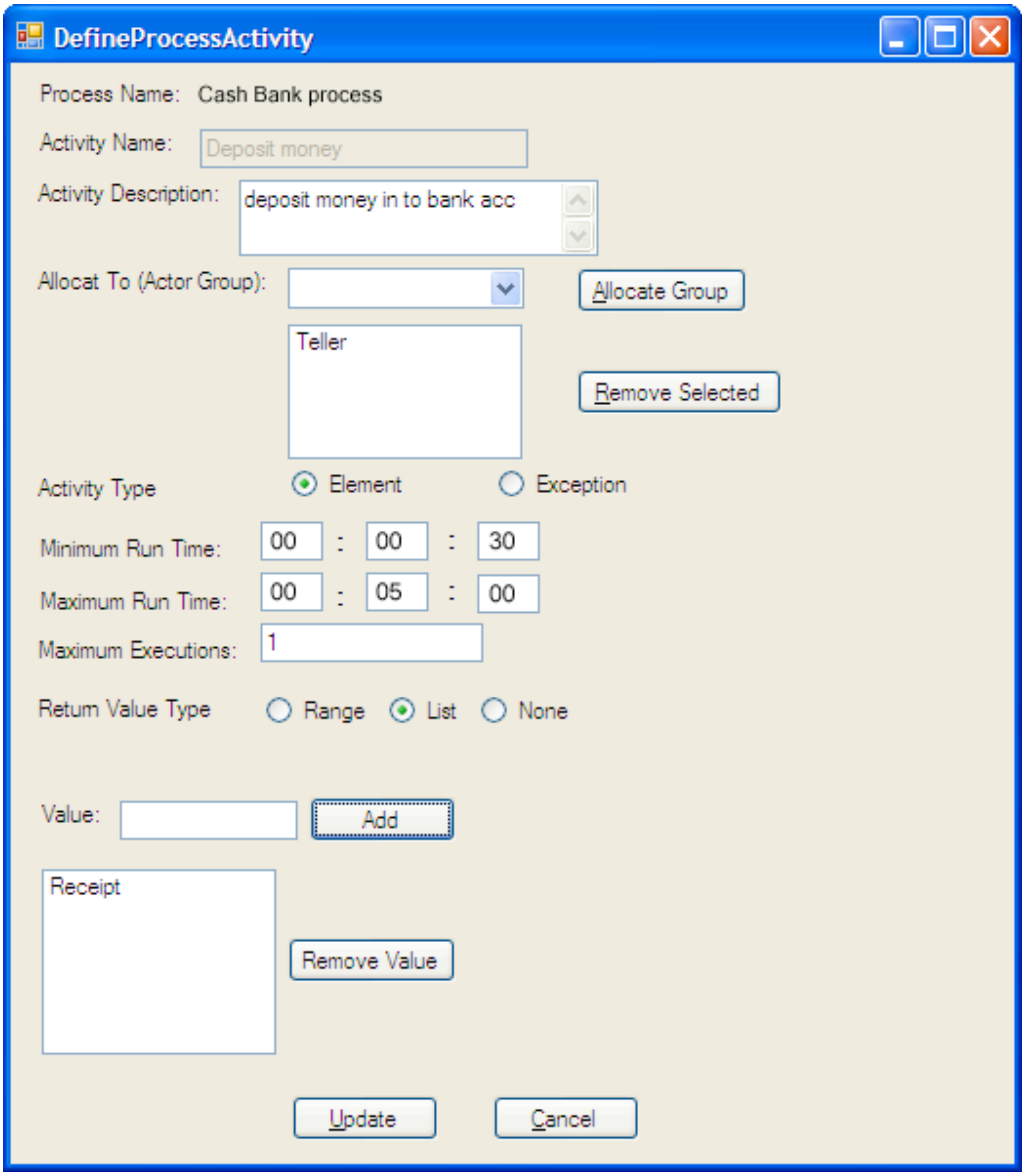


Figure 26: Activity Definition

6.4.3 Specifying Rules

This simulation tool provides an assortment of interfaces to easily define and apply rules to the process enactment data automatically. However, in some cases the user may want to define specific rules that are not supported by the interface to apply automatically. In these cases, rather than attempting to provide a complicated interface capable of predicting all potential rule types, a simple interface is provided which gives the user the power to manually define and save rules at rule level, without relying on the tool to automatically generate them. This interface is shown in the following section in the lower part of Figure 27.

This simulation tool creates rules by converting the user-defined rule into an SQL database query that is stored and later used to evaluate the process enactment data. Each rule is stored in the database and retrieved and executed when the time comes to evaluate process enactment data. Along with the SQL query used to evaluate the process enactment data, the rule is stored with an English description of what it is/does and the activity to which it is linked (or the process in general) as depicted in section 6.3.1, Figure 20.

When the SQL query is run, it will check the process enactment data for any conditions set by the rule and return a result. If the result set is populated with data, it means the rule has found process enactment data that satisfies the rule, which in the simulation tool is equivalent to the rule being true. If the result set is empty, then the rule returns false, as there was no process enactment data to satisfy the conditions of the rule.

To demonstrate this logic using a simple example, say we wanted to create a rule that:

- an activity whose unique identifier value was "4";
- cannot have an execution time of greater than 5 minutes.

An SQL query could be constructed to test this data as follows:

```
SELECT 1 FROM Activity_Enactment
WHERE TIMEDIFF(End_Time, Start_Time) > '00:05:00'
AND ActivityEnactID = 4;
```

If a result is returned by this query, then the rule returns true. This rule will ultimately be contained within a rule set, which will specify that if this rule returns true, log a detected non-conformance instance.

For the most part, rules and rule sets in this component are automatically generated via the interfaces to make things as quick and easy as possible for the user. However the addition of this feature grants users the power to define any rules they may require while maintaining interface simplicity, and also demonstrates the true capabilities of the GARDEN methodology.

6.4.4 Specifying Rule Sets

When rules are defined, they are automatically assigned rule sets. A lot of the rules and rule sets are automatically generated in this component, however it does provide some flexibility to make rule set specification as simple as possible.

Figure 27 depicts an interface provided by the definition engine to add, edit and modify some of the rules that are contained by certain rule sets. As shown, some flexibility with the "Return_Value" field in the enactment table is afforded, with the interface facilitating dynamic rule generation. If this is insufficient, a field is also provided where users can manually define their own rules as SQL queries designed to evaluate the process enactment data. The mechanism for providing the user with this feature is explained in section 6.4.3.

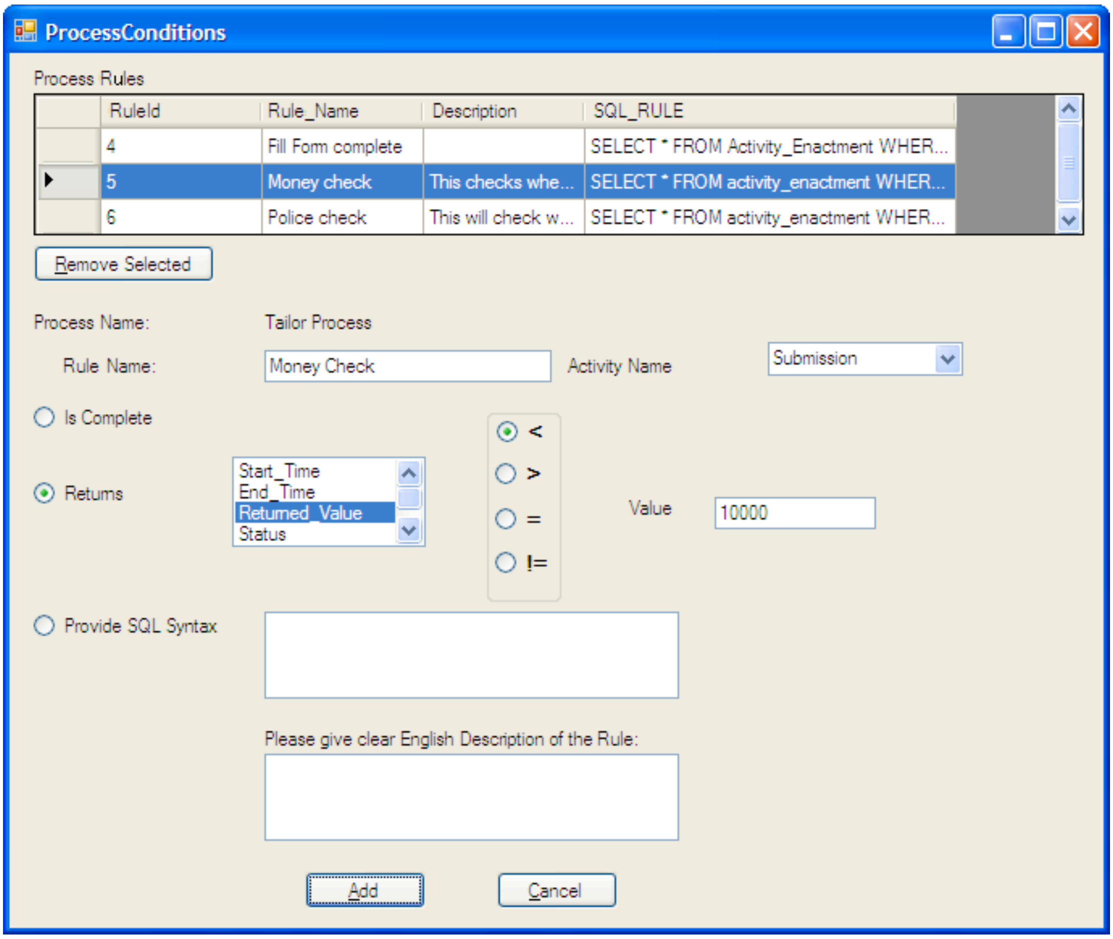


Figure 27: Rule Set Flexibility Interface

The process definition component also facilitates the creation and editing of rule sets from existing rules that have been recorded to the database. The interface for manipulating this is depicted in Figure 28.

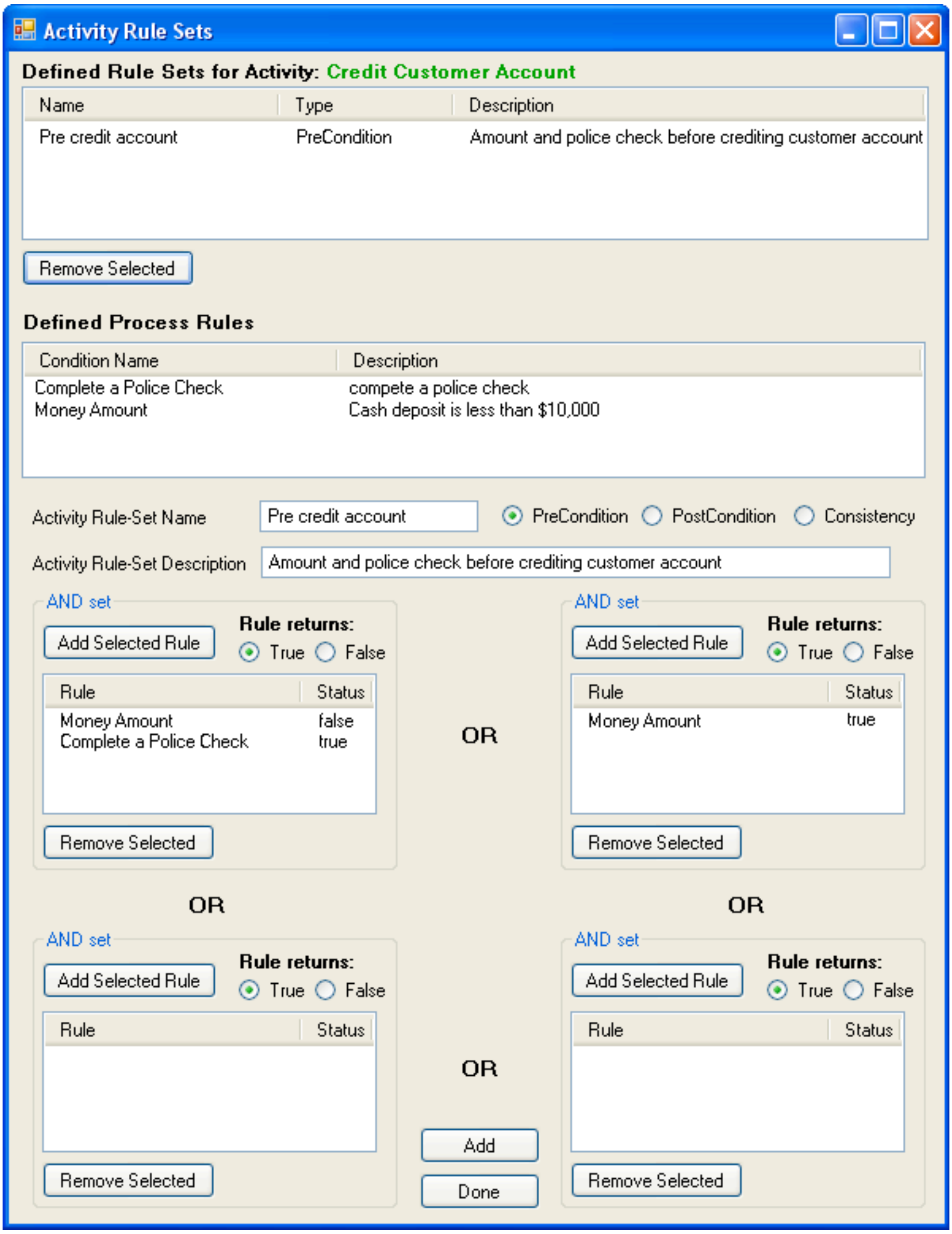


Figure 28: Rule Set Manipulation Interface

The purpose of the interface depicted in Figure 28 is to provide a mechanism for grouping individual rules into rule sets. The interface also provides a mechanism for specifying "AND" and "OR" logic-type rule sets. "NOT" logic-type rule sets can be specified via the "true" and "false" radio buttons depicted near the rule lists.

6.5 Process Simulator

Before the process simulator engine can begin, the non-conformance engine should be running. It runs as a daemon (running background application) and is capable of detecting non-conformance on the fly, and is described further in section 6.6.

The purpose of the process simulation engine is to provide a virtual enactment of a defined process. Although the simulator simulates the process automatically, it also provides a mechanism for the user to manually intervene in order to ensure all their rule sets are tested thoroughly and that GARDEN is working as it should. This includes the ability to add and change different process enactment data values or change the order in which process activities are carried out.

The simulation engine starts up with the screen depicted in Figure 29. Here a process can be selected to simulate from the list of defined processes that are stored in the database modelled previously in Figure 20.

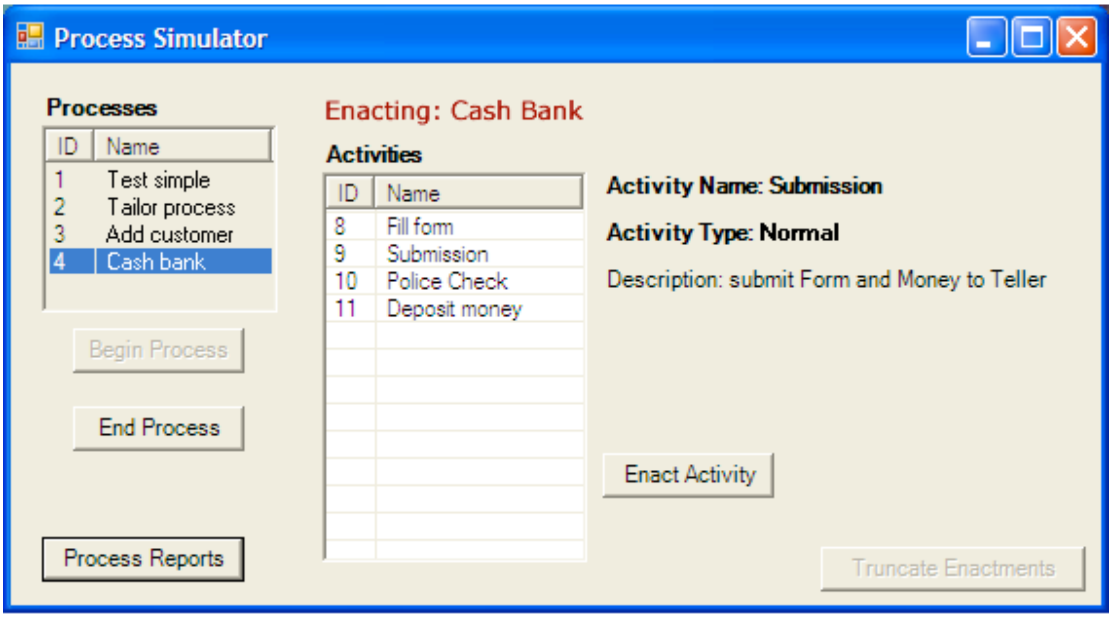


Figure 29: Process Simulator

The simulation begins when a process is selected and the "Begin Process" button is clicked. Activities may then be selected that are part of the process and "enacted" through the simulator. Figure 30 shows an interface allowing a user to manipulate how an activity might be enacted.

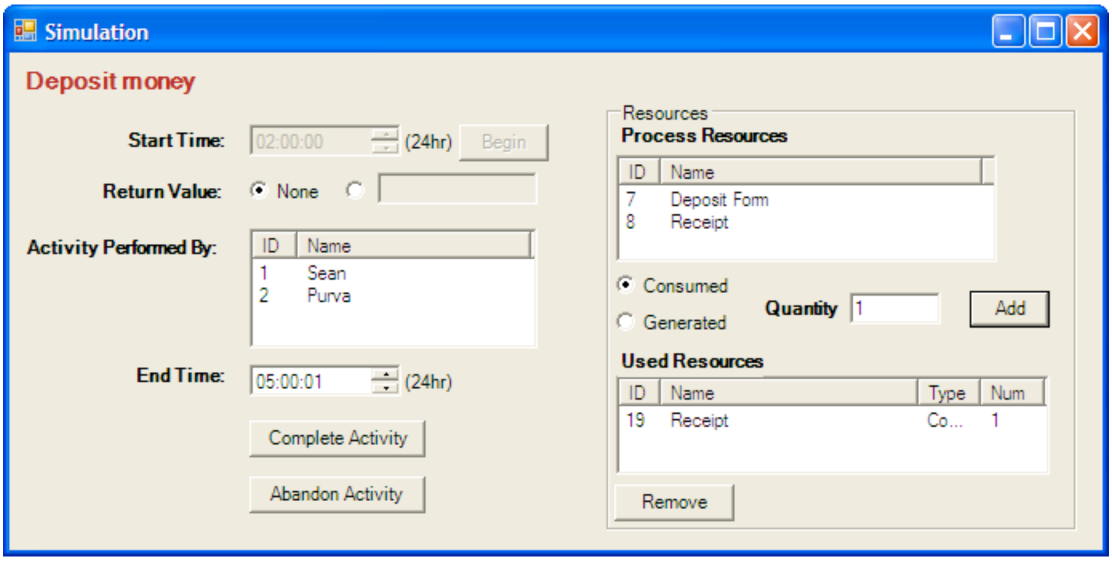


Figure 30: Activity Simulation

As each activity is enacted through the simulator, the enactment data is stored and the non-conformance engine uses it in non-conformance detection analysis by applying the defined rules and rule sets. Once the simulation is set to conclude, the "End Process" button shown in Figure 29 can be clicked to end the simulation of the process enactment. The non-conformance instances detected by the detection engine can also be viewed by clicking the "Process Reports" button also shown in Figure 29. This report presents all the non-conformance instances detected by the non-conformance detection engine, which is discussed in the following section.

6.6 Non-Conformance Detection Engine

The non-conformance detection engine is coded to run as a daemon which must be running before the simulation engine can begin simulating a process. There is a trigger implemented in the code to notify the non-conformance detection engine every time data is inserted into the enactment data tables. When this happens, the rules and rule sets are checked to see if any should be run on the new data. If so, the rules check the inserted data and if any non-conformance is detected, it is logged and the simulation continues. Once the simulation concludes, a report of all the non-conformance instances detected over the enactment can be generated through the simulation engine. A screenshot of an interface from this report is depicted in Figure 31.

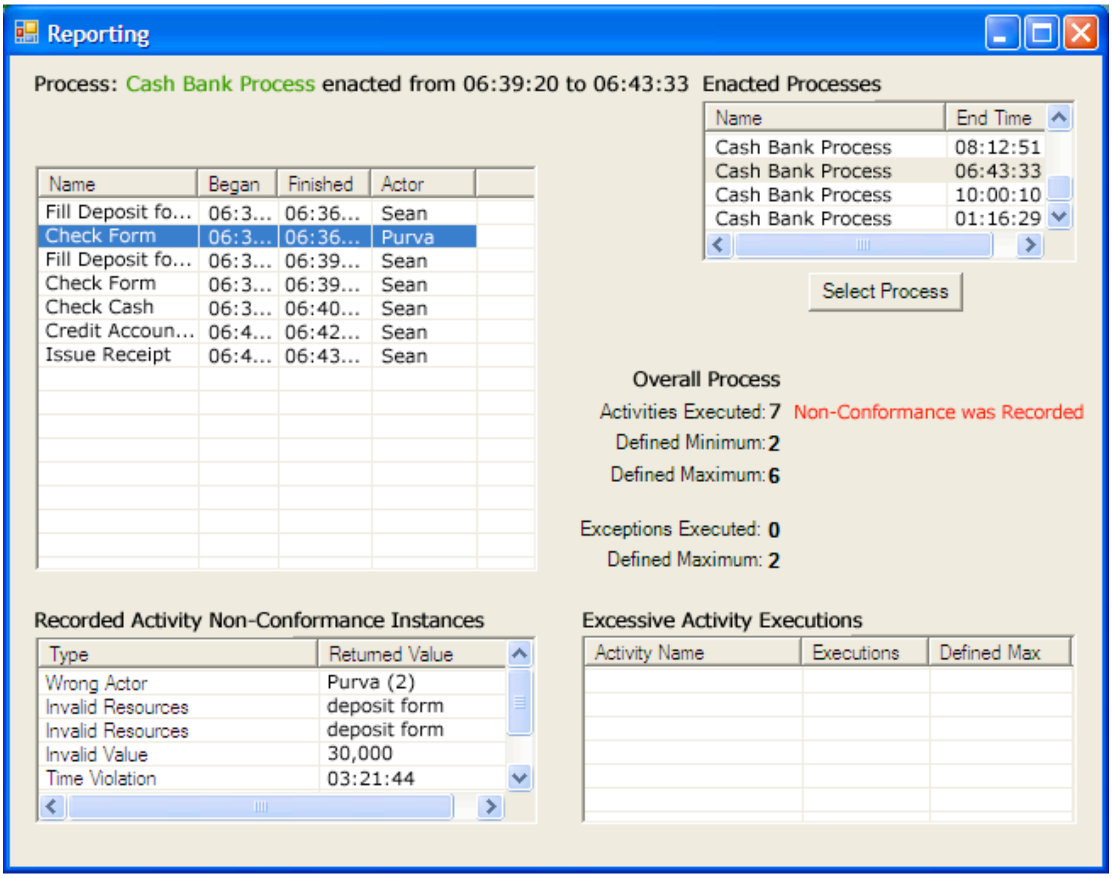


Figure 31: Non-Conformance Reporting

This interface was created in order to give the user a brief rundown of what non-conformance instances were detected using the rule sets applied in this implementation of the GARDEN methodology. In real world cases, thorough examination of the non-conformance logs would be required in order to determine an appropriate course of action for improving the process or to mitigate any non-conformance caused consequences. This screen however, provides a concise snapshot to the user of how the process was enacted and what non-conformance instances were detected along the way.

In the top right hand corner, users can select the different processes that have been enacted. When selected, the process activities will show in the activities pane on the top left of the interface, which show what activities were enacted during the process, who enacted them and their start and finish times. When an activity is highlighted in this pane, the non-conformance instances relating to that activity are shown in the pane directly below, with some enactment data values which show what caused the non-conformance to be detected.

If any auto-generated rule sets were defined that place a limit on how many times a particular activity may be executed during the process enactment, and that limit was broken, then it is reported in the pane on the bottom right of the interface shown in Figure 31. Above this pane are some general details about the process enactment, showing how many activities in total were executed along with the minimum and maximum set (if these rule sets were implemented by the user), also how many rule sets marked as exceptions were executed, and the defined maximum exception count for the process as set by the user.

6.7 Test Case

In this section, one of the simulated processes is covered to show how the environment was used to define the process, define the rules and rule sets that were applied to the process and how it was simulated to detect non-conformance instances. Although many processes were simulated using this tool during the evaluation and testing of GARDEN, a reasonably simple and easy to understand process is described here as an example to demonstrate how this tool works, and how it is useful in verifying the GARDEN approach.

The process is informally described with a simple flow chart (Figure 32) to show a basic specification of how the process would normally be enacted. This specification is then cut down and simplified by removing any aspects of the process that are not relevant for the GARDEN approach. In the interests of clarity, only the activities and properties to which GARDEN is applicable are depicted in Figure 33. Some non-conformance rules that would be reasonable to impose on the process are then identified. These rules are then grouped into the formal rule sets that are used by the environment to show how it handles the process and detects the non-conformance instances that occur during its simulation.

6.7.1 The Test Process Basics

A simple, every day, easy to understand process was selected for this example which should make it easy to see where non-conformance might occur and how it can be detected. The process is the depositing of cash at a bank. Normally, this is carried out by taking a sum of money in the form of cash to the bank, filling a deposit form, and then handing both the form and the cash to the teller. The teller should then credit the account and hand the customer a receipt. When we break such a process down atomically, we can see other facets and considerations involved also. These additional process considerations are depicted more clearly in Figure 32.

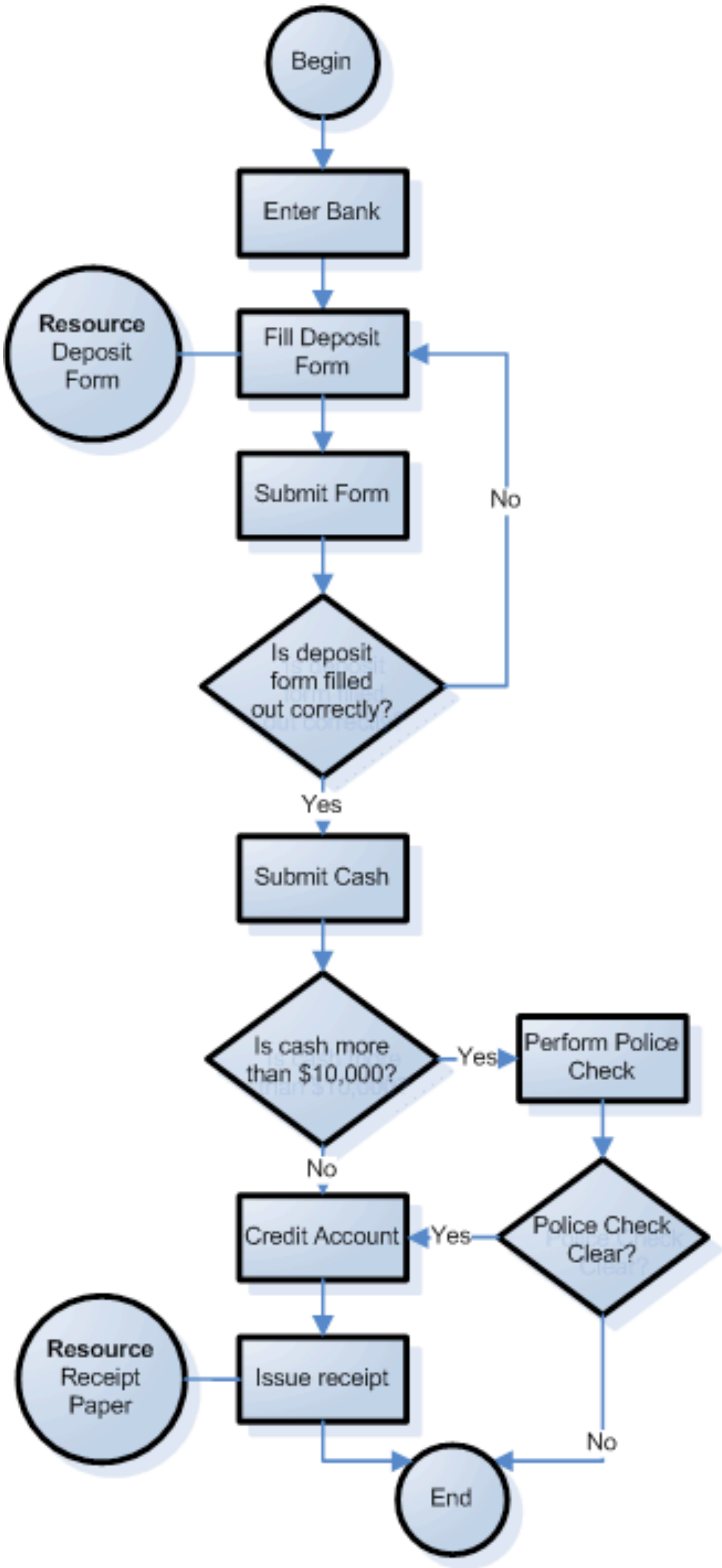


Figure 32: Bank Deposit Process

As seen in Figure 32, there are two expected decision checkpoints that would be made while performing this process. The first is checking whether or not the deposit form has been filled in correctly. The second involves a security tactic commonly used in Australian banks. If a person attempts to deposit more than $10,000AUD in cash at any time, a cursory police check is issued on the customer depositing the money. There are various reasons for this, but all we care about as non-conformance detectives is that the activity exists within the process, and can occur while the process is being enacted.

6.7.2 Identifying the Activities

Some of the entities in this process can be specified as activities whereas some will be omitted. When describing a process logically, steps such as entering the bank are considered, however when the objective is non-conformance detection steps of this nature need not be considered as process activities. Instead, if we break this process down, as it would appear in a process specification fit for the application of the proposed approach, we may end up with something as shown in Figure 33.

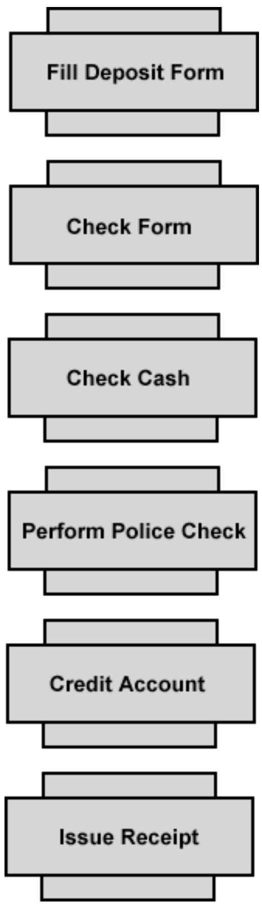


Figure 33: Bank Process Structure

The structure depicted in Figure 33 is a more basic process specification. The actions such as resource usage and police checks can be implemented as rules that check for non-conformance once the process has been properly specified in the process definition engine.

From this point, a process can be created for depositing cash and given an appropriate name and description, the activities shown in Figure 33 can also be defined as part of the process.

6.7.3 Identifying Rules

By studying the basic process in Figure 32, some of the more obvious rule checks become self-evident immediately. For example, once the process has been adequately defined and stored in the environment, it is clear that a minimum of five activities should be enacted while performing this process. Since the police check is not always obligatory, if it is not performed, a minimum of five activities that must be performed if the process is to be completed successfully. Entering values such as this is facilitated by the interface previously shown in Figure 24.

Since the police check activity is not always obligatory, we can now ponder what conditions might arise that would warrant this activity to be enacted or not. Figure 32 shows that this activity is usually performed only when a person attempts to deposit a sum of cash greater than or equal to $10,000AUD.

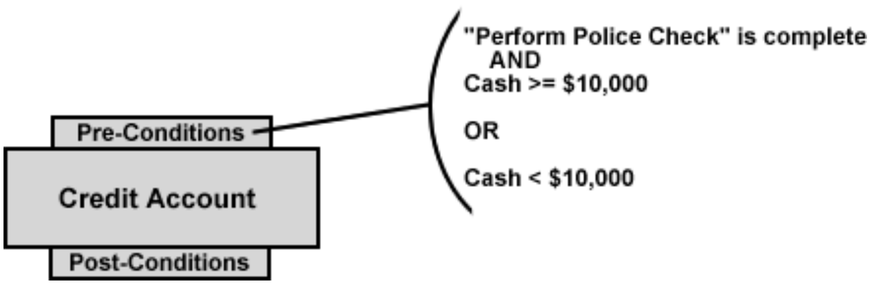


Figure 34: Credit Account Pre-Condition Example

A list of the rule sets applied to this process is defined following in section 6.7.3.1.

<u>6.7.3.1 Rule Sets Applied to the Bank Simulation Process</u>

The rule sets defined and applied to the bank deposit processes described in this section are not exhaustive for the process. In fact, in the real world, depending on the objectives of the organization and the process, new rules may be added and others may be removed as the process matures and evolves. The rule sets described are simply there to illustrate the ease with which they can be defined and applied to the process, and also how these rule sets can be practically used to detect non-conformance instances occurring in an easy-to-understand process.

6.7.3.1.1. Fill Deposit Form

Preconditions:

- None

Consistency Rules:

- None

Post-conditions:

- None

6.7.3.1.2. Check Deposit Form

Preconditions:

- None

Consistency Rules:

- Details Filled Check {is_exception: false; Logic-Type: AND}
    - Account number filled
    - Account number correct
    - BSB Filled
    - BSB Correct
    - Name filled out
    - Amount Entered

Post-conditions:

- None

6.7.3.1.3. Check Cash

Preconditions:

- None

Consistency Rules:

- Check Cash with Deposit Slip
    - Ensure amount handed over matches deposit slip

Post-Conditions:

- None

6.7.3.1.3. Perform Police Check

Preconditions:

- Deposit Amount Check {is_exception: false; Logic-Type: AND}
    - Deposit Amount is >= $10,000AUD

Consistency Rules:

- None

Post-Conditions:

- None

6.7.3.1.4. Credit Account

Preconditions:

- Pre Credit Check {is_exception: false; Logic-Type: XOR}
    - Deposit Amount is < $10,000AUD
    - Deposit Amount is >= $10,000AUD AND Police Check is Complete

Consistency Rules:

- None

Post-Conditions:

- None

6.7.3.1.4. Issue Receipt

Preconditions:

- Credit Account Activity Check {is_exception: false; Logic-Type: AND}
    - Credit Account Activity completed successfully

Consistency Rules:

- None

Post-conditions:

- None

6.7.3.1.5. General Process Consistency Rule Sets

Consistency Rules:

- Activity Executions {is_exception: false; Logic-Type: AND}
    - o Number of Total Activities must be >= 5

- Total Time {is_exception: false; Logic-Type: AND}
    - o Process Execution time <= 8 mins
    - o Process Execution time >= 2 mins

6.7.4 Applying Rule Sets

The rules that may be applicable to this process now need to be structured in tune with GARDEN so that they may be applied to the process. Fortunately, the simulation environment takes care of most of this for us; we only need to use the interface to specify the constraints and boundaries we wish to apply. The rule set model for the example described in Figure 34 is depicted in Figure 35.

Rule-set	
Name:	Pre-Credit_Check
Description:	Ensure it is ok to credit the account with the money
Application-Type:	Pre-condition
Process_Section:	Activity - Credit Account
is_Exception:	False
Logic-Type:	XOR
Rules	
cash < $10,000	
cash > $10,000 && Police check complete	

Figure 35: Pre-Credit Check Rule Set Example

It was in this particular implementation that we discovered that in order to implement the approach, the cash/police check rule had to be implemented as “cash > $10,000 && police check complete” in order to fulfil the parent OR rule, cash < $10,000.

This is not ideal, because the rule itself is now not completely atomic because the one rule must evaluate two separate conditionals. After much consideration on this issue, we found that only a single level of abstraction provided by the rule set, being the logic-type attribute, was practical especially given one of the primary objectives of the approach: to keep it practical. In other words, it is easier and more practical for a user of GARDEN to combine the evaluation of multiple conditionals into a single rule than it is to be forced into dealing with multiple deeper levels of further rule set abstraction.

This concession theoretically means that each additional layer of logic must be implemented at rule level, i.e. if double conditionals are required, then the one rule will be built to check both. Realistically, this is not very likely for most processes, but if it does occur, it can still be handled. It was decided after much deliberation and experimentation that the one level would suffice in the interests of practicality. Additional layers of logic abstraction would unnecessarily complicate the model of the approach and make it much more difficult to use and understand, for only a very small minority of processes. This issue was explained in detail in chapter 4 section 4.3.3.6.

6.8 Evaluation of the Process Data

In order to ensure that the rule sets defined for the process were appropriate, non-conformance instances were deliberately performed during the enactments using the simulator. For example, simulations were conducted at three different time intervals in durations of both less than 2 minutes, between 2 and 8 minutes and more than 8 minutes, to ensure that the general process rule set constraining time duration fired and non-conformance was logged.

Once the process simulations had completed their enactments, the process enactment data returned by the process and recorded in the database was examined. The rule queries were checked to ensure they were properly applying the rules and rule sets and the rule sets were checked against process enactment data manually to ensure that non-conformance was indeed being detected when it had occurred.

As will be shown in chapters 7 and 8, analysis of the non-conformance data can produce some valuable process improvement knowledge. First, it confirmed that the

SPC boundaries on time that can be calculated using this tool are unreliable unless a very large amount of data is available. This is something that is not particularly practical and useless when attempting to set boundary values on a new process with little enactment data. Second, if non-conformance instances are being detected because a rule set constraining boundary values is set at an inappropriate level, i.e. the boundary value is set too high or too low, this can still be beneficial. Potential issues with the process itself can be uncovered (relating to previous enactment data values that caused the boundary value to be set at an inappropriate level) and these erroneous values can be used to further improve the rule sets resulting in a better GARDEN implementation.

A more detailed and comprehensive analysis and evaluation of the GARDEN approach as a whole is provided in chapter 8.

6.9 Conclusion

The simulations of the processes are confined to the developed environment and the data resulting from them is constricted to the build of the component engines and the data storage structure. Therefore, a real process with real process enactment data is still required to fully evaluate the validity of the proposed approach. This, along with showing how the proposed approach can actually help improve the process is presented next in chapter 7.

An environment used to simulate and observe different processes, and then detect and record instances of non-conformance has been presented in this chapter. The backend storage schema has also been covered along with the logic of how this environment was set up and used. Finally a test process has been presented to show how the environment can be used to confirm that the proposed approach has merit and can detect non-conformance whilst addressing the concerns of the issues presented in chapter 3.

The next chapter takes the approach further by presenting a real world process in a different environment and also showing before and after improvement phases to demonstrate how the proposed approach can be used to improve a process.

References

[1] S. Thompson, T. Torabi, P. Joshi, A Framework to Detect Deviations During Process Enactment, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

[2] J. Banks, J.S. Carson, Discrete-Event System Simulation, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

Chapter 7 – Case Study 2: Real World Online Process

7.1 Introduction

This chapter presents the details of another implementation of the GARDEN process non-conformance detection solution, accompanied by several key objectives:

- to supplement the case study presented in chapter 6 with a real world implementation with real users;
- to further test the GARDEN solution robustly, many times against many different users;
- to evaluate a large number of process enactments from both before and after GARDEN has been implemented to provide concrete data as evidence of how GARDEN can be used to improve a process.

This chapter presents the details of a secondary implementation of GARDEN, following the initial simulation-based implementation presented in chapter 6. The chapter begins with the motivations behind carrying out a second implementation, which include the goals for this case study and how it is used to further justify GARDEN as a viable non-conformance detection approach. An overview is then provided, which gives a brief description of the case study, how it is implemented, and the reasoning behind implementing GARDEN into the case study this way.

Following the overview and introductory sections is an explanation of the "market" (World of Warcraft players) at which this case study is aimed in section 7.4. An explanation is provided as to how this particular community is well suited to the goals of this implementation, and why this is a good environment for a case study for testing the GARDEN approach.

Section 7.5 deals with the process of the survey itself, and explains what the process of completing the survey involves. The technical details of the implementation of the survey environment are then described in section 7.6. Section 7.7 provides the details of the actual GARDEN application in the process environment, including the defined rule sets, why these rule sets were chosen and what they are designed to check.

In section 7.8, we analyse the data collected over the course of the case study. This includes the logged non-conformance instances, information on the people involved in the case study, and how the data shows improvement in the process as a result of implementing GARDEN. Raw data, as well as summaries and explanations of the findings and how they relate to the data are provided in this section. Finally, the conclusions are presented, after which the evaluation of the entire approach is presented in chapter 8.

7.2 Motivation for this implementation

The purpose of this implementation is to test the validity and applicability of the GARDEN approach in a real world setting. A critical objective is to explore the potential real benefits that non-conformance detection can deliver and how such detections can be used to create a real and measurable improvement in a process. Outlined in this section are several concerns that have been addressed while considering the implementation presented in this chapter.

First, it is necessary that the target process for this case study is capable of producing an adequate amount of enactment data so that it can be evaluated properly. As discussed in previous chapters, the acquisition of process enactment data is not an issue facing the research goals themselves but it is still a vital challenge to a non-conformance detection approach because enactment data is critical in detecting non-conformance. All things considered, an appropriate process for this implementation should facilitate the gathering of enactment data as easily and as timely as possible.

Another goal for this implementation is to use non-conformance instances detected by GARDEN as a basis for changing the process and then measuring the impact of those changes. To achieve this, a first phase can be instigated where the process is run as normal with GARDEN applied. Using only the non-conformance instances (and their associated data) logged by GARDEN during this phase, process changes can be made solely using the knowledge gained from the GARDEN implementation. The process may then be run again in a second phase, also with GARDEN applied. Comparing the non-conformance data from both phases gives an indication of how GARDEN has helped make real improvements to the process. It is therefore important that this data is as consistent and accurate as possible, so that the data from both phases of running the process can be compared and measured.

To ensure both that the GARDEN approach works as expected and also to facilitate a solid comparison between each implementation phase, a repetitious process is required in order to gather a large enough pool of enactment data to properly scrutinise and identify trends. The process would need to be enacted multiple times in both phases in order to generate enough enactment data to facilitate the goals for this implementation.

7.3 Case Study Overview

Taking into account the motivations for this implementation and the considerations affecting it, an online process seemed like a suitable option. An online environment would open the process up to countless different participants across the world. Through some clever backend code, every action the user made and when they made them whilst at the site of the process would be monitored and recorded until the process was complete or until the user left the site. My professional background is in web development, specifically in PHP, so an online process was the logical choice.

The only problem was obtaining access to a big enough community of users that could be enticed to participate over both phases of the process. The online game of "World of Warcraft" provided this opportunity, that is, not so much the game itself as the enormous community of people associated with it. A forum on the official World of Warcraft website [3] provides a medium for millions of active subscribers to come together and discuss the game. The posting of forum threads attempting to elicit players to take part in the process was possible on the forums provided there was an active, paid subscription to the game, which I had. It also provided access to the player base for us to research, provided we had an active subscription to the game.

Given this opportunity, the decision was made to construct an online survey based around World of Warcraft. The process of filling out the survey according to the instructions would be the process specification and the actors performing this process would be the players who had visited the site. Threads on the World of Warcraft forums were then posted in an attempt to entice active players to the site and fill out the survey.

The survey was implemented in two phases. In the initial phase, players came to the site through a link posted in the forum threads and filled out the survey. Enactment data was observed and recorded for each enactment and compared with some pre-defined rule sets. Once enough submissions were collected, the results of the non-conformance instances were analysed. Based on this analysis, some small changes were made to the survey and the process of filling out the survey. The next phase then began in much the same way, with participants being drawn to the survey via a link in the forum threads posted on the World of Warcraft forums. Enactment data was again observed and recorded from the new improved process and the results were compared against the first phase results. The objective was to measure exactly how "improved" the new process had become as a result of the detected non-conformance instances.

The survey, the rule sets specified, non-conformance instances, and the analysis of results from this study are the focus of this chapter.

7.4 World of Warcraft

In order to understand the implementation presented in this chapter, some background knowledge of World of Warcraft is required because the survey questions in the process relate directly to the game and the rule sets implemented are also based on some of the game mechanics. It should be noted that at the time this experiment was conducted, the limits and rules stated in this section were accurate as of the state of the game in February/March 2009. Since then, the game has undergone some changes including the addition of an "expansion pack", which was not available at the time this case study was conducted. Therefore, affected rules regarding level maximums, what races can play which classes, and other currently modified game mechanics were not in effect at the time of this experiment.

World of Warcraft was chosen primarily because of its large community base. It is a game from the "Massively Multiplayer Online Game" (hereafter MMOG) genre and is and has been immensely popular. Its active subscriber base is now over 11.5 million people [1] and it has a MMOG market share of over 62% against all other currently played MMOGs [2]. Since the World of Warcraft forums are only open to paid and active players, threads have a little more credibility than other highly spammed forums, so the chances of convincing large numbers of people to participate is higher. Also, since the participants all come from forums where a paid and active account is required, we can assume that each participant should be familiar enough with the game to be able to successfully complete the process and fill out the survey. The immense community provides access to literally millions of people with an active and working knowledge of the subject matter, which is also the subject matter of the process that we had defined.

In the game, World of Warcraft players create their own character that resides in the "world". The player is allowed to customize their character quite freely however there are certain constraints that apply. For example, only certain character "races" can play certain "classes". So, if I wanted to play the "Druid" class, then my character race would have to be either a "Tauren" or a "Night Elf". No other available races have the ability to play this class. It is constraints such as these that are tested in the survey in this implementation.

World of Warcraft characters start out at the minimum level of 1 and this level increases periodically as the players progress through the game to a certain maximum. This maximum level is dependent upon what version of the game the player is running. If the player is running the standard game, the level cap is 60, plus there are certain races and classes that are unavailable to the player. There are two expansion packs that may be bought and installed to run with the game, which are "The Burning Crusade" and "Wrath of The Lich King" respectively. If the first is running, the level cap is increased to 70. If both expansions are running, then the maximum level for the player's character is 80.

These game constraints and intricacies provide us with a medium to detect inconsistencies in process data, as they are forced constraints that people unfamiliar with the game may not know about. It is also a point of reference, which is useful in spotlighting people who deliberately give erroneous answers in the survey. This is shown in the test results in the coming sections.

7.5 The Survey

The process designed is based on a simple online form, containing questions based on the World of Warcraft game. In order to complete the questions satisfactorily, a user would need at least a rudimentary knowledge of the game. We can assume that a great majority of these users should have this knowledge, because participants were enticed to the survey via the World of Warcraft official forums [3] which requires an active and paid game account.

A screenshot from the survey is provided in Figure 36. This screenshot was taken after the second phase, when the “improved” version of the process was in effect based on the analysis of the first phase.

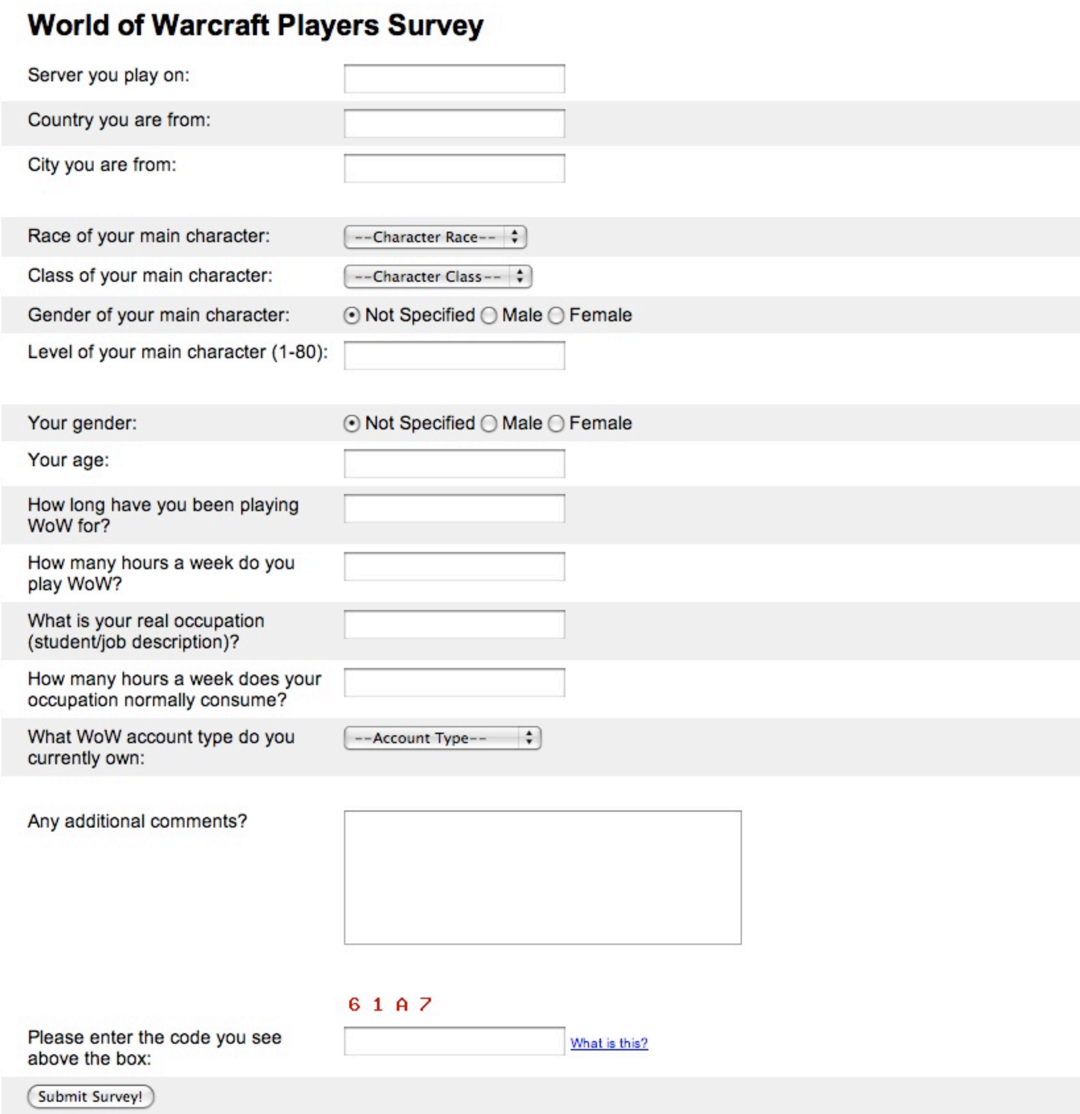


Figure 36: Screenshot of Survey

Participants had been led to believe that the researchers were interested in the response data when filling out the survey. However, what we were actually interested in was the process of *how* they filled it out, that is, what they actually did and when they did it when filling out the survey was observed and recorded. Participants were deliberately misled into thinking the research was concerned with the game itself rather than the process of actually completing the survey to negate any potential Hawthorne effect [6] on the process results.

Silva and Travassos mentioned concerns about the Hawthorne effect relating to participants in research affecting the internal objective of the research if they know about the nature of why they are being observed. This issue and the benefit of disguising the research in such a way are discussed at length in Silva and Travassos [7].

Some of the questions presented are geared to give participants an "opportunity" so that they may not conform to the process specifications. For example, the type of class a person can play will be based on his or her chosen race as certain races can only play certain classes; but the select box on the survey provides the facility to choose all races and classes. Mismatching them should cause a non-conformance instance when the data is tested within the framework if there is a rule set implemented to detect such non-conformance instances. Also, we can test other aspects such as boundary values on the character levels (1-80 depending on the account type) and the person's claimed real age. The actual rule sets specified and implemented into this case study are explained further in section 7.6.

Once the survey questions have been completed, the survey needs to be validated via a captcha code. Captchas are a form of the Turing test [4] used to prevent non-human users such as worms and bots from submitting form data. Solving the captcha code is a task which humans should easily be able to pass, whereas current computer programs cannot [5]. This activity is to validate that the user has completed his or her answers and is ready to submit them and exit. As such, this should be the final activity completed in the process.

The final activity in the process is simply clicking the "What is this?" link which refers to the captcha code. For participants unfamiliar with the use of captchas, the link will open a new window displaying some text explaining what the code is used for. Since embarking on this activity implies that the participant is unfamiliar with captchas, it should only be started before beginning the captcha validation activity.

7.6 Implementation of the Survey and the Approach

The survey was implemented in PHP 5 in a LAMP environment with all data being stored in a MySQL 5 database. This includes all the mechanics of the survey itself, the capturing and recording of enactment data and the implementation of rules and rule sets. So the entire process and the GARDEN non-conformance detection approach are implemented in the same LAMP environment on the web server.

Session cookies along with IP addresses from each visitor are used to identify individual survey users, including their user agent and platform. Every action the user takes including failed attempts, navigation and visit/exit timestamps were recorded and stored in a relational database. This information clearly shows the actions of the user the entire time they are participating and performing the process. It is important because it allows for easy determination of what activities they were performing, when they performed them, for how long, and exactly what they did. This information can be used to have the rule sets applied for non-conformance detection purposes.

Two distinct MySQL 5 databases were used for each phase of the implementation. After this experiment was completed, a third dummy database was hooked into the survey and the two used databases were collated and archived. The purpose of doing this was only to maintain a useable version of the survey process that could be used for future scrutiny. However the data from this ongoing version would not pollute the data from previous versions used to conduct this experiment. A test version of this survey that is hooked into the dummy database is still available online at http://wowsurvey.nostin.com.

We wanted to be sure that every answer given by a user was intentional, so every question in the survey begins with a blank answer box, thus not forcing an answer. Controls such as the radio and select options on the site, which normally have a response automatically selected have been set with a "not-specified" option as default. This measure ensures that users have to manually select answers to questions. If they do not, the "not-specified" option will be sent through preventing data being captured that the participants did not manually set themselves.

If the participant fails a survey attempt i.e. enters the wrong captcha code, the survey will retain his or her answers so the user does not have to fill out all the answers out again. The survey screen will simply reload with the content of the form remaining for the participant to address. This measure prevents user frustration and having participants giving up on the process if they fail certain activities, and increases the likelihood of them following through to the end. It is also valuable as it eliminates the need for participants to redo activities they have already successfully completed if they cause non-conformance later in the process.

If the user clicks the "What is this?" captcha link, the target URL will open in a new window, so the user does not navigate away from the page. Use of a captcha [5] is a common technique amongst web developers to differentiate between human and non-human users. A captcha was implemented in this case to help weed out non-human responses to the survey. No non-human responses to the survey have been included in any of the results or analysis presented in this case study. All non-conformance instances, result data and analysis data come from real human responses to the survey.

7.7 Process and Rule Set Specification

The process performed in this case study is the act of the participants filling out and submitting the online survey that was constructed. Once the survey has begun, it counts as an enactment of the process regardless of whether or not it was completed or whether or not it was correct. This section explains the process in full, including each activity and what completing the survey entails along with each rule and rule set specified for the whole process.

The process of completing the survey was used here as it is relatively simple. It is outlined in Figure 37.

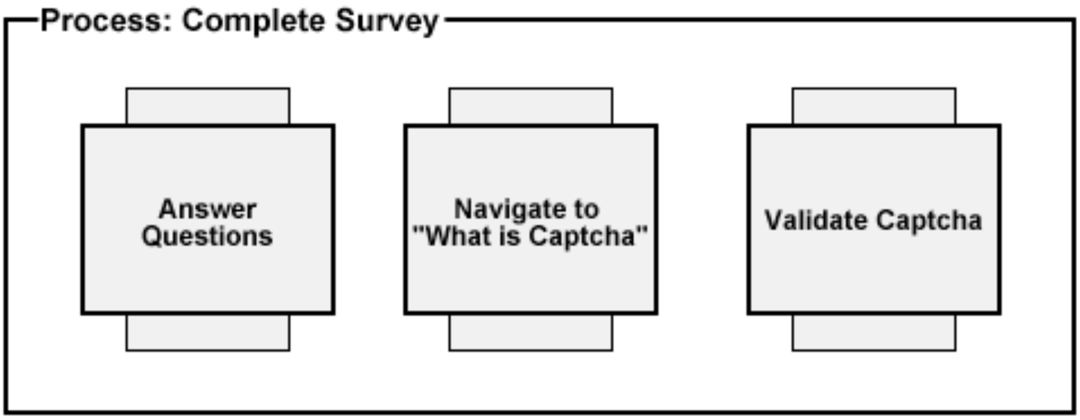


Figure 37: Process Outline

As shown in Figure 37, the process has three distinct activities described in the specification:

1. answering the survey questions;
2. validating the survey Captcha;
3. navigating to the Captcha explanation page.

Here the activities of the process are explained along with the rule sets specified to detect non-conformance to both the activities and also the process as a whole. These rule sets were implemented in PHP code interacting with enactment data stored in a MySQL database, but for the sake of clarity, we present the rules here in structured plain English.

Each rule set is displayed here as a bullet point, with each of its associated rules as a sub-bullet point underneath its rule set bullet. In the interests of brevity, the description is omitted for each rule set with only the name identifying it. The rule sets are given activity by activity, and then finally, for the general process so it is evident where the rule set applies. For each activity, rule sets are specified by pre-condition, consistency, post-condition to denote the application-type for the rule sets (how the rule set is applied to the activity, i.e. as a pre-condition, consistency or post-condition). In the bullet points for each rule set depicted in this section, braces '{}' follow the rule set name that show whether or not the rule set is an exception and what logic-type the rule set is following.

7.7.1 Answer Questions Activity

This activity is concerned with the answering of questions pertaining to the game itself and the player. Due to certain inbuilt game constraints, we can adopt similar restrictions on certain answers to questions. For example, certain game character classes can only be matched with certain game character races. If a player enters the race "Orc" and the class "Warrior", then this is acceptable and not non-conformance because it is possible within the confines of the game. However, the race "Orc" and the class "Priest" are not compatible. So, if the participant enters this race/class paring, then non-conformance will be flagged.

Also, the level range provided in the game is from 0 to 80 if all expansions are active. If no expansions are active, the maximum level is 60. Since game constraints like this are known to us, we can write rule sets to check if the survey participant conforms to them. If a user enters information that is not compatible with the constraints of the game, this information is non-conformant and we can create rule sets to cater for this.

The rules and rule sets defined for the "Answer Questions" activity is as follows:

Preconditions:

- None

Consistency Rules:

- Level Rule-set {is_exception: false; Logic-Type: AND}
  - Must be numeric
  - Must be greater than 0
  - Must be less than or equal to 80
- Account Level Rule-set {is_exception: false; Logic-Type: AND}
  - If no expansions are active, level must be less than or equal to 60
  - If one expansion is active, level must be less than or equal to 70
- Age Rule-set {is_exception: false; Logic-Type: AND}
  - Must be numeric
  - Must be greater than or equal to 3
  - Must be less than or equal to 100
- Race/Class Rule-set {is_exception: false; Logic-Type: AND}
  - If race "Dranei" is specified, class must be one of types: Hunter, Mage, Paladin, Priest, Shaman, Deathknight or Warrior.

- If race “Dwarf” is specified, class must be one of types: Hunter, Paladin, Priest, Rogue, Deathknight or Warrior.
- If race “Gnome” is specified, class must be one of types: Mage, Rogue, Warlock, Deathknight or Warrior.
- If race “Human” is specified, class must be one of types: Mage, Paladin, Priest, Rogue, Warlock, Deathknight or Warrior.
- If race “Night Elf” is specified, class must be one of types: Druid, Rogue, Priest, Hunter, Deathknight or Warrior.
- If race “Blood Elf” is specified, class must be one of types: Hunter, Mage, Paladin, Priest, Rogue, Warlock or Deathknight.
- If race “Orc” is specified, class must be one of types: Hunter, Rogue, Shaman, Warlock, Deathknight or Warrior.
- If race “Tauren” is specified, class must be one of types: Druid, Hunter, Shaman or Warrior.
- If race “Troll” is specified, class must be one of types: Hunter, Priest, Rogue, Mage, Shaman, Warrior or Deathknight.
- If race “Undead” is specified, class must be one of types: Mage, Priest, Rogue, Warlock, Warrior or Deathknight.

- Account Race Type {is_exception: false; Logic-Type: AND}
    - If race Dranei is specified, Account type specified must have at least one active expansion.
    - If race Blood Elf is specified, Account type specified must have at least one active expansion.
- Account Class Type {is_exception: false; Logic-Type: AND}
    - If Class Deathknight is specified, Account type must have both expansions active.

Post-conditions:

- Questions Complete Rule-set {is_exception: false; Logic-Type: AND}
    - At least one question must be answered (either a text box filled in or one of the radio or select boxes not be unspecified).

As we can see, especially with the consistency rule sets, some liberties must be taken with the constraints. Some are steadfast – we know that character levels must be between 0 and 80, but some are arbitrary – it is possible that a 100-year-old person might play World of Warcraft and answer our survey, but it is the boundary value we have chosen. If either of these rules return false, all it means is that non-conformance was picked up by the system, and we then know to manually check the offending record.

7.7.2 Validate Captcha Activity

This activity validates the completed survey before submission. Normally, the sole purpose of these types of activities is to verify that the response from the user is human and genuine. The same applies here.

Preconditions:

- None (The completion of the answer questions activity was handled in that activities post-conditions)

Consistency Rules:

- None

Post-conditions:

- Check specified Captcha {is_exception: false; Logic-Type: AND}
  - Captcha code must match user entered code (case insensitively)

7.7.3. Navigate to "What is Captcha?" Activity

Despite the use of captcha-like Turing tests being all the more common nowadays across the web, tt is also the norm for sites that implement them to have an explanation. Some people still do not know what they are or why they are used, so a link has been provided in order to explain their usage. The only rule implemented was to make sure the user did not first submit the captcha code before checking what it was, as doing so would cause the page to redirect. This rule set was specified because it is illogical to validate the captcha and then learn about what its function is afterward.

Preconditions:

- Check Captcha Activity Status {is_exception: false; Logic-Type: AND}
  - Validate Captcha activity must not have been enacted

Consistency Rules:

- None

Post-conditions:

- None

7.7.4 Overall Process Rule Sets

Since the process is regarded as a whole entity, pre and post-conditions do not apply. Only consistency rule sets are defined. They were:

- Captcha Fails. {is_exception: false; Logic-Type: AND}
  - Validate captcha activity must not be failed more than once.
- Total Time Taken {is_exception: false; Logic-Type: AND}
  - Must be greater than 15 seconds.
  - Must be less than 3 minutes.

Only two non-conformance types required checking concerning the process as a whole and these types were covered adequately via the use of two rule sets. The first checked to see if the captcha had been failed more than once. Once the survey has been completed, the validation captcha is entered. If it is entered incorrectly, the contents of the survey simply refresh with a new captcha code to be entered and a message telling the participant to check the code and try again. If the participant fails this a second time or more times, this is considered non-conformance. The other rule set applied was to check if the participant took an inordinate amount of time to complete the process. This was capped at a minimum of fifteen seconds and a maximum of three minutes.

7.8 Analysis of the enactment data

Over the entire duration of the experiment presented in this chapter, the survey was completed a total of 503 times by 410 different people over two approximately evenly spread implementation phases. This dataset size is not relatively large compared to other datasets used for analysis, however it is suitable for this purpose. Many processes that are eligible to be subjected to the proposed approach will never be enacted as many times as was recorded in this case study. Furthermore, the dataset is large enough to confirm non-conformance patterns and indications of improvements based on analysis of the data over both phases without aberrations interfering with the analysis.

In the first phase, 213 different people filled out the survey a total of 286 times. In the second phase, 197 different people filled out the survey a total of 217 times. This section first looks at the enactment data from phase one and the non-conformance instances that were detected from the rule sets applied. Analysis of these non-conformance occurrences is performed and some basic process changes intended as improvements based on the results are presented. The phase two results of the implementation are then analysed with the non-conformance instances detected in phase two to see if the changes had any benefit on the process.

7.8.1 Phase One Enactment Analysis

From the rule sets presented in section 7.6 of this chapter, the following non-conformance instances were detected in phase one:

Rule Set Broken	Process Area	Type	Occurrences
Account type inconsistent with character level	Answer Questions Activity	Consistency	1
Age	Answer Questions Activity	Consistency	25
Character Level	Answer Questions Activity	Consistency	20
Race and Class mismatch	Answer Questions Activity	Consistency	11
Captcha code and user code mismatch	Validate Captcha Activity	Post-condition	81
Captcha fail more than once	Overall Process	Consistency	16
Overall time duration	Overall Process	Consistency	49
Validate captcha activity must not be complete	Navigate to "What is captcha" Activity	Pre-condition	2
Questions Complete	Answer Questions Activity	Post-condition	5

Table 10: Phase One Non-Conformance Instances

From a dataset the size of 213 people making 286 submissions, there are a few figures in Table 10 that draw immediate concern. The most glaring of these is the captcha code mismatches indicating that a lot of people had significant trouble completing the captcha validation code. Inspection of the enactment data revealed that 52 different people failed the captcha code validation rule set at least one time. This equates to a new participant starting the survey having a 24.4% chance of also causing non-conformance from this rule set.

Not only is the mismatch rate high for captcha validation, so is the number of non-conformance instances relating to additional fails. According to the data, 16 people failed to validate the captcha more than once, meaning 7.4% of participants from the first phase of the survey could not correctly enter the captcha code in two tries or less. Upon closer inspection of the data, it was revealed that two people had failed to validate the captcha code 5 times and another two people had failed to validate it 4 times.

Another area of concern regarding the non-conformance statistics is the number of people who failed to complete the process within the time constraints set by the overall process rule set, which was more than 15 seconds and less than 3 minutes. Upon closer inspection of the enactment data, it was revealed that 4 of these 49 instances were for completing the process too quickly whereas 45 took too long. Additionally, 18 people took over 5 minutes to complete the survey with the three greatest discrepancies being 15 minutes 18 seconds, 20 minutes 30 seconds and 23 minutes 47 seconds.

The other two areas of concern with the non-conformance instance data were the participants' age input and the characters level input. For such a simple thing as inputting age, something that is a common query on many online forms across the web and the level of your character, it seemed quite a disproportionate number of people entered values that were unacceptable according to the non-conformance detecting rule sets that were applied.

7.8.2 Improvements implemented

This section is concerned with addressing a few specific parts of the process from the first phase. The parts that are addressed stem only from the non-conformance data, so that each change can be sourced directly to non-conformance detections from the first phase of the implementation. These changes are subtle, not major, so the process is essentially the same as before but the changes made are measurable. In this way, we can compare the data from the first phase of enactments to the second phase to test whether these changes caused any noticeable or measureable improvements in the process.

7.8.2.1 Validating the captcha code

The non-conformance data concerning both the validation of the captcha code and the time taken to complete the process was the priority in improving the process for the second phase. It was extrapolated from the non-conformance data because there seemed to be a lot of trouble validating the captcha code, and because the number of people who failed to validate it on more than two consecutive occasions may have contributed to the overly long time people were taking to complete the process. If the captcha code could be improved somehow, then hopefully this would alleviate some of the overall time taken for non-conformance occurrences also. Therefore, the captcha code validation activity was modified in the following ways:

7.8.2.1.1 Reduction of Captcha code characters

Validating the captcha code was one of the biggest non-conformance generating aspects of the process. Since there were no problems with automated bots submitting garbage data through the form, it was considered that the number of characters used in the captcha code could be reduced to make it easier for humans to pass it successfully yet still deter bots. Therefore, the number of characters in the captcha code was reduced from 6 to 4.

7.8.2.1.2 Elimination of lowercase characters from the Captcha code

Upon closer inspection of the randomly generated captcha codes presented on the survey form, it appeared that there were several characters that could potentially confuse participants. For example, the lowercase ‘l’ can look similar to the uppercase ‘I’ or even the number ‘1’. It was possible that survey participants were confusing similar looking letters and getting the captcha code wrong. Removal of all lowercase characters leaving only digits and uppercase letters was considered to alleviate the potential for this problem to occur.

7.8.2.1.3 Elimination of ‘O’ and ‘0’ from the Captcha code

In addition to the removal of lowercase letters as stated in section 7.7.2.1.2, it was decided that the uppercase ‘O’ and the number ‘0’ should also be removed for the same reason. The two characters look too similar and could cause problems for participants to distinguish them.

7.8.2.2 Age and Character Level Inputs

On inspecting the enactment data relating to age and character level, a few things became apparent. People were triggering the non-conformance detection rule sets in three different ways with these inputs.

The first was the input of characters. Both these fields have rule sets which check for numeric only input, so specifying age or level using text triggered non-conformance detections. This was made possible because there were no “maxlength” attributes set on the form fields, so participants could submit values consisting of long strings.

Another common occurrence was that participants deliberately and obviously submitted erroneous data. One participant, for example, claimed he was 400 years old. This value triggered the rule set stipulating that ages must be numeric values of less than 100. Other participants enjoyed filling the survey with similar nonsense that triggered non-conformance detections.

The third and far less common cause was simply innocent users unaware of the non-conformance detection mechanics, inputted values they thought to be helpful. One participant input the value for their character level as “80 (alt is 72)”, for example. This person was probably specifying their main characters’ level, but also claiming that they had an alternate character with a current level of 72.

7.8.2.2.1 Addition of the maxlength Attribute

In HTML text fields, there exists an attribute called “maxlength” that prevents users from entering more values into fields than is stipulated. For both the age field and the character level field, the maxlength attribute was added with a value of ‘2’, meaning participants could not enter a string of more than 2 characters into these fields.

It was felt that this small change was the most prudent course of action. It is subtle, yet it should convey to participants that numeric values only are required since 2 characters is not long enough to enter age or level values textually. It will also prevent users from specifying things such as their alternate character’s level.

This change will do nothing however, to prevent people from deliberately entering erroneous values. However, it was decided that this was a good thing. The garbage data that was entered deliberately was easily detected by the rule sets in place, unbeknownst to the user submitting it. As long as the garbage data is detected, then this is enough. Moreover, making it difficult for people to deliberately enter erroneous data also makes it harder to detect. If we were genuinely interested in the submitted values, the fact that we were comfortably detecting the malicious participants demonstrated that it is easy to weed them out at the end. So, it was better not to attempt to prevent people from doing this.

7.8.3 Phase Two Enactment Analysis

Once the changes were made to the survey and the process, phase two was implemented. The non-conformance data detected and recorded from phase two is presented in Table 11.

Rule Set Broken	Process Area	Type	Occurrences
Race/Account mismatch	Answer Questions Activity	Consistency	1
Account type inconsistent with character level	Answer Questions Activity	Consistency	2
Age	Answer Questions Activity	Consistency	7
Character Level	Answer Questions Activity	Consistency	8
Race and Class mismatch	Answer Questions Activity	Consistency	3
Captcha code and user code mismatch	Validate Captcha Activity	Post-condition	22
Captcha fail more than once	Overall Process	Consistency	1
Overall time duration	Overall Process	Consistency	24

Table 11: Phase Two Non-Conformance Instances

In order to illustrate the improvements made solely from non-conformance detection from phase one of the process through to phase two, the figures in Table 10 and Table 11 are illustrated for clarity in Figure 38. This graph shows the number of non-conformance instances detected for each rule set that detected non-conformance for each of the two phases.

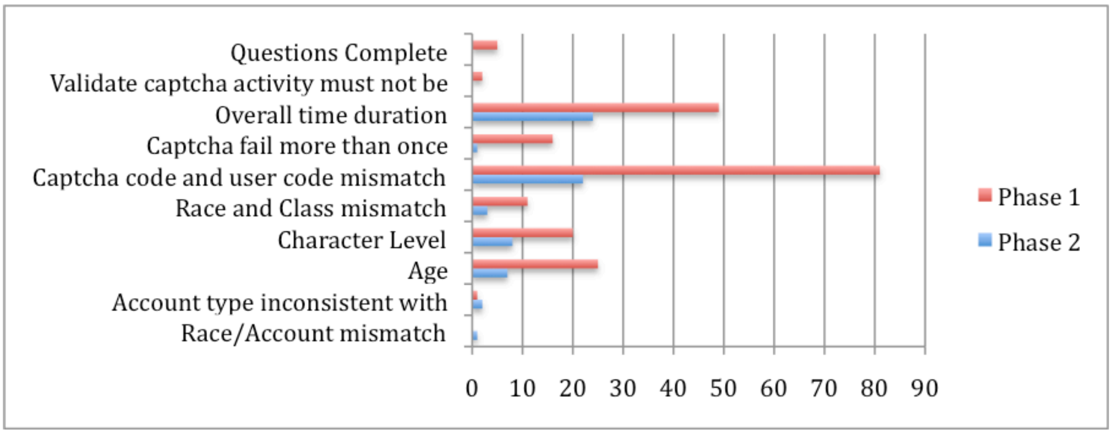


Figure 38: Process Phase Comparison Graph

Although the improvement in the data is immediately noticeable, it should also be noted that phase two had a slightly lower participation rate than phase one. In this phase, 197 people filled in the survey 217 times, so there were 16 less people involved in phase two than phase one.

Nevertheless, the non-conformance data from phase two indicates that the captcha validation failures have been reduced from 28.3% (81 failures from 286 submissions) to 10.1% (22 failures from 217 submissions). Furthermore, the percentage of people who failed the catpcha twice or more and setting off a non-conformance instance from the associated rule set also fell from 7.4% in phase one to now only 0.5% in phase two. Also, only one person in phase two failed the captcha more than once. This is strong evidence to suggest that the measures taken in improving the usability of the captcha from phase one to phase two were successful. The captcha, incidentally, still served its intended purpose as no automated responses managed to sneak through into the database.

Also noticeable in the dataset was the fall in the number of age and character level non-conformance instances, with the bulk of these being from participants eager to corrupt their survey.

Considering the fall in non-conformance instances as a result of the changes made going into phase two, the data also indicated that the overall time duration non-conformance type instances had also dropped considerably. Apart from the changes made to the captcha and the limiting of the input sizes on the age and level fields, no other alterations were made to the survey. So this considerable drop in time violations was most likely due to the improved usability features included in phase two.

Upon a closer inspection of the enactment data, it was discovered that there were 3 instances where participants took less than the minimum 15 seconds to complete the process as compared with 4 in phase one. There were also only 21 instances where people took greater than the maximum 3 minutes to complete the survey in phase two compared with 45 in phase one (the total figures for overall time duration violations of 49 in phase one and 24 in phase two in Table 10 and Table 11 also include minimum time limit violations – 4 in phase one and 3 in phase two). The three greatest time durations were also improved from phase one: 15 minutes 18 seconds, 20 minutes 30 seconds and 23 minutes 47 seconds; to phase two being 7 minutes 1 second, 7 minutes 35 seconds and 7 minutes 54 respectively.

From the screenshot shown in Figure 36, we can see that a comments field is available for participants to complete with any data they like. Another interesting point regarding the ineffectiveness of the captcha code that was detected from the phase one data is that not one single person in 410 mentioned the captcha code difficulties in the comments section of the survey. This implies that the non-conformance detection was vitally important in picking up this weakness in the process as not a single person thought to mention it out of 410 total people. This includes 213 people in the first phase when the captcha was at its worst (and most mentionable) condition.

7.9 Conclusion

One of the most useful features of finding non-conformance instances is the power it gives to weed out bad data along with errors in process specification. Even in a simple survey such as this one, once the response data starts coming through it is surprising how quickly it is to see where people are having trouble understanding or completing parts of it. Even where the process specification is incorrect, through the detection of non-conformance in the process enactments these issues can be picked up surprisingly quickly.

One example of this was the non-conformance rule set we defined stating that the “what is captcha” activity should always be performed before completing the “validate captcha” activity. This makes logical sense in theory, but when the detected non-conformance instances started coming in, they indicated that people were only clicking the “what is captcha” link if they had already had the captcha wrong when trying to validate it, which pragmatically is also a logical thing to do. There were 2 instances of this happening in phase one and none in phase two, suggesting that this non-conformance rule set is unnecessary and would best be removed.

Another useful feature of the non-conformance detection was the ability it afforded us to detect when people were deliberately attempting to sabotage their survey. The “Answer Questions” activity had a rule set defined in its post-conditions that at least one survey question must be answered. In phase one, this occurred 5 times from 4 people. For the most part, we can assume they accidently submitted the survey before completing it because they also failed the captcha. One person however, successfully completed the captcha, indicating that the lack of response was deliberate. In any case, if we were actually interested in gathering information derived from the survey questions, we could easily exclude these results from the findings, as they are easy to spot because the detected non-conformance instances tell us exactly where to look. Interestingly, this did not occur at all in phase two.

This illustrates the importance of having a flexible observatory framework. If we chose a process support methodology which forced the users along a particular route, it would be much more difficult to spot where the errors in the process specification were.

From a technical point of view, despite the ease of implementation in an environment such as the web, it was found that the implementation of GARDEN to this process was straightforward. This was one of the goals set when this implementation was first being considered.

The next chapter follows on from the approach detailed in chapter 4 and its two implementations that were presented in chapters 5 and 6. Here the proposed approach and its effectiveness will be evaluated along with the positive and negative experiences when applying this methodology.

References

[1] Blizzard Entertainment Press Release, "WORLD OF WARCRAFT® SUBSCRIBER BASE REACHES 11.5 MILLION WORLDWIDE", http://www.blizzard.com/us/press/081121.html, 23 December 2008

[2] http://www.mmogchart.com/Chart7.html, MMOG Market Share at April 2008

[3] World of Warcraft official game forums, http://forums.worldofwarcraft.com/index.html

[4] Stanford Encyclopedia of Philosophy, Stanford University, California, USA, http://plato.stanford.edu/entries/turing-test/

[5] Official CAPTCHA Site, http://www.captcha.net/

[6] R. Gillespie, Manufacturing knowledge: a history of the Hawthorne experiments, Cambridge: Cambridge University Press, 1991.

[7] L.F.S. Silva, G.H. Travassos, Tool-Supported Unobtrusive Evaluation of Software Engineering Process Conformance, International Symposium on Empirical Software Engineering, ISESE '04, ISBN:0-7695-2165-7, September 2004.

Chapter 8 – Evaluation of the GARDEN approach

8.1 Evaluation Objective

This chapter provides an objective evaluation of the GARDEN approach to process non-conformance detection. This is achieved in two ways. In section 8.2, the issues listed and described in chapter 3 are revisited. The way GARDEN has addressed each of these concerns is discussed and analysed. The benefits and drawbacks of addressing each issue in this way is presented along with supporting evidence from the case studies and testing of the approach.

In section 8.3, the unanticipated benefits and consequences of approaching process non-conformance detection with the GARDEN methodology is evaluated. This section identifies the additional benefits GARDEN provides that were discovered during testing and the case studies, which were not directly related to the research objectives listed in chapter 3. This also identifies any possible drawbacks that were unanticipated when the solution was designed, marking them as areas for improvement and attention in the future work on this topic.

In conjunction, this chapter provides an overall impact and suitability analysis of the GARDEN approach to process non-conformance detection. The positives, both expected and unexpected are discussed along with some of the realisations that occurred during the implementation and testing of the approach. This includes the limitations and shortcomings of GARDEN along with some ideas for improving the approach and rectifying these limitations. Section 8.4 then concludes the chapter and gives a prelude to the final chapter that will conclude this thesis.

8.2 Evaluation of how GARDEN addresses its stated objectives

In chapter 3 of this thesis, the goals of GARDEN were presented and discussed as issues of concern to process non-conformance detection. Aside from successfully and comprehensively detecting non-conformance, approaches that aim to achieve this need to keep certain other issues in mind in order to maximise the usefulness of the approach. This section readdresses the issues and challenges from chapter 3 in the context of how the approach presented in this thesis met these issues and challenges and how successfully they were met based on the solution, its implementation and testing.

In the interests of objectivity, it should be noted that the approaches GARDEN is compared to in this thesis are described in the context that was provided in the literature in terms of the papers that presented those approaches and other papers that referenced them. Clearly GARDEN is presented from an expert perspective whereas the commentary on other approaches can only be provided from what is available in the literature, so comparisons between the different solutions need to be read with this caveat in mind.

8.2.1 Defining the contextual representation of process non-conformance

The detection of process non-conformance is a relatively young research area. Perhaps not young in terms of the amount of time researchers have been addressing the problem but certainly young in terms of the quantity of work devoted to this area of research over the time period until now. In any case, the idea of non-conformance is still yet to be defined in a steadfast manner. The terms and concepts are still used loosely and inconsistently in the literature, despite previous attempts to try and formalise these in a consistent manner [4].

To address this, a concept map of non-conformance was developed. This concept map is reproduced in Figure 39.

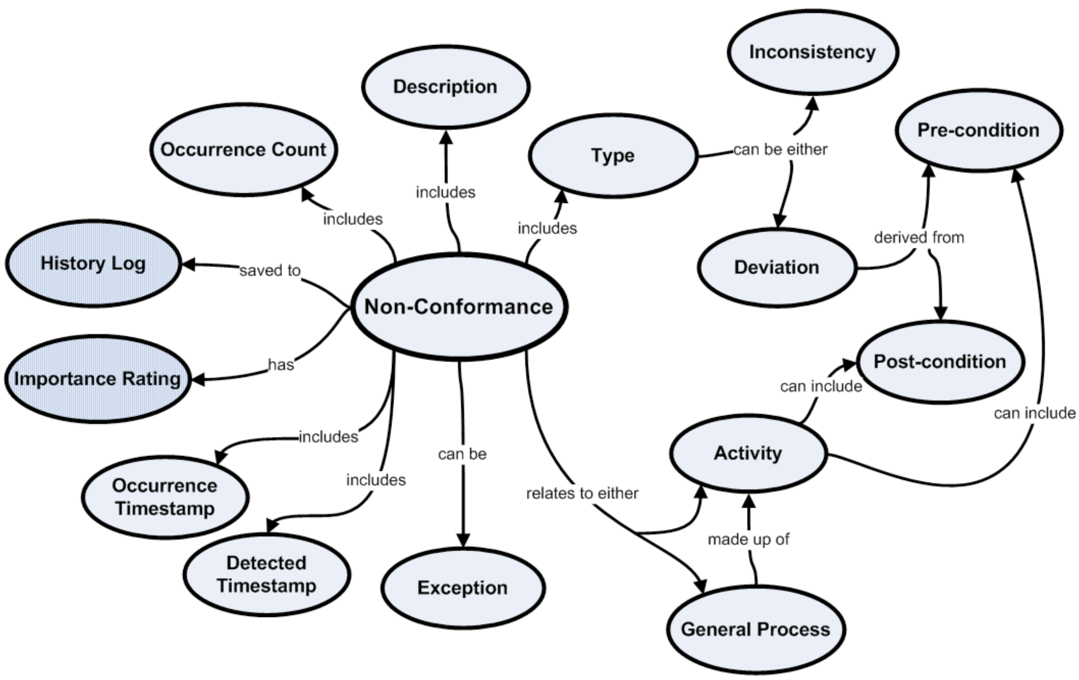


Figure 39: Non-Conformance Concept Map

The concepts included in this representation of non-conformance stem from the work presented in Cugola et al. [4] along with some innovations from both the publications from the proposed approach in this area along with ideas of other researchers also published in this research domain. The aim here was to define non-conformance in such a way that would facilitate the application of the proposed approach, but also frame the concept of non-conformance generically in a way that could be used by different approaches for different processes in different domains.

The way the concept of process non-conformance was presented succeeded in providing a basis from which non-conformance could be extended. This included two representations of non-conformance. The first was a "rule set" concept, which is a description of what is being checked for by GARDEN, that is, the attributes associated with non-conformance that would flag detection when GARDEN analysed the process enactment data. The second representation is the "instance" concept, related to the instance itself and includes attributes pertaining to values apparent when the instance occurred, such as when it actually occurred, when it was detected and the number of occurrences. This is also covered in depth in chapter 4 and depicted in Figure 5 and Figure 8. In conjunction, these two representations form the basis for what the concept of "non-conformance" means, and the attributes and properties associated with it. This is reproduced in Figure 39.

It is expected that the attributes "History Log" and "Importance Rating" in the concept map shown in Figure 39 will be extended into something more formal and practical in determining a methodology for ascertaining non-conformance severity and remedial action recommendations. However, future considerations of non-conformance detection are discussed further in section 8.2.7.

The power of the rule set approach to non-conformance detection is dependent upon the prowess of the domain expert responsible for its implementation. This extends to both their level of knowledge of the process in question and also their skill in identifying what rule sets should be defined to properly constrain the process enactment values. This task is made easier through an interface provided to the domain expert to achieve this, but that person is still ultimately responsible for defining appropriate rule sets.

The challenge of defining an appropriate frame to describe what non-conformance will constitute for a given process is not unique to approaches like GARDEN. Cîmpan and Oquendo [2] require a successful model of a flawless process to be simulated, something that is not always a simple task. Discovery approaches such as Huo et al. [1] and Cook and Wolf [5] require a reference model to be successfully built from previous approaches, the quality and quantity of which may be substantially different from process to process. In order to succeed, one must demonstrate satisfactory knowledge with the process itself, and skill with the methodology used. Kabbaj et al. [3] stated they were working on a list of predefined "deviation types". Can Kabbaj et al. really produce a comprehensive list of all possible types of non-conformance? It seems likely that there are certain rare types that may be left unconsidered. If they cannot, the consequences are unlikely to be too severe. Their approach will still be useful in detecting both the common and uncommon types they list in their approach.

In the GARDEN approach, any occurring non-conformance instance triggered by its associated rule set will be recorded and stored in the form of the concept of "non-conformance", illustrated in Figure 39. If some form of non-conformance happens to occur that has not been catered for by the defined rule sets, the beauty of the rule set methodology is that a new rule set can easily be defined and implemented to cater for the new non-conformance type. Of course, there exists the possible danger that such undetected non-conformance types could go unnoticed completely, and therefore remain unaccounted for by the GARDEN implementation.

A similar conundrum regarding the quality of the reference model and unanticipated non-conformance types faces the approaches by Huo et al. [1] and Cîmpan and Oquendo [2]. The discovery approach by Huo et al. [1] requires a process model to be derived from previous enactment data. Cîmpan and Oquendo's approach [2] requires an "ideal" model to be constructed. In either case, there is a question mark on the practical quality of each reference model from process to process. In Cîmpan and Oquendo's [2] case, this is due to the subjective nature of what constitutes an "ideal" enactment. In the case of Huo et al. [1], the quality of the reference model is directly related to the quality of the previous enactments from which it was discovered. The quality of previous process enactment examples is also a subjective judgement. Catering for unexpected non-conformance types is always going to be more challenging to deal with than expected types. However, the way non-conformance has been conceptually scoped using the GARDEN approach makes dealing with unanticipated types of non-conformance much easier when they are discovered.

8.2.2 Minimising interference with the existing process

GARDEN is designed to be an observatory approach, meaning it functions by using the process enactment data and does not exert any control over the process itself. Therefore, interference with the existing process is automatically minimised to some extent, compared to support-based approaches as shown in chapter 2, section 2.5. However as shown in chapter 3, interference is not only concerned with how the approach affects the process enactment. It is also concerned with how the resources required to implement the approach can be redirected from the specification, performance and maintenance of the process to the implementation of the non-conformance detection mechanism. This occurs because the people and resources available to implement the approach would usually be the same people who would otherwise use that time working on the existing process. Their time is diverted from supporting the process to implementing and maintaining a non-conformance detection solution. GARDEN is specifically designed to minimise this tax on process resources, as the solution can be applied with minimal effort and maintenance is only required on the affected process areas when the process evolves.

According to the literature, the more successful of the existing approaches are also based on observing the process rather than controlling or supporting it. The exception is Cugola et al. [6], which was shown in chapter 2 to not necessarily be aimed at simply non-conformance detection but more to support a process when non-conformance occurs. Discovery approaches such as Huo et al. [1] and Cook and Wolf [5] have also been known to require a vast amount of setup effort, which although does not interfere with the enactment of the process directly, still redirects resources from the process itself to the implementation of the approach.

The approach presented in this thesis combats indirect interference by remaining flexible in the way it is implemented. The implementation of GARDEN will still likely need to be completed by a person experienced with the existing process, which inevitably causes indirect interference on some level. The key difference is that the implementation can be done on any level desired and is appropriate for the target process. If only a certain part of the process needs to be addressed, then rule sets can be specified to address only that part. This eliminates any indirect interference with any other part of the process to which GARDEN is not being applied.

8.2.3 Minimising cost and effort of implementation and maintenance

The biggest problem facing process discovery approaches like Huo et al. [1] and Cook and Wolf [5] is that it requires an excessive amount of enactment data to build the reference model before the detection of non-conformance can even begin. As discussed in chapter 3, the effort of implementing the approach should never offset the benefits the approach will bring, so minimising effort and cost is a major concern. Practically speaking, the greater the cost and effort required in implementing the approach, the less likely anyone will be to invest in its implementation.

As the case studies have shown, the GARDEN approach deals with keeping cost and effort low through a number of innovative ways:

1. GARDEN can be applied to specific parts of the process only. If only certain aspects of the process are a concern, then GARDEN does not interfere with the parts of the process that do not require addressing, meaning no wasted implementation effort on areas that do not need it.

2. Since GARDEN associates itself with specific parts of the process, when the process evolves and changes, only the affected parts need to be addressed. The aspects of GARDEN attached to unchanged parts of the process can remain as is. This flexibility saves time and effort in maintenance.

3. Given access to a GARDEN implementation tool, such as the tool described in chapter 6, it is a simple matter for a domain expert to apply GARDEN to a process. Its application, maintenance and removal is easy and fast.

4. Unlike other approaches that use a reference model [1] [2] [5], there is no need with GARDEN to spend energy on mining enactment data from past process instances, or to carefully simulate flawless enactments for reference. Especially with approaches that use process discovery, there is no requirement

to collect and mine the data from new enactments again once the process changes and the reference model needs to be rebuilt.

In GARDEN, the solution to minimising implementation and maintenance effort and cost was tested in the first case study presented in chapter 6. Using the interface provided for the process simulation tool, a domain expert is given the ability to implement the GARDEN approach by defining different rule sets for different parts of the process. During the internal testing phase of this case study, a sample "domain expert" was timed as they defined and applied different rule sets to sample processes. For the purposes of this testing, the constraints were pre-defined informally so the domain expert simply had to use the framework to translate these constraints into GARDEN rule sets and apply them to the process. Therefore, any time spent by the domain expert on considering these constraints or analysing the process is not included in these figures. The details of this case study are presented in further detail in chapter 6.

According to the data, the amount of time spent applying GARDEN to each part of the process is somewhat variable depending on the nature of that part of the process and the constraint being applied. Applying time duration constraints on an activity is quick for example, but more complex rule sets with numerous rules within them can be more time consuming. Nevertheless, the relatively short amount of time required to implement GARDEN in this way is certainly well worth the benefits the approach can bring to the process, especially if the process is repetitive in nature. The chart in Figure 40 depicts the distribution of time spent by a process domain expert on using the simulation tool interface to define rule sets for some example processes.

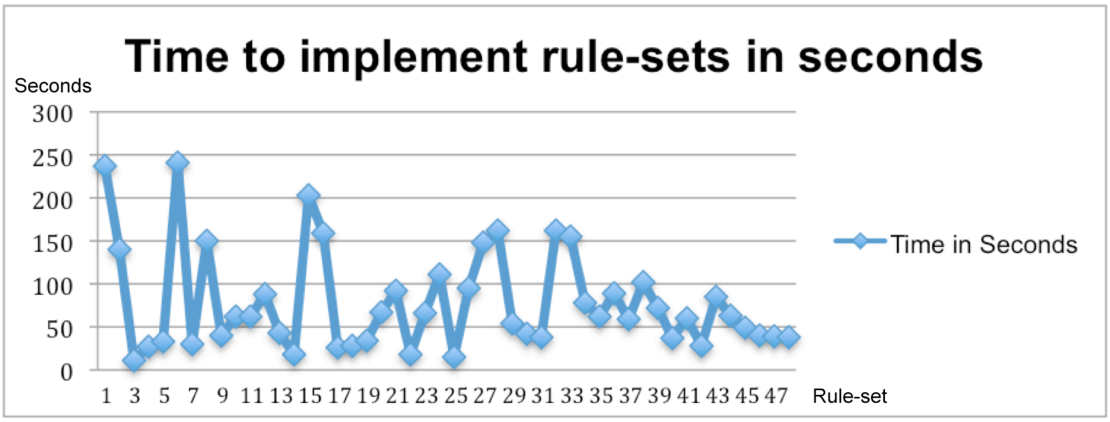


Figure 40: Time duration to implement a rule set using the simulation tool

As can be seen in the data in Figure 40, there is a high degree of fluctuation in the time it takes to define and implement different rule-sets to the sample process. The reason for this fluctuation is the way in which the simulation tool presents functionality to the user. For example, placing a maximum time duration constraint on an activity is as simple as selecting the activity and entering the number of seconds the activity is allowed to run for. Rule-set applications like this are quick and easy when GARDEN is being applied using the simulation tool's rule-set definition engine. However, it is also possible to define customised rules applicable to the data itself and then group them into rule-sets, which is provided for using the tool. This is more time-consuming and is responsible for the instances in the data where it took comparatively longer to define and apply the rule-set.

The key benefit with this type of approach is the savings on maintenance effort. When a process evolves and the non-conformance detection approach needs to be analysed, only the parts of GARDEN linked with the sections of the process that were affected by the change need to be addressed. If we surmise that a process evolution causes 20% of the process to change, then only rule sets applied to that 20% of the process require attention, plus any rule sets applied to the process in general. Any approach that utilises a reference model that needs to be rebuilt will require a 100% rebuild of the reference model. Since the other approaches do not indicate the amount of effort required to implement and maintain their approaches, which is largely dependent on the process anyway, it is difficult to otherwise conduct a meaningful comparison of the data. However, showing that GARDEN can make a real and measurable time and effort saving difference during the maintenance phase of the process highlights the benefit of this approach.

As with all non-conformance detection approaches, their effectiveness depend on the availability of process enactment data. GARDEN is not immune from this dependency, and the collection of process enactment data could ultimately be a significant task in terms of time and effort. This affects the net gain GARDEN has the potential to bring to a process in terms of cost versus benefit. Additionally, since GARDEN relies upon the application of rule-sets to the process enactment data, there remains the possibility that the data may require "massaging" or some form of transformation in order for the rule-sets to be adequately applied. If this scenario eventuates it will also increase the cost involved in applying the GARDEN solution and requires careful consideration.

In regard to the effort required to set up the approach once the enactment data has been confirmed as available and sufficient, GARDEN needs relatively very little effort. This is especially true when compared to other approaches such as Cîmpan and Oquendo [2] which require the entire process to be completely modelled as an ideal enactment before it can be applied. In addition to the set up described in this section, the same analysis of the existing process will still need to be made as with any other approach, along with forethought given to what constraints and boundaries would be appropriate for GARDEN to apply to that specific process.

8.2.4 Keeping the approach flexible

As discussed in chapter 3, a flexible and maintainable approach to non-conformance detection makes it a much more attractive integration choice from a practical perspective. Processes change over time, and as they change the non-conformance detection approach needs to adapt to these changes. It seems that previous approaches in this area have not placed this consideration high on the list of priorities.

The data mining approaches published in Huo et al. [1] and Cook and Wolf [5] will struggle the most, because according to the literature, the reference model used is derived from enactment data. This implies that the changed process will need to be enacted a certain number of times in order to generate enough enactment data to build the reference model from which to compare future enactments. In fact this is a key concern cited in the literature from Cook and Wolf [5].

The "ideal enactment" model mentioned in Cîmpan and Oquendo [2] will also need to be updated every time there is a change in the process specification. When the process specification is modified, we would need to assume that what constitutes an "ideal" enactment would have changed from what it was previously. Re-modelling the reference model may be cumbersome every time the process evolves, but it is likely to be less of an annoyance than re-discovering the reference model from enactment data as in Huo et al. [1] and Cook and Wolf [5].

This is where GARDEN provides a practical benefit. GARDEN is applicable in sections which reference different parts of the process. When the process specification changes, only those sections of GARDEN referencing the changed part of the process will need to be addressed. Any aspect of GARDEN that references parts of the process that were not changed can be left alone, as was shown previously in section 8.2.3. This is a considerable time saver for the domain expert or person handling the non-conformance detection approach maintenance.

One limitation of GARDEN however, is that every time the process does change, the aspects of GARDEN associated with the general process emergent properties will need to be reconsidered. This is because these constraints are usually based on assumptions about the process as a whole, which may change when its activities change. These properties may include constraints such as total process duration time limits, limits on the number of activities that can be enacted, or the time of day the process may be commenced or completed. If one of the activities in the process is modified or removed, this can change the meaningfulness of any constraints applied to the process as a whole.

8.2.5 Keeping the approach generically applicable

Keeping a non-conformance detection approach generic usually means accepting a trade-off on the benefits a specific approach might bring to the situation at hand. Generic approaches are applicable to a wider range of processes, but may lack the focus of a solution designed for a specific process. Specific approaches may better target the individual requirements unique to a single process, but have limited applicability and practicality since they are not useful outside that specific process. A specific approach requires the total amount of effort and cost being contributed to develop a non-conformance approach for possibly only a single process specification.

Ideally, a successful non-conformance detection approach would be applicable to as wide a range of processes as possible, but possess the flexibility to be tailored to fit the unique requirements of each individual process to which it is applied. The GARDEN approach is designed with this goal in mind.

The other process non-conformance detection solutions approach this issue in different ways. A viable solution needs to detect non-conformance successfully, but it also needs to be applicable to a wide range of processes. Generic applicability is cited as a key concern in the approach by Rozinat and van der Aalst [9]. Approaches rooted in process model discovery through mining the enactment data [1] [5] employ a generic methodology but each application will be specific since the reference model is derived from a processes enactment data. This is similar with the "ideal enactment" simulation presented by Cîmpan and Oquendo [2]. Kabbaj et al. [3] have chosen to anticipate all possible non-conformance instances for which they want to check advance, which also employs a specific application of a generic methodology.

In GARDEN, the objective, as in approaches such as Huo et al. [1] and Cîmpan and Oquendo [2], was to develop a generic technique or methodology. The rule set approach fulfils this objective. The actual rule sets specified is what provides the approach's specific non-conformance detection power but the real value regarding this issue is the generic methodology that has been presented in this thesis, not the individual tools and rule sets built for one process in one situation. This was confirmed using the different process types tested during the different implementation test cases.

8.2.6 Maintaining an observational methodology in the approach

The use of an observation methodology rather than a support methodology is largely related to the minimisation of interference in the process, as covered in section 8.2.2. A non-conformance detection approach that is observational will have minimal interference with the process enactment because it has no interest in control.

This leads to another key reason to use an observational approach as stipulated in chapter 3, section 3.1.6, which was to avoid missing an opportunity for process improvement. Support methodologies such as Cugola et al. [6] can prevent non-conformance occurring, and therefore also prevent the consequences the non-conformance instance may have caused. However, by preventing non-conformance an opportunity to gain valuable knowledge about the process is missed. Perhaps there was a good reason the process specification was not followed, resulting in a better outcome. Or perhaps the process specification is flawed in some places and non-conformance occurring there helps highlight these flaws so we know where to improve it. This knowledge can ultimately be used to improve the process overall, without rigidly constricting its flow.

There is an opportunity for observation-based approaches like GARDEN to contribute to the improvement of the process by utilising the knowledge gained about non-conformance quickly. GARDEN yields a dual benefit by operating in this way. First GARDEN ensures that the valuable information gained by remaining observational and not interfering with the process is retained. Second, the possible consequences resulting from non-conformance that was not prevented as it would have been in a support-based system can be dealt with very quickly, as GARDEN minimises the time between occurrence and detection. This is more of an issue for observation-based approaches that rely on reference models built after process enactment [1] [2] [5], since it affects their time-to-detection.

Since GARDEN is designed to work by evaluating process enactment data without exerting any control over the process itself, it succeeds in being classified as an observation-based approach. In addition, the timeliness of GARDEN helps protect it from the risks of delayed detection that affects other observation-based approaches, as is shown in section 8.2.7, and the future considerations support built into GARDEN help provide useful information for process improvement, as shown in section 8.2.9.

8.2.7 Minimising time to detect non-conformance

The elapsed time between process non-conformance occurrence and its later detection is an important issue. The more time required to detect non-conformance, the longer we have to wait to realize process improvement and the longer the amount of time the consequences of non-conformance has to manifest itself in potentially harmful ways.

The primary way GARDEN achieves the goal of minimising the required time before non-conformance could be detected was through enabling the rule sets to be applied to the enactment data as soon as the data becomes available. This worked well as shown in the implementation studies, however as these studies (from chapters 6 and 7) also show, it is always dependent on the scenario and the way GARDEN has been set up.

As shown in chapter 5, there is a lot of flexibility afforded to the team responsible for implementing the GARDEN approach to a process. Depending on the process, how the enactment data is gathered and recorded, along with how the rule sets have been set up to access the data, will have a significant effect on the amount of time it will take to detect the non-conformance.

The best case scenario is that the enactment data can be formulated into its desired structure as it is gathered and the rule sets can access it immediately, even if the process has not concluded yet – i.e. the rule sets can access the enactment data as it is recorded. As shown in chapter 6, this is certainly possible using GARDEN, so on-the-fly detection is afforded if the process and the environment also support it.

In terms of time-to-detection, the worst-case scenario for GARDEN would be if the enactment data required a complete reworking into a separate structure before GARDEN could begin to evaluate it. Some aspects of the enactment data evaluation may also require a completed data set before it can occur. An example of this might be checking to see if the total process time duration had exceeded its limit, as the process must be concluded before this can be determined. These factors can increase the time to detection.

So, although the timeliness of non-conformance detection is very much dependent on the particular case in question, it has been shown that GARDEN at least provides the opportunity to detect non-conformance quickly. This is an important resolution as acknowledged in Huo et al. [1] and Cook and Wolf [5], whose process discovery approaches can suffer from a wide time gap due to the need to mine enough enactment data in order to derive the process reference model.

8.2.8 Acquiring process enactment data

The acquisition of process enactment data is more of an indirectly related concern to process non-conformance detection rather than a key issue. However, it is still a concern because non-conformance detection approaches require process enactment data to determine whether the process specification has been followed or not. Other approaches have varying ways of dealing with this, such as Rozinat and van der Aalst stating the assumption that all process enactment data logs are structured in a specific way that suits their approach [9].

Acquisition of process enactment data is not an issue covered in any of the other process non-conformance detection approaches including Cîmpan and Oquendo [2], Kabbaj et al. [3] and Cugola et al. (1996) [4], however Cugola et al. (1995) [6] have an implied relationship with the monitoring and recording of enactment data since the SENTINEL approach is based upon process support and monitoring. Rozinat and van der Aalst [9] also mention this issue and state their assumptions in order to deal with it.

Nevertheless, the issue of acquiring process enactment data in such a way that GARDEN could be implemented and tested properly was explored. The case studies presented in this thesis handle this task in two different ways. In the simulation tool case study, a common data structure that plugs into the process simulation environment was developed, and the enactment data is written straight to the data structure. In the online implementation, the action of each users interaction with the system was logged, and then the GARDEN methodology was applied to these logs to detect non-conformance.

Similar to Rozinat and van der Aalsts necessary assumption [9], it is acknowledged that any methods employed to acquire the process enactment data are inconsequential. The important thing is that GARDEN is capable of using and evaluating the process enactment data so that non-conformance instances may be detected if they have occurred.

Other approaches in this area make similar concessions regarding the acquisition of process enactment data. The process discovery approaches [1] [5] assume that the enactment data already exist and build a reference model from whatever is available. It is also assumed here that the available enactment data to build the reference model should be on the same level as the enactment data from a process the model is to be applied to, which was an issue that Huo et al. [1] addressed to ensure that the levels of enactment data and reference model data remained the same to detect non-conformance.

The reality is that processes are different and will return enactment data in different formats and at different points during the process. The monitoring and recording of such data will be vastly different, depending on each unique situation and a good non-conformance detection approach will want to be as applicable to as many different processes as possible. This is why GARDEN can offer a key benefit that other approaches find more difficult to deal with. The versatility of the GARDEN approach, in this instance, is its inherent ability to be applied to process enactment data regardless of its form. As long as the process enactment data is available, structured and consistent, GARDEN is applicable. This is explained in the application guideline in greater detail in chapter 5, section 5.3.

8.2.9 Future considerations post non-conformance detection

One of the key objectives of the GARDEN approach is to provide some consideration for what happens after non-conformance is detected. If non-conformance occurs, it may have consequences. Decisions may need to be made to address these consequences, and supporting these decisions by reporting high quality information about the non-conformance instances is an achievement of the GARDEN methodology. GARDEN achieves this through a mechanism to determine relative non-conformance severity, and also remedial action guidance is given by logging past remedial actions and their judged effectiveness.

These considerations have not yet been acknowledged in the literature, so introducing them as part of the GARDEN approach is a positive step toward advancing this research area. These considerations, however, have only just begun to be explored and have not been researched and tested comprehensively as yet. The initial work conducted regarding severity and recommendations was published previously in Thompson and Torabi [7].

8.2.9.1 Non-Conformance Severity

The first future consideration GARDEN addresses is determining the relative severity of an instance of non-conformance. Other approaches have made attempts to provide a severity value, such as the severity rules described in Kabbaj et al. [3] or the toleration thresholds for allowing the process to continue in the face of deviations described in Cugola et al. [6]. However, GARDEN looks at the concept of severity in a different light and instead, GARDEN provides a framework for calculating a severity value that represents the relative importance of the non-conformance instance in relation to the process itself. This is something more meaningful than the literal difference between actual and expected value and is indicative of how this specific instance of non-conformance could affect the outcome of the process, or how it could affect the organisation in which the process runs, which differentiates GARDEN from approaches like Rozinat and van der Aalst [9] and Cîmpan and Oquendo [2].

To illustrate this, consider an example of how other approaches consider severity as a function of literal difference between actual and expected values depicted in Figure 41:

Non-Conformance Detection Constraints	
Activity:	Calculate deposit
Minimum Time:	10 seconds
Maximum Time:	**120 seconds**

Process Enactment	
Activity:	Calculate deposit
Start Time:	13:12:45
End Time:	13:15:33
Duration:	**168 seconds**

Severity:	**48 seconds**

Figure 41: Current Severity Evaluation

To show how GARDEN improves on this depiction of severity, consider the same example with a subtle difference. The way GARDEN would evaluate the same occurrence of non-conformance is depicted in Figure 42.

Non-Conformance Detection Constraints	
Activity:	Calculate deposit
Minimum Time:	10 seconds
Maximum Time:	120 seconds
Importance:	**Minor**

Process Enactment	
Activity:	Calculate deposit
Start Time:	13:12:45
End Time:	13:15:33
Duration:	**168 seconds**

Severity:	**Minor**

Figure 42: Improved Severity Evaluation

Instead of an ambiguous figure that is of no practical use to the domain expert, this alternate representation informs them that this non-conformance occurrence is *minor*, which is much more meaningful information in the scope of the process and its operating environment.

The key significance in the conceptual difference between some of the other approaches idea of severity and GARDEN's idea of severity is that in other approaches, the severity value can only indicate how far an actual value has strayed from what was expected. No other approach takes into account how important a specific non-conformance instance is or how critical the part of the process is where the non-conformance occurred. The GARDEN framework facilitates the structuring of this information so that a more reliable and useful severity value can be derived. This was also discussed in several parts of chapter 4, section 3.

Although this facet of the research has only proceeded as far as what is published in Thompson and Torabi [7] along with some other unpublished work, the foundation has been laid to expand on this idea and provide a useful tool for ascertaining the real significance of a detected instance of non-conformance.

8.2.9.2 Remedial Action Recommendation

Another unexplored area of process non-conformance detection is forging a way to automatically suggest what should best be done to either limit exposure to the consequences of non-conformance or to improve the process itself. The objective in this thesis is to contribute a way of dealing with this issue, however it will likely remain the focus of much research and study before it is completely resolved. In this vein, a simple framework was presented and described in sections 4.3.5.4 and 4.3.5.5 of chapter 4 as a basis for future research in supporting the decision-making process post non-conformance detection. This feature of GARDEN is based loosely on the tried and tested recommender system model, the basis of which was implemented and tested as shown in Thompson and Torabi [7]. The form this took was in the inclusion of two attributes in the non-conformance detection log. This was illustrated in chapter 4, Figure 10. These attributes are intended as a mechanism for the process domain expert to log the remedial actions they take after non-conformance detection, and then after evaluating these actions, to then log an associated "success" rating of the action.

In the same light as other recommender systems, it was surmised that a log of previous non-conformance occurrences and their outcomes could provide useful information for similar future occurrences. After all, gathering a history of as much data as possible concerning the process should facilitate the gaining of knowledge and the spotting of any patterns if they exist. The key goal is to make sure the non-conformance logs were recorded and structured properly.

The basic format of the history log was to provide a simple record of the non-conformance instances along with remedial actions taken (if any) and their perceived effectiveness. This information can then be used when the same or similar non-conformance instances occur in the process in the future. If a particular action was successful in dealing with a form of non-conformance, then it suggests this action may be worth repeating if the non-conformance occurs again in the future. Similarly, if an action was judged as a bad idea, then future repetitions of such an action can be avoided.

As this is the only direct attempt at facilitating this problem in the literature, there is certainly room for expansion and improvement. According to the testing of this feature of GARDEN, it has been shown that this method can provide a reasonably useful indication of what a process domain expert should do when certain non-conformance instances are detected. However, in its present form, this feature is only useful when non-conformance instances are recurring, as detection of new instances will not have a "history" and therefore decision support cannot be advised using this aspect of the GARDEN solution.

8.3 Evaluation of further outcomes from GARDEN

The purpose of this section is to provide further evaluation outcomes of the GARDEN solution that were not explicitly detailed previously as objectives for this research. Here, various pros and cons that were discovered while developing and testing GARDEN are discussed. Some of these findings were surprising and some were not, but these findings give an indication as to the usefulness of the GARDEN methodology. It also highlights some areas where the GARDEN approach could be improved.

8.3.1 Recommender System Style Drawbacks

Part of the value of the GARDEN solution is to provide a hybrid "recommender" style advice portal intended to support post non-conformance decision-making, presented in sections 4.3.5.4 and 4.3.5.5 of chapter 4. The feature is loosely based on recommender systems, where the goal is to determine the best course of action for a given non-conformance instance, based on the success of past remedial actions. Similar to recommender systems, GARDEN relies on the existence and quality of past process enactment data in order to suggest an effective remedial action. To prevent information overload, an analysis is performed on the available history of the same type of non-conformance and the actions that were taken, hiding the irrelevant or poorer past actions and returning the more successful and useful [8].

Since an appropriately-sized log of previous enactments is required in order for the recommender aspect of GARDEN to advise effectively, it means that dependence upon a log of data that takes time to accumulate may make this aspect practically unviable in certain situations. This is simply because the best remedial action to non-conformance may be to change the process specification. If the process specification is modified, then all previous enactments based on it are now rendered obsolete and out-of-date.

Until an appropriate amount of process enactment data can be re-accumulated against the new process specification, the process domain expert will have to rely on subjective judgement in conjunction with the old process enactment data, which he or she will be aware is now inconsistent with the new process specification. This resolution is workable for modified aspects of the process, but brand new non-conformance types will not have any history associated with them, so the domain expert will have no help from a feature like this in such a case.

This section of the research is ongoing, however, and has not yet been exhaustively explored. Although this line of research seems to have some merit, the main drawback of any methodology requiring a large backlog of previous enactment data (problems acknowledged by Huo et al. [1]) is that a number of enactments are required in order to obtain this data. Once it is used to improve the process, the process has now changed and so the data may be useless for further improvements.

This is not to say it is not worth the effort of collecting data so that an improvement may be made only for the collected data to be discarded; it is simply a drawback of this type of methodology.

8.3.2 Applying a complete and comprehensive solution

An issue inherent to approaches that employ rule specification as a means to detect non-conformance is that not all the ways in which non-conformance can occur will be known before the approach is implemented. There will be a set of known possible non-conformance instances, some obvious and some not so obvious that can be checked, but there will also be other unknown possibilities that should also be picked up by a non-conformance detection approach. This is a main weakness in other rule-based approaches such as Kabbaj et al. [3] and Zazworka et al. [10], where non-conformance types are specifically defined. The problem with defining a list of rules to check for specific types of non-conformance is that the solution may miss non-conformance types that were not anticipated. Unexpected non-conformance instances may then occur without being noticed, potentially causing undesirable consequences.

Approaches of this nature are faced with two burdens to resolve:

1) Identifying as many likely non-conformance types as possible and tailoring the non-conformance detection solution to detect them;

2) Ensuring that the implementation of the non-conformance detection solution is comprehensive enough to also detect any non-conformance types that may not be expected or anticipated.

This of course is an issue that also affects all approaches for process non-conformance detection. The challenge is to conceive a solution capable of detecting non-conformance types that are not known to exist before they occur. The other approaches directly related to GARDEN do not mention explicitly dealing with this issue. The effect this issue may therefore have on these other approaches is surmised as follows:

1. Kabbaj et al. [3] employ a list of non-conformance types they have pre-determined, so any non-conformance occurring that does not match an entry in this list will not be detected. Therefore, as stated previously this issue is not addressed by Kabbaj et al.

2. Cîmpan and Oquendo [2] require an ideal process to be modelled, but this does not take into account the different ways a process might be legally enacted and also the different possibilities of non-conformance. In other words, the same process may be enacted perfectly legally in two completely different ways, and this appears difficult to model using the approach provided by Cîmpan and Oquendo. Therefore, it follows that it is difficult to model some processes comprehensively using this approach.

3. Huo et al. [1] derive their comparison model from already enacted processes, but this forces a dependency on the previous enactments to be adequately comprehensive so as not to miss anything obscure or legal but rare. This begs the question, if some form of the process is legal in specification but rare in occurrence, then how many enactments are needed in order to derive a

comprehensive model to compare? This was a self-criticism Huo et al. have stated about their approach. Also, if one of these rare process atoms does occur, then how will we know if it is non-conformance or simply just a rare yet still legal part of the process?

4. Zazworka et al. [10] explicitly admit to creating a list of "partial non-conformities". In other words, they are not concerned with the issue of monitoring for the detection of all non-conformance types and instead offer a solution to only check for the select few they care about.

This issue can be argued as trivial against the approach by Huo et al. [1] because if the occurrence in question is so rare that it is not picked up by any of the enactments used to build the comparison model then it might as well be considered non-conformance. Addressing the occurrence will not waste much time because it is so rare and if it really is a non-conformance instance then it needs to be addressed anyway.

The GARDEN approach addresses this issue much more explicitly than the aforementioned approaches because it is designed to provide a way to check for specific non-conformance types, as well as to adequately check the process enactment for unanticipated non-conformance types. In this way, GARDEN helps to explicitly address an issue that is common to all non-conformance detection methodologies that has not been explicitly covered in the literature.

8.3.3 Source of knowledge

The opportunity for learning more about the process in question, its weaknesses, the types of non-conformance that are likely to occur and how to handle them is all related to covering the process as comprehensively as possible as discussed in section 8.3.2. The better the coverage of the process GARDEN can provide, the more data it will yield to provide valuable information about the process. This can also be information that is not always directly related to detected non-conformance instances.

This discovery was made in the online implementation of GARDEN, covered in chapter 7. Despite some reasonably rigorous development and testing of what was really a relatively simple online tool for compiling survey data, some interesting flaws with the tool were brought to light quite quickly after GARDEN was applied to it.

Essentially, the analysis of non-conformance data provided by GARDEN served to uncover a weakness in the process that would have gone unnoticed had GARDEN not been applied, thus allowing an improvement in the process specification that was also shown in data given in chapter 7, section 7.7.3.

To assist in explaining how GARDEN can help produce greater knowledge about a process and how it was discovered using the online implementation, refer to the form used in the online implementation in chapter 7, section 7.5, Figure 36. Also note that the CAPTCHA section on the form is the red code that the participant is requested to copy. Although this is shown as being four characters in length, it was actually six characters long in the initial stage of the online GARDEN implementation.

According to the data from this GARDEN implementation that was given in chapter 7, one telling facet of this implementation is that although the rate of CAPTCHA fails in the first phase was over 28% (81 fails from 286 attempts), not a single user mentioned it on the feedback form, despite the fact that there was a space for any additional comments the participant may want to give as feedback, noting that the form data was still submitted even when the CAPTCHA code was entered incorrectly.

The result from this analysis is that even though there was a problem with the process that was so severe that failures were encountered more than 28% of the time, not one participant decided to give feedback on this encounter, and in fact, this problem would never have been discovered if not for the application of the GARDEN methodology. Furthermore, the application of GARDEN in this case study had no rule sets specified that were specifically looking for this kind of flaw in the process.

This shows how GARDEN managed to uncover a serious usability issue through providing non-conformance data, that when analysed showed the unreported problem. It also shows that through other cases there is an opportunity to learn a lot about a process to which GARDEN has been applied. No rule sets were specified to detect a problem with the CAPTCHA form, but GARDEN uncovered this problem because rule sets were specified to ensure data input consistency regarding the CAPTCHA. This comprehensive approach allowed this flaw in the process to be discovered, showing the kind of beneficial knowledge that can be gained not just from the application of a non-conformance detection methodology to a process, but specifically a comprehensive rule based methodology such as GARDEN. Whether knowledge acquisition to this extent in other non-conformance detection methodologies is possible is open to debate, however this benefit has not been emphasised in any of the literature concerning these competing approaches.

8.3.4 Ease of implementation

This issue relates to the cost in terms of time, effort and pecuniary in implementing the GARDEN approach for a process. In terms of the other non-conformance detection approaches discussed in this thesis, this is usually rather predictable and measurable after an initial analysis of the target process has been performed.

For example, the approach presented by Huo et al. [1] involves the mining of enactment data in order to construct a reference model. The cost of performing this task can be gauged by scrutinising the process and the available enactment data and making an estimate of how many enactments with what sort of data would be necessary in order to discover the reference model required. The disparity in relative implementation costs here would be the availability and attainability of the process enactment data.

Similarly, Cîmpan and Oquendo [2] require that the reference model be conceived as a "flawless execution", so the cost involved here is estimated by how much work is involved in producing what would be considered a flawless execution of the process to compare enactments against.

With GARDEN however, the approach is much more susceptible to differing implementation costs, depending on the process and the environment in which is it enacted. First, the same as in Cîmpan and Oquendo [2], GARDEN will rely on the availability and format in which the enactment data comes. Cîmpan and Oquendo [2] assume this data will be at a level where discovering the reference model can be derived from it. When implementing GARDEN, a decision will need to be made as to whether to evaluate the data in the form it is supplied or to apply a data transformation process to it to make this evaluation easier.

Other concerns will also become relevant as the environment the process runs in becomes clearer. Factors such as the complexity of the specific implementation, the requirement for comprehensiveness, the number of constraints and boundary values required to be set, the availability and format of enactment data, the environment the process runs in and the environment in which GARDEN is implemented will all have varying degrees of influence on the required implementation cost.

Therefore, GARDEN can in some cases be exceptionally easy to implement as was found in the online implementation where data was easy to accumulate and use, or it could be more cumbersome, as was found in the alternate case study and which is implied for other processes running in increasingly complicated environments.

In turn, this will also affect change management. The more difficult GARDEN is to implement, the more difficult it may be to change and maintain once implemented. This issue is not overly large however, because an analysis can be made before implementation to decide whether the cost of implementing GARDEN is advantageous compared to the potential improvements it could yield.

8.4 Conclusion

The evaluation of the GARDEN non-conformance detection methodology has shown that it was successful in both detecting non-conformance in processes, as tested with the case studies and also achieving the objectives set out when attempting to detect non-conformance.

This chapter has shown how GARDEN achieved the objectives first presented in chapter 3 as targets for what a non-conformance detection approach should aspire to. These objectives were reiterated and the way GARDEN achieved them was evidenced through the case studies also presented in this thesis and the data that was derived from these case studies.

In addition to this evaluation of the stated objectives, GARDEN was further evaluated through some additional benefits and issues that were discovered while developing, testing and evaluating the approach. These issues and benefits were covered in detail in this chapter is section 8.3.

References

[1] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[2] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[3] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[4] G. Cugola, E. Di Nitto, A. Fuggetta, C. Ghezzi, A framework for formalizing inconsistencies and deviations in human-centered systems, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 5 Issue 3, ACM Press, July 1996.

[5] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM)**,** Volume 7 Issue 3, July 1998.

[6] G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[7] S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

[8] Y.Z. Wei, L. Moreau, N.R. Jennings, A market-based approach to recommender systems, ACM Transactions on Information Systems (TOIS), Volume 23 Issue 3, ACM Press, July 2005.

[9] A. Rozinat, W.M.P. van der Aalst, Conformance Checking of Processes Based on Monitoring Real Behaviour, Journal of Information Systems, Volume 33, Issue 1, Elsevier Science Ltd., Oxford, UK, March 2008.

[10] N. Zazworka, V.R. Basili, F. Shull, Tool Supported Detection and Judgment of Nonconformance in Process Execution, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, October 2009.

Chapter 9 – Conclusion

9.1 Review of the Problem

According to the review of existing literature covered in chapter 2, there were several key issues identified as important problems to solve going forward in the research area of process improvement through non-conformance detection. The solution presented in this thesis and the case studies used to support it were designed to address the following concerns, alongside the successful and practical detection of non-conformance instances between a process specification and its enactments:

- to clarify what is meant when referring to particular terms and concepts in the context of this research area. The existing literature uses these terms and concepts with inconsistent meanings, so acknowledging this inconsistency and very clearly defining what is meant when referring to certain terms in this research was a high priority.

- to minimise the interference with the actual process that is caused by the task of integrating and maintaining the non-conformance detection approach. This relates mainly to keeping the number of resources redirected from supporting the process to integrating the approach as low as possible.

- to keep the amount of effort and the cost of implementing the approach low enough to ensure that the benefits of having the approach far outweigh the cost of integrating it.

- to ensure that the approach is capable of evolving with a changing process and that maintenance of the approach is as simple and straightforward as possible.

- to keep the design of the solution generic enough to maximise the number of different process types running in different environments that have the potential to utilise it.

- to focus solely on non-conformance detection rather than prevention. This allows the possibility of benefiting from non-conformance instances that actually yield an improvement to the process.

- to keep future decision-making relating to non-conformance in mind with the design of the approach. This includes concerns such as determining how “severe” non-conformance can be judged and what future implications this may have. It also includes the tracking of what actions were taken following non-conformance detection and the relative success or failure of those actions for future consideration.

- to minimise the amount of time that elapses between the occurrence of non-conformance in the process enactment and the time that it is detected.

- to present an approach capable of dealing with the different ways process enactment data may be provided for use in detection, and ensuring that the approach can still run effectively given these differences.

9.2 Summary of work presented

The body of work presented in this thesis can be separated into three main categories: the problem, the solution, and the evaluation.

Chapters two and three of this thesis begin the core body of work aside from the introduction and contain a survey of the literature related to the domain and detail the remaining issues and holes that are left to be explored and resolved. Existing approaches to process non-conformance detection are analysed and summarised with their strengths and weaknesses and how they address the remaining challenges. This was followed by a chapter dedicated to the problem statement, where the issues were identified, described and justified as the issues that the proposed approach was going to resolve.

The following chapters four and five provided a detailed description of the solution, an explanation as to how it works, why it was designed this way, and the methodology for how it can be successfully implemented into an existing process to detect non-conformance. Some examples were used to illustrate the solution. This section of the thesis is the core offering of GARDEN and shows how the issues mentioned previously will be resolved.

The final main section of the thesis involved the testing and evaluation of the GARDEN approach via an implementation in a process simulation environment, a real world case study, data capturing from both initiatives and evaluating the results in chapters 6, 7, and 8 respectively. The case studies and evaluation showed how a process could undergo analysis through a GARDEN implementation and have the issues associated with non-conformance highlighted. Once the issues are addressed, these case studies and evaluation show how such a process can be improved overall. Chapters 6 and 7 present an implementation of the solution along with the resultant data and outcomes and Chapter 8 provides an evaluation of the data and an analysis of what went wrong and what went right with the implementations, and also the positive and negative aspects of the solution.

The purpose of this final chapter is to tie together these main sections and conclude the thesis.

9.3 Recap of the Solution

The work presented in this thesis described and justified the methodology named "GARDEN", a solution to detect process non-conformance. This solution was designed to address the issues currently facing this research domain covered in chapter 3. In short, GARDEN was designed to detect non-conformance between a process specification and its enactments in the most practical way possible, where practicality is stressed from its first integration through to its maintenance and evolution with the process.

The GARDEN journey begins with an analysis of the process specification, with the intention of discerning exactly what the process enactments are supposed to do. From this analysis, a two-pronged approach is employed to adequately detect all types of non-conformance that could occur during process enactment. GARDEN is applied to different sections of the process for the purposes of evaluating the enactment data from that part of the process. The approach is kept light and flexible in order to reduce implementation effort and increase its practical appeal, as opposed to other potentially heavy approaches like Huo et al. [4], Cook and Wolf [5], and Cîmpan and Oquendo [6].

The two-pronged approach refers to the tactic of first applying the GARDEN solution in such a way that it adequately constrains the process so that movements from the intended specification can be detected; and second by checking for specifically listed types of non-conformance. This helps reduce the burden of anticipating all possible non-conformance types associated with approaches such as Kabbaj et al. [1] and Zazworka et al. [7].

Since GARDEN is implemented at a level that can evaluate process enactment data as soon as it becomes available, this enables GARDEN to detect non-conformance as quickly as possible. Also, since GARDEN is associated with specific parts of the process specification, when the process evolves and changes only the parts of the solution associated with the affected parts of the changes process need to be re-addressed, helping to reduce maintenance cost.

As GARDEN detects instances of non-conformance, it records them to a log, where each non-conformance instance is related to the specific part of the process responsible for the non-conformance occurring. Along with the non-conformance data logged, there are provisions for a severity rating and for remedial action to be logged in future, so that in future analysis we can see what the relative severity or impact the non-conformance had, what actions were subsequently taken, and an indication of the success of those actions. This helps decision making down the line, and helps improve the process in the long run.

The GARDEN approach, once developed to maturity, was tested via a variety of mediums. The approach was tested via a simulation engine that was built to simulate a wide variety of process types to ensure GARDEN could successfully detect non-conformance in the process enactments when they occurred. This implementation was presented in chapter 6. GARDEN was also tested in a real world process environment, where a lot of real data was collected and evaluated both before and after the integration of GARDEN. The results of this test were presented to show how GARDEN helped improve the process by summarizing and depicting the process data resulting from a large number of process enactments recorded before implementation and after. This case study was presented in chapter 7. The data from both phases of this experiment is also available in appendix A and appendix B of this thesis.

9.4 Summary of Contributions

This section will readdress the contributions made by GARDEN, both revisiting the achieved objectives that were set in chapter 3 and evaluated in chapter 8. Also, the unexpected benefits listed in chapter 8 discovered through evaluating GARDEN are given in summary form. Finally, the benefits yielded from cataloguing the literature are also discussed.

9.4.1 Detecting Non-Conformance

As can be clearly seen from the case studies, rule-based non-conformance detection is a successful methodology. This has been verified by the later inception of the rule based methodology introduced by Kabbaj et al. [1] and Zazworka et al. [7]. GARDEN utilises the power of rule-based detection and consolidates it into "rule sets", which are designed to be easy to define, easy to implement, easy to maintain and are inherently flexible and scalable with the process. The real contribution of this methodology is the power it gives the domain expert to practically tailor the GARDEN solution to any situation.

The rule set approach that GARDEN takes to detecting non-conformance is simple to understand and implement. Being rule-based, it successfully uses an approach that has since been used by other researchers [1] [7] in this area to successfully detect non-conformance in processes.

GARDEN employs a methodology that is clear, modular and uncoupled with the process. It is this methodology for detecting process non-conformance that lays the foundation for achieving the additional benefits and contributions covered in this section.

9.4.2 Consolidation of language, terms and concepts

As covered in great detail in chapter 2, process non-conformance detection is an area that is relatively specialized and "new" relative to the research world. It is not new in the research domain in terms of age, as research has existed in this field as early as Cugola et al. in 1995 [2]. Rather, the available literature and the state of the research suggests that process non-conformance detection has not yet matured as a research area and as such, much of the language is inconsistent from paper to paper.

This is a common phenomenon in new research areas [3] and one of the challenges is to consolidate the language and standardize the terms used. Until this happens naturally through publication, the need still remains to define quite explicitly what we mean when we use certain terms, such as "deviation" and "inconsistency". This is covered comprehensively in chapter 2. The contribution here is that the language discrepancy is acknowledged and a standard offered that if accepted, could apply to all the previous work conducted in process non-conformance detection. It also clarifies to those interested in this area the scope of non-conformance, what constitutes non-conformance and what does not.

Through illustrating the concepts related to process non-conformance as provided in Figure 3, the existing terminology is clarified and standardised. This makes it much easier to move forward with this line of research and helps make the research conducted by others much easier to understand.

9.4.3 Minimising interference to the target process

One important consideration when applying a non-conformance detection approach to a process is to ensure there is minimal interference with the way the process was originally set up and the people involved. With observational approaches such as Kabbaj et al. [1] or Huo et al. [4], this did not seem to be a problem although it was not addressed as a consideration in the literature.

The problem with interference is that it redirects resources allocated to the process elsewhere. As shown in chapter 8, section 8.2.2, GARDEN was designed specifically to have minimal impact on the existing process environment and the people involved in the process. This is achieved through flexible applicability and easy implementation and maintenance. A domain expert only has to set the GARDEN approach up from scratch once to have the power to apply it to only the sections of the process that are necessary. This removes interference with all other parts of the process.

The GARDEN approach clearly shows how interference is minimised when it is being applied to a target process, which helps maintain the benefits which non-conformance detection yields for the process.

9.4.4 Minimising the effort and cost of implementation and maintenance

As shown predominantly in chapter 5 where an explanation of the design presented in chapter 4 is practically applied, GARDEN is an approach that can be applied as minimally or as comprehensively as desired.

With all other process non-conformance detection approaches presented thus far, it seems that implementation can only be achieved in an “all or nothing” way. If, for example, a process existed where only part of it needed to be monitored for non-conformance and the effort required to cover the entire process was undesired then the other approaches could not accommodate this.

GARDEN addresses this shortcoming if the situation demands it, which increases its practical desirability. As GARDEN is implemented in the form of modular rule sets, it can be applied only to the parts of the process that require addressing. This is one aspect of GARDEN that reduces implementation effort.

Also, because GARDEN can be applied separately to different parts of the process, when the process evolves only the aspects of GARDEN that are applied to the affected parts of the process require changing. The entire solution does not need to be reformulated like other approaches [4] [5] [6] [7]. This significantly reduces the maintenance effort required for using GARDEN.

The overall benefit, the proof of effort and cost minimisation was shown in chapter 5 and 6, but also in the data presented in chapter 8 section 8.2.3. Figure 40 provides an excellent indication of the evidence of effort minimisation through depicting data on how long it takes for an example GARDEN implementation. Furthermore, after GARDEN has been implemented this effort and cost requirement is drastically reduced since only the affected parts require addressing when the process changes.

Lastly, GARDEN does not require a large repository of enactment data in which to form a reference model, which further reduces the effort required, compared to “process discovery” approaches such as Huo et al. [4] and Cook and Wolf [5].

9.4.5 Providing a highly flexible and scalable solution

The conception of a non-conformance detection approach that is suitable for application to any process and also capable of evolving with it is a significant contribution to the body of research in terms of practicality. As has been mentioned several times in this thesis, other approaches, although capable of successfully detecting non-conformance, are still not realistically practical for various reasons. This could be because of the amount of pre-existing data required in order to build a reference model to detect non-conformance [4] [5], the effort required in building a reference model and having to rebuild it from scratch every time the process changes and evolves [4] [6].

Much effort has gone into ensuring that GARDEN can be applied to any process as desired as long as it complies with a small number of conditions that GARDEN requires in order for it to operate. GARDEN has been designed to be applicable irrespective of whether there is previous enactment data available or how often the process is realistically enacted.

GARDEN has also been designed with modularity in mind. The solution ensures that when the process evolves, only the aspects of GARDEN applied to the affected sections of the process need to be readdressed. The rest of the implementation can be trusted to work as normal, which grants GARDEN a much higher level of flexibility and scalability than other approaches that have to be implemented again when the process is modified.

For these reasons, GARDEN offers a significant practical set of benefits to a situation where non-conformance detection is important for a real process.

9.4.6 Keeping the approach generic

One of the goals set for GARDEN was that it could be applied generically, regardless of the target process domain. Most of the literature in this area is aimed directly at the software process, and the rest at other specific domains such as manufacturing or business processes. GARDEN was designed with the intention that it could be applied to processes in all domains. This is not to allege that other approaches are incapable of being applied to cross-domain processes, but it is not explicitly stated in the literature. The reason GARDEN was built with a generic application in mind was to further increase its practical appeal.

Finding the balance between developing an approach that is highly useful but also has a narrow field of application uses, and an approach that can be applied widely but is too generic to be particularly useful is a delicate task. With GARDEN, the offering of easy and fast customization was seen as a viable solution to keeping the approach as generic as possible but also highly useful when it is practically applied.

Maintaining widespread applicability to processes has not really been addressed fully in the literature. The mechanisms GARDEN employs to maximise its applicability amongst a wide range of processes, but also to be easily tailored for a specific process provide an important benefit few other approaches are capable of providing.

9.4.7 Minimising the time to detection

One characteristic common to process non-conformance detection approaches is that they depend on the process enactment data to be available for some form of evaluation. Typically, most other approaches must wait until the process has concluded before they can begin to use the process enactment data.

GARDEN has shown, especially in chapter 5, that it is entirely possible in some cases (depending on each individual process environment) to begin evaluating the enactment data as it is being compiled, without having to wait for a complete enactment "model" as in other approaches such as Huo et al. [4] and Cîmpan and Oquendo [6].

Time to detection has been stated several times in this thesis as an issue of importance, because the longer the period of time between occurrence and detection, the greater the potential of damaging consequences. Oddly, however, speedy detection has not been a key issue addressed in the literature, so it is therefore a key contribution of the GARDEN approach.

Through a modular design where process enactment data can be executed as soon as it becomes available, GARDEN takes a positive step forward in reducing the amount of time "wasted" between the non-conformance taking place and our knowledge that it has occurred. Section 8.2.7 in chapter 8 provides ample evidence to show how GARDEN helps improve time to detection compared to other approaches, especially approaches that require models to be built after the enactment data has been compiled [4] [5] [6].

9.4.8 Future considerations

Addressing future considerations is an important contribution to the research field because GARDEN begins to address what the way forward should be once non-conformance is detected. Different people will address non-conformance in different ways once it is detected, but all will want as much information as possible about the non-conformance so they can make an informed decision. Supporting this decision, whatever it may be, is something to which GARDEN has contributed that has so far not been discussed in any of the preceding literature on process non-conformance detection.

Although non-conformance severity has been touched on in approaches such as Cugola et al. [2] and Cîmpan and Oquendo [6], there has been nothing concrete so far in terms of relevant importance. This can be considered at several levels, including how far a value has deviated from the expected, how important certain aspects of the process are in relation to others (depending on the consequences of non-conformance occurring in one part of the process compared to another), to how important the process itself may be in an organizational context.

GARDEN introduces considerations such as relative severity and uses them along with historical non-conformance data to build a recommendations log, indicating what could be done and what has been done in the past when certain non-conformance types occur. Figure 41 and Figure 42 in chapter 8 clearly show how the concept of relative severity provide the domain expert with a much more meaningful idea of how "severe" a particular instance of non-conformance might be. This body of work is unique so far in this research area and contributes significantly to what issues need to be considered once non-conformance has actually been detected, something that so far has not been explored.

9.5 Future Work

Future work on both GARDEN and process non-conformance detection in general are likely to be related to the issues discussed in chapter 3, and the contributions listed in this chapter in section 9.4. This thesis creates a compelling argument that these considerations and issues are key to the improvement of non-conformance detection approaches. It is for this reason that these issues were highlighted and addressed with the objective of creating a practical solution.

Foremost in respect to GARDEN itself, there is room to extend and improve the workings of the logic-type attribute of the rule sets as covered in chapter 8, section 8.3.2. Although it seems that most processes will lend themselves well to the current implementation, it is possible that some more complicated processes may benefit from having this feature extended and this will be looked at more closely to further extend the range of processes to which GARDEN can be applicable.

The balance of applicability will need to be challenged in a more formal manner. GARDEN, takes an approach designed to be generic and widespread but is also highly customizable to maintain its usefulness in specific situations. For future work in this research area, improvements in customization ease will be explored.

In this vein, specific use frameworks of GARDEN could be tailored for certain process types. GARDEN itself is a generic methodology, but frameworks could be built specifically aimed at processes in one certain process domain. These frameworks would use all the principles of GARDEN but would provide the "plumbing", so to speak, that would cater for processes running in the environment the framework was built for. This "plumbing" would include data acquisition and massaging into appropriate storage or rule set definition and application engines as examples, similar to the engine presented in chapter 6, but practically tailored for real world use.

If process non-conformance detection research becomes an accepted and practiced part of process improvement, a large part of the future of this research lies in the ability to accurately gauge a severity rating on non-conformance instances and also to provide remedial advice. When non-conformance data becomes easily and readily available, we can create real value by guiding people as to how to make the best use of that information. Therefore, maximization of available data will need to be explored and improved and then mined to provide some practically useful information both on the decisions to be made post-non-conformance detection and also on how to improve the process as a whole.

9.6 Conclusion

In summary, this thesis has presented a practical approach to process non-conformance detection named GARDEN. The presentation of GARDEN included a detailed analysis of the existing literature and current issues facing the research area of process non-conformance detection. A list of goals and objectives were clearly defined to highlight how GARDEN could contribute a unique offering in this research area.

The design of the GARDEN solution was presented in two chapters, one to show the inner workings, concepts and high level design of the solution and the other showing how to take this design and apply it to a process in a practical setting. Two chapters showing different implementations and testing of GARDEN were also presented, showing GARDEN working on actual processes and how it achieved the stated objectives whilst detecting non-conformance instances occurring in those processes.

Finally, the GARDEN approach was evaluated in terms of its design, application, implementation and case study testing in a comprehensive evaluation chapter. Overall, this thesis has shown how process non-conformance detection can make a significant contribution to process improvement. It really depends on how much of a practical solution can be conceived, which was the primary intention in the design and development of GARDEN.

The project ends in a state where there is considerable opportunity for future research both in terms of improving GARDEN and delving further into process improvement via non-conformance detection. Specifically, considerations after detection present an area previously unexplored that could contribute a great deal more to the current state of this research.

References

[1] M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

[2] G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

[3] J.H. Kroeze, M.C. Matthee, T.J.D. Bothma, Differentiating data- and text-mining terminology, Conference of the South African institute of computer scientists and information technologists on Enablement through technology, South African Institute for Computer Scientists and Information Technologists, September, 2003.

[4] M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

[5] J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 7 Issue 3, July 1998.

[6] S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

[7] N. Zazworka, V.R. Basili, F. Shull, Tool Supported Detection and Judgment of Nonconformance in Process Execution, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, October 2009.

Bibliography

“Process” definition, Merriam-Webster, http://www.merriam-webster.com/dictionary/process, May, 2010.

“Specification” definition, Merriam-Webster, http://www.merriam-webster.com/dictionary/specification, May, 2010.

“Workflow Management Coalition,” http://www.wfmc.org.

A. Blyth, Business process re-engineering: What is it? ACM SIGGROUP Bulletin, Volume 18 Issue 1, April 1997.

A. Borgida, T. Murata, Tolerating exceptions in workflows: a unified framework for data and processes, ACM SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99, Volume 24 Issue 2, ACM Press, March 1999.

A. Caetano, A.R. Silva, J. Tribolet, Using roles and business objects to model and understand business processes, ACM symposium on Applied computing, ACM, March, 2005.

A. Caetano, J. Tribolet, Modeling organizational actors and business processes, ACM symposium on Applied computing, ACM, April, 2006.

A. Fuggetta, Software Process: A Roadmap, Proceedings of the Conference on the Future of Software Engineering, ACM Press, May 2000.

A. Rozinat, W.M.P. van der Aalst, Conformance Checking of Processes Based on Monitoring Real Behaviour, Journal of Information Systems, Volume 33, Issue 1, Elsevier Science Ltd., Oxford, UK, March 2008.

A. Zisman, W. Emmerich, A. Finkelstein, Using XML to Build Consistency Rules for Distributed Specifications, Tenth International Workshop on Software Specification and Design (IWSSD'00), IEEE, November, 2000.

B. Curtis, M.I. Kellner, J. Over, Process Modeling, Communications of the ACM, Volume 35 Issue 9, ACM Press, September, 1992.

B. Curtis, Software Process Improvement: Methods and Lessons Learned, Proceedings of the 19th international conference on Software engineering, ACM Press, May 1997.

B.C. Boynton, Identification of process improvement methodologies with application in information security, 4th annual conference on Information security curriculum development, ACM, September 2007.

B.M. Carter, J.Y.C. Lin, M.E. Orlowska, Customizing internal activity behaviour for flexible process enforcement, 15th Australasian database conference, Australian Computer Society Inc, Volume 27, January 2004.

Blizzard Entertainment Press Release, "WORLD OF WARCRAFT® SUBSCRIBER BASE REACHES 11.5 MILLION WORLDWIDE", http://www.blizzard.com/us/press/081121.html, 23 December 2008

C.D. Klingler, A Case Study in Process Definition, Proceedings of the conference on TRI-Ada, ACM, October 1993.

D. Avrilionis, N. Belkhatir, P.Y. Cunin, A unified framework for software process enactment and improvement, 4th International Conference on the Software Process, Brighton, England, 3-5 December, 1996.

D. Bogia and S. Kaplan, Flexibility and Control for Dynamic Workflows in the wOrlds Environment, Proceedings of the Conference on Organisational Computing Systems, ACM Press, Milpitas, CA, November 1995.

D. Bogia, S. Kaplan, Flexibility and Control for Dynamic Workflows in the wOrlds Environment, Proceedings of the Conference on Organisational Computing Systems, ACM Press, Milpitas, CA, November 1995.

D.E. Perry, A.L. Wolf, Session 1: People, Processes, and Practice, Proceedings of the 9th International Software Process Workshop, IEEE Computer Society Press, October 5-7 1994, Airlie, Virginia, USA.

D.M. Ferrin, M.J. Miller, D. Muthler, Lean Sigma and simulation, so what's the correlation? V2, Winter Simulation Conference, 2005 Proceedings of the, Bus. Prototyping Inc., Naperville, IL, USA, 4-7 Dec. 2005.

E. Ellmer, W. Emmerich, A. Finkelstein, D. Smolko, A. Zisman, Consistency Management of Distributed Documents using XML and Related Technologies, UCL-CS Research Note 99/94, 1999.

G. Cugola, E. Di Nitto, A. Fuggetta, C. Ghezzi, A framework for formalizing inconsistencies and deviations in human-centered systems, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 5 Issue 3, ACM Press, July 1996.

G. Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

G. Cugola, Tolerating Deviations in Process Support Systems VI Flexible Enactment of Process Models, IEEE, Transactions on Software Engineering, vol. 24, No 11, 1998.

G. Engels, W. Schäfer, R. Balzer, V. Gruhn, Process-centered software engineering environments: academic and industrial perspectives, Proceedings of the 23rd International Conference on Software Engineering ICSE '01, IEEE Computer Society, July 2001.

G. Grossmann, M. Schrefl, M. Stumptner, Modelling inter-process dependencies with high-level business process modelling languages, fifth on Asia-Pacific conference on conceptual modelling, Australian Computer Society Inc, Volume 79, January, 2008.

G.Cugola, E. Di Nitto, C. Ghezzi, M. Mantione, How to deal with deviations during process model enactment, Proceedings of the 17th international conference on Software engineering, ACM Press, April 1995.

http://www.mmogchart.com/Chart7.html, MMOG Market Share at April 2008

I. Sommerville, Software Engineering, 8th edition, Addison Wesley, 2007.

J. Banks, J.S. Carson, Discrete-Event System Simulation, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

J. Eder, W. Liebhart, The Workflow Activity Model (WAMO), 3rd international conference on Cooperative Information Systems (CoopIs), 1995.

J. Mendling, G. Neumann, W. van der Aalst, On the correlation between process model metrics and errors", 26th International Conference on Conceptual Modeling, Volume 83, Australian Computer Society Inc, November, 2007.

J.D. Reese, N.G. Leveson, Software deviation analysis, Proceedings of the 19th international conference on Software engineering, ACM Press, May 1997.

J.E. Cook, A.L. Wolf, Automating process discovery through event-data analysis, Proceedings of the 17th international conference on Software engineering ICSE '95, ACM Press, April 1995.

J.E. Cook, A.L. Wolf, Discovering models of software processes from event-based data, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 7 Issue 3, July 1998.

J.E. Cook, A.L. Wolf, Software process validation: quantitatively measuring the correspondence of a process to a model, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 8 Issue 2, ACM Press, April 1999.

J.H. Kroeze, M.C. Matthee, T.J.D. Bothma, Differentiating data- and text-mining terminology, Conference of the South African institute of computer scientists and information technologists on Enablement through technology, South African Institute for Computer Scientists and Information Technologists, September, 2003.

J.M. Cobleigh, L.A. Clarke, L.J. Osterweil, Verifying Properties of Process Definitions, international symposium on Software testing and analysis (ISSTA), ACM, August, 2000.

J.S. Brown and P. Duguid, Balancing Act: How to Capture Knowledge without Killing It, Harvard Business Review, Vol. 78, No. 3, 2000, pp. 73-80.

L. Aversano, G. Canfora, Introducing eservices in business process models, Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02, ACM Press, July 2002.

L. Osterweil, Automated Support for the Enactment of Rigorously Described Software Processes, 4th international software process workshop on Representing and enacting the software process, ACM, April, 1988.

L.F.S. Silva, G.H. Travassos, Tool-Supported Unobtrusive Evaluation of Software Engineering Process Conformance, International Symposium on Empirical Software Engineering, ISESE '04, ISBN:0-7695-2165-7, September 2004.

M.A.A da Silva, R. Bendraou, X. Blanc, and M.P. Gervais, Early Deviation Detection in Modeling Activities of MDE Processes, Proceedings of the 13th international conference on Model driven engineering languages and systems: Part II, Oslo, Norway, October, 2010.

M. Dowson, B. Nejmeh, W. Riddle, Concepts for Process Definition and Support, Proceedings of the 6th International Software Process Workshop, IEEE Computer Society Press, October 28-31 1990, Hakodate, Japan.

M. Huo, H. Zhang, R. Jeffery, An Exploratory Study of Process Enactment as Input to Software Process Improvement, International Workshop on Software Quality at International Conference on Software Engineering (ICSE), Shanghai, 2006.

M. Kabbaj, R. Lbath, B. Coulette, A Deviation-tolerant Approach to Software Process Evolution, Ninth international workshop on Principles of software evolution (IWPSE'07), Dubrovnik, Croatia, September 2007.

M. Komuro, Far east experience papers: software process: Experiences of applying SPC techniques to software development processes, Proceeding of the 28th international conference on Software engineering ICSE '06, ACM Press, May 2006.

M. Ramage, Engineering a smooth flow? The links between workflow and business process reengineering, MSc thesis 1994, University of Sussex, England.

M. Sherriff, N. Nagappan, L. Williams, M. Vouk, Early estimation of defect density using an in-process Haskell metrics model, 1st international workshop on Advances in model-based testing (A-MOST), ACM, July, 2005.

M. Van Hilst, P.K. Garg, C. Lo, Repository mining and Six Sigma for process improvement, ACM SIGSOFT Software Engineering Notes, Proceedings of the 2005 international workshop on Mining software repositories MSR '05, Volume 30 Issue 4, ACM Press, May 2005.

M.A. Lantzy, Application of statistical process control to the software process, Proceedings of the ninth Washington Ada symposium on Ada: Empowering software users and developers, ACM Press, July 1992.

M. Hammer, J. Champy, Reengineering the Cooperation, Harper Business, New York, 1993.

M.J. Miller, D.M. Ferrin, T. Flynn, M. Ashby, K.P. White, Jr., M.G. Mauer, Using RFID technologies to capture simulation data in a hospital emergency department, 38th conference on Winter simulation, Winter Simulation Conference, December, 2006.

M.M. Fathee, R. Redd, D. Gorgas, B. Modarres, The effects of complexity on business processes reengineering: values and limitations of modeling and simulation technologies, 30th conference on Winter Simulation, IEEE Computer Society Press, December 1998.

N. Park, H. Kim, K. Kim, H. Park, J. Chun, Y. Hwang, An eLearning Activity Control Model for SCORM's Sequencing Prerequisites, Fourth International Conference on Semantics, Knowledge and Grid, IEEE, 3-5 December 2008, pp. 322 – 329.

N. Zazworka, K. Stapel, E. Knauss, F. Shull, V.R. Basili, K. Schneider, Are Developers Complying with the Process: An XP Study, Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Bolzano, Italy, September 2010.

N. Zazworka, V.R. Basili, F. Shull, Tool Supported Detection and Judgment of Nonconformance in Process Execution, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, October 2009.

N.H. Madhavji, K. Toubache, E. Lynch, The IBM-McGill project on software process, conference of the Centre for Advanced Studies on Collaborative research, IBM Press, October 1991.

N.H. Robertson, T. Perera, General manufacturing applications: feasibility for automatic data collection, 33rd Conference on Winter Simulation, IEEE Computer Society, December, 2001.

Official CAPTCHA Site, http://www.captcha.net/

P. Checkland, Systems Thinking, Systems Practice, Chichester: John Wiley & Sons, Chapter 2, 1981.

P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, M. Teschke, A comprehensive approach to flexibility in workflow management systems, ACM SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99, Volume 24 Issue 2, ACM Press, March 1999.

P. Jalote, A. Saxena, Optimum control limits for employing statistical process control in software process, IEEE Transactions on Software Engineering, Volume 28, Issue 12, Dec. 2002 Page(s): 1126 – 1134.

P. Struss, Deviation Models Revisited, Working Papers of the 15th International Workshop on Principles of Diagnosis (DX-04), June 2004.

R. Gillespie, Manufacturing knowledge: a history of the Hawthorne experiments, Cambridge : Cambridge University Press, 1991.

R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, G. Coleman, Detection and explanation of anomalous activities: representing activities as bags of event n-grams, Conference on Computer Vision and Pattern Recognition, IEEE, Volume 1, 20-25 June, 2005, pp. 1031 – 1038.

R. Harper, M.J. O'Loughlin, Manufacturing process analysis—tools and applications, Proceedings of the 19th conference on Winter Simulation, ACM, 1987.

R.E. Gallardo-Valencia, S.E. Sim, Planning and improvisation in software processes, ACM, Volume 14, Issue 1, September, 2007.

R.L. Glass, Software Creativity, Englewood Cliffs, New Jersey, Prentice-Hall, 1995.

S. Acuna, A. De Antonio, X. Ferré, M. López, L. Maté, The Software Process: Modelling, Evaluation, and Improvement, Handbook of Software Engineering and Knowledge Engineering, World Scientific Publishing Company, 2000.

S. Cîmpan, F. Oquendo, Dealing with software process deviations using fuzzy logic based monitoring, ACM SIGAPP Applied Computing Review, Volume 8 Issue 2, ACM Press, December 2000.

S. Sørumgård, “Verification of Process Conformance in Empirical Studies of Software Development”. Ph.D. thesis, Norwegian University of Science and Technology, 1997.

S. Thompson and T. Torabi, An Observational Approach to Practical Process Non-Conformance Detection, The Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), London, England, August 4 – 6, 2009.

S. Thompson, T. Torabi, An Observational Approach to Practical Process Non-Conformance Detection, Journal of Information Technology Review, ISSN 0976-2922, 2010.

S. Thompson, T. Torabi, Determining Severity and Recommendations in Process Non-Conformance Instances, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, July, 2008.

S. Thompson, T. Torabi, P. Joshi, A Framework to Detect Deviations During Process Enactment, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

Stanford Encyclopedia of Philosophy, Stanford University, California, USA, http://plato.stanford.edu/entries/turing-test/

S.U. Hani, Impact of process improvement on software development predictions, for measuring software development project's performance benefits, Proceedings of the 7th International Conference on Frontiers of Information Technology, Abbottabad, Pakistan, December 2009.

T. Barothy, M. Peterhans, K. Bauknecht, Business process reengineering: emergence of a new research field, ACM SIGOIS Bulletin, Volume 16 Issue 1, August 1995.

T. Davenport, Process Innovation: Re-engineering Work through Information Technology, Harvard Business School Press, Boston MA, 1993.

T. Dybå, Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context, ACM SIGSOFT Software Engineering Notes, Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-11, Volume 28 Issue 5, ACM Press, September 2003.

V. Ambriola, R. Conradi, A. Fuggetta, Assessing process-centered software engineering environments, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 6 Issue 3, ACM Press, July 1997.

V.R. Basili, F.E. McGarry, R. Pajerski, M.V. Zelkowitz, Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory, Proceedings of the 24th International Conference on Software Engineering, ACM Press, May 2002.

W.A. Florac, A.D. Carleton, J.R. Barnard, Statistical Process Control: Analyzing a Space Shuttle Onboard Software Process, IEEE Software, vol. 17, no. 4, July/Aug. 2000, pp. 97–106.

W.A. Florac, A.D. Carleton, Measuring the Software Process: Statistical Process Control for Process Improvement, Addison-Wesley, 1999.

W.M.P. van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, H.M.W. Verbeek, Choreography Conformance Checking: An Approach based on BPEL and Petri Nets, Proceedings of the Seminar on The Role of Business Processes in Service Oriented Architectures, Dagstuhl, Germany, July 2006.

W.M.P Van der Aalst, S. Jablonski, Dealing with workflow change: identification of issues and solutions, International Journal of Computer Systems Science and Engineering, pp 267-276, September 2000.

World of Warcraft official game forums, http://forums.worldofwarcraft.com/index.html

X. Zhao, K. Chan, M. Li, Applying agent technology to software process modelling and process-centered software engineering environment, Proceedings of the 2005 ACM symposium on Applied computing SAC '05, ACM Press, March 2005.

Y. Rezgui, F. Marir, G. Cooper, J. Yip, P. Brandon, A Case-Based Approach to Construction Process Activity Specification, Intelligent Information Systems, 8-10 Dec, 1997, pp. 293 – 297.

Y.Z. Wei, L. Moreau, N.R. Jennings, A market-based approach to recommender systems, ACM Transactions on Information Systems (TOIS), Volume 23 Issue 3, ACM Press, July 2005.

Z. Zhou, S. Bhiri, M. Hauswirth, Control and data dependencies in business processes based on semantic business activities, 10th International Conference on Information Integration and Web-based Applications & Services, ACM, November, 2008.

Appendices

Appendix A – Online Implementation First Phase Dataset Sample

Attached is the actual survey data from the first phase of the online implementation covered in chapter 7. The field sizes have been narrowed, which unfortunately cut some of the data out however this is necessary to keep the large amount of data logged during this case study in a printable area. Some data unrelated to the results and evaluation of this case study has been omitted from this appendix, such as the action log of users and their stored session data, which contains participant identifying information. Furthermore, this data is censored by blacking out words in some places due to the vulgar language used by some of the participants, however it has not been altered in any way.

id	session	start time	end time	server	country	city	level	char gend	race	class	real genc	age	play time	hours wee
1	4983d9eeb78el	2009-01-31 04:56:14	2009-01-31 04:58:18	Barthilas	Australia	Melb'ourn	77	Male	Night Elf	Druid	Male	27	3 years on	4
2	4983e22c9d7b8	2009-01-31 05:31:24	2009-01-31 05:32:58	Barthilas	Australia	Gold Coast	80	Male	Dwarf	Paladin	Male	18	4 Years	30+
3	4983e306ea8a	2009-01-31 05:35:02	2009-01-31 05:37:04	Barthilas	australia	melbourne	80	Male	Tauren	Warrior	Male	21	4 years	42 hours
4	4983e317b6712	2009-01-31 05:35:19	2009-01-31 05:37:52	Tichondrius	Australia	Mackay	80	Female	Troll	Shaman	Not Speci	32	3+ years	42+
5	4983e3c160dfb	2009-01-31 05:38:09	2009-01-31 05:39:37	Barthilas	United Sta	San Diego	80	Female	Night Elf	Druid	Female	19	4 years.	12+
6	4983e3c5ebbd	2009-01-31 05:38:13	2009-01-31 05:40:50	Frostmourne/	Australia	Torquay (Victor	80	Male	Blood El	Death Kn	Male	19	Since Oper	Depends..
7	4983e4c1e2c7c	2009-01-31 05:42:25	2009-01-31 05:45:22	Frostmourne	Australia	Brisbane	80	Male	Night Elf	Druid	Male	22	2 years	20ish
8	4983e61742ac6	2009-01-31 05:48:07	2009-01-31 05:49:01	Barthilas	Australia	Sydney	80	Male	Undead	Mage	Male	22	18 months	10-30
9	4983e7dc7719f	2009-01-31 05:55:40	2009-01-31 05:56:38	Barthilas	Australia	Melbourne	80	Male	Night Elf	Warrior	Male	19	3 years	12
10	4983ec92b6fec	2009-01-31 06:15:46	2009-01-31 06:16:58	Agamaggan	Washingtc	Spanaway	80	Female	Tauren	Shaman	Female	22	3 years	A LOT
11	4983eda408bc0	2009-01-31 06:20:20	2009-01-31 06:22:28	Bleeding Holl	USA! USA	Fair Lawn, Nev	80	Female	Gnome	Warlock	Male	16	Since relea	20+
12	4983f16414631	2009-01-31 06:36:20	2009-01-31 06:37:28	Azshara	Belarus	Minsk	78	Female	Troll	Death Kn	Male	17	1.5 years	25
13	4983f16414631	2009-01-31 06:37:28	2009-01-31 06:37:39	Azshara	Belarus	Minsk	78	Female	Troll	Death Kn	Male	17	1.5 years	25
14	4983f16414631	2009-01-31 06:37:39	2009-01-31 06:37:47	Azshara	Belarus	Minsk	78	Female	Troll	Death Kn	Male	17	1.5 years	25
15	4983fb00687aa	2009-01-31 07:17:20	2009-01-31 07:18:47	Azshara	canada	Sault ste marie	80	Male	Human	Rogue	Male	17	2 years	30-40
16	4983fb00687aa	2009-01-31 07:18:47	2009-01-31 07:18:55	Azshara	canada	Sault ste marie	80	Male	Human	Rogue	Male	17	2 years	30-40
17	4983fb969c04a	2009-01-31 07:19:50	2009-01-31 07:26:22	Lightbringer	Canada	Sault Ste Marie	80	Female	Blood El	Paladin	Male	21	2 years	60+
18	4983fd0f6c4cc2	2009-01-31 07:26:07	2009-01-31 07:27:30	Sargeras	USA	Buffalo	80	Male	Night Elf	Druid	Male	23	3yrs	20+
19	4983fd0f6c4cc2	2009-01-31 07:27:30	2009-01-31 07:27:38	Sargeras	USA	Buffalo	80	Male	Night Elf	Druid	Male	23	3yrs	20+
20	498408c35b8ec	2009-01-31 08:16:03	2009-01-31 08:17:16	Bleeding Holl	Canada	Nanaimo	80	Male	Draenei	Shaman	Male	16	2 years	20-40
21	49840bf886a48	2009-01-31 08:29:44	2009-01-31 08:32:02	Bleeding Holl	US	Athens, Ga	80	Male	Night Elf	Druid	Male	32	almost 6 m	10-20
22	498417c50704k	2009-01-31 09:20:05	2009-01-31 09:23:04	Proudmoore	United Sta	Kenosha, WI	80	Male	Human	Warlock	Male	25	1.5 years	30-40
23	49841fb54d66f	2009-01-31 09:53:57	2009-01-31 09:55:19	Frostmourne	Australia	Melbourne	80	Male	Undead	Warlock	Male	17	2 years	35
24	49842857d85e	2009-01-31 10:30:47	2009-01-31 10:32:15	Proudmoore	Australia	Cairns	80	Female	Gnome	Rogue	Male	25	2 years	40
25	49842b8e62ad	2009-01-31 10:44:30	2009-01-31 10:45:24	frostmourne	australia	sydney	80	Female	Human	Paladin	Male	16	2 years	50ish?
26	49842b8e62ad	2009-01-31 10:45:24	2009-01-31 10:46:20	frostmourne	australia	sydney	80	Female	Human	Paladin	Male	16	2 years	50ish?
27	4984419a27aa	2009-01-31 12:18:34	2009-01-31 12:19:22	Proudmoore	Australia	Brisbane	70	Male	Undead	Warrior	Male	21	2 years	0
28	49846763362al	2009-01-31 14:59:47	2009-01-31 15:02:26	Lightbringer	Canada	Fredericton	80	Female	Night Elf	Priest	Female	19	2 years.	15-20
29	4984f06146232	2009-02-01 00:44:17	2009-02-01 00:45:42	Runetotem U	Us	rural area	72 (alt	Female	Tauren	Shaman	Female	39	3 years	6-12
30	498626d8a45b7	2009-02-01 22:48:56	2009-02-01 22:49:57	Executus	USA	Detroit, MI	80	Female	Draenei	Shaman	Female	??	3 years	too many
31	498626d8a45b7	2009-02-01 22:49:57	2009-02-01 22:50:10	Executus	USA	Detroit, MI	80	Female	Draenei	Shaman	Female	??	3 years	too many
32	49861d0a6a893	2009-02-01 22:50:06	2009-02-01 22:51:15	Frostmourne	Australia	Melbourne	sixty	Male	Undead	Priest	Male	25	2 yearsish	10-15
33	49862f3827e5e	2009-02-01 23:24:40	2009-02-01 23:26:27	Hellscream	Canada	Toronto	77	Male	Gnome	Warrior	Male	21	4 years	1-4
34	49862f3827e5e	2009-02-01 23:26:27	2009-02-01 23:26:41	Hellscream	Canada	Toronto	77	Male	Gnome	Warrior	Male	21	4 years	1-4
35	498648ed0302l	2009-02-02 01:14:21	2009-02-02 01:15:39	Silver Hand	USA	McCook, NE	80	Male	Blood El	Rogue	Male	18	2 years	10-15
36	4986515badc30	2009-02-02 01:50:19	2009-02-02 01:55:10	BARTHILAS!	nz	nelly	80	Male	Tauren	Druid	Male	16	burning cru	40ish
37	49866d31e01d5	2009-02-02 03:49:05	2009-02-02 03:51:56	Bleeding Holl	U.S.	Tampa	80	Female	Draenei	Shaman	Male	17	2 and a hal	30
38	49867079ee76f	2009-02-02 04:03:05	2009-02-02 04:03:59	Stormscale	USA	Branson	80	Female	Blood El	Paladin	Male	28	4 years	20
39	4986708001138	2009-02-02 04:03:12	2009-02-02 04:06:03	Zul'jin	U.S	Raleigh	80	Male	Orc	Death Kn	Not Speci	24	Off and on	60
40	4986715e60ea	2009-02-02 04:06:54	2009-02-02 04:08:37	silvermoon	america fu	ny ny ny BBY	80	Male	Human	Paladin	Male	16 ar	since abou	a good soli

occupation	job time	game settings	captcha	valid	comments	real_start_time
web developer	40	Wrath of the Lich King	Y	Y	cool site!	2009-01-31 04:56:14
Student, University	20	Wrath of the Lich King	Y	Y		2009-01-31 05:31:24
electrician	40	Wrath of the Lich King	Y	Y		2009-01-31 05:35:02
N/A	0	Not-Specified	Y	Y		2009-01-31 05:35:19
Student	12+	Wrath of the Lich King	Y	Y	Omgz, good l	2009-01-31 05:38:09
Kitchen Hand	20	Wrath of the Lich King	Y	Y		2009-01-31 05:38:13
Salesperson	casual, change	Wrath of the Lich King	Y	Y	have fun time	2009-01-31 05:42:25
Uni Student + P/T Sales	50-80	Wrath of the Lich King	Y	Y		2009-01-31 05:48:07
Student/Casual Work	60	Wrath of the Lich King	Y	Y		2009-01-31 05:55:40
disabled	none	Wrath of the Lich King	Y	Y		2009-01-31 06:15:46
Student/Stock Boy at a s	school plus job	Wrath of the Lich King	Y	Y	Sorry about s	2009-01-31 06:20:20
Student	25	Wrath of the Lich King	N	N		2009-01-31 06:36:20
Student	25	Wrath of the Lich King	N	N		2009-01-31 06:36:20
Student	25	Wrath of the Lich King	Y	Y		2009-01-31 06:36:20
sales clerk	24	Wrath of the Lich King	N	N	nice survey	2009-01-31 07:17:20
sales clerk	24	Wrath of the Lich King	Y	Y	nice survey	2009-01-31 07:17:20
Student	20	Wrath of the Lich King	Y	Y	I think this su	2009-01-31 07:19:50
Military Fix medical equi	40hrs a week	Wrath of the Lich King	N	N	OBAMA FOR	2009-01-31 07:26:07
Military Fix medical equi	40hrs a week	Wrath of the Lich King	Y	Y	OBAMA FOR	2009-01-31 07:26:07
student/restaurant	10-20	Burning Crusade	Y	N		2009-01-31 08:16:03
Web Production Designe	30-40	Wrath of the Lich King	Y	Y		2009-01-31 08:29:44
Network Engineer	50	Wrath of the Lich King	Y	Y	Hope this hel	2009-01-31 09:20:05
year 12	50	Wrath of the Lich King	Y	Y		2009-01-31 09:53:57
Electrician	38	Wrath of the Lich King	Y	Y		2009-01-31 10:30:47
		Not-Specified	N	N		2009-01-31 10:44:30
student	35 hours	Wrath of the Lich King	Y	Y		2009-01-31 10:44:30
Student	30	Burning Crusade	Y	Y		2009-01-31 12:18:34
Cook	30-40 Hours	Wrath of the Lich King	Y	Y		2009-01-31 14:59:47
Medical assistant	40	Wrath of the Lich King	Y	N		2009-02-01 00:44:17
Hospice RN	40+	Wrath of the Lich King	N	N		2009-02-01 22:48:56
Hospice RN	40+	Wrath of the Lich King	Y	N		2009-02-01 22:48:56
student	12	Classic	Y	N		2009-02-01 22:50:06
Student; Kitchen Staff	18;16	Wrath of the Lich King	N	N		2009-02-01 23:24:40
Student; Kitchen Staff	18;16	Wrath of the Lich King	Y	Y		2009-02-01 23:24:40
student (college)	8	Wrath of the Lich King	Y	Y		2009-02-02 01:14:21
student	7hrs	Wrath of the Lich King	Y	Y	in the holiday	2009-02-02 01:50:19
Student/pizza boy	47	Wrath of the Lich King	Y	Y		2009-02-02 03:49:05
Page Layout	35	Wrath of the Lich King	Y	Y		2009-02-02 04:03:05
Unemployed; Last job: G	0; last job: 70-8	Wrath of the Lich King	Y	Y		2009-02-02 04:03:12
i wrk at deli / student	7	Wrath of the Lich King	Y	N	[redacted]	2009-02-02 04:06:54

41	4986728fd2821	2009-02-02 04:11:59	2009-02-02 04:13:42	Frostwolf	United Sta	Powell	80	Male	Draenei	Hunter	Male	16	2 years	35
42	4986728fd2821	2009-02-02 04:13:42	2009-02-02 04:13:51	Frostwolf	United Sta	Powell	80	Male	Draenei	Hunter	Male	16	2 years	35
43	49867cf4d2ee7	2009-02-02 04:56:20	2009-02-02 04:57:07	eldre'tha	canada	montreal	80	Female	Blood El	Paladin	Female	27	4	25
44	49867da12ea4	2009-02-02 04:59:13	2009-02-02 05:01:22	Lothar	USA	Natick	72	Female	Human	Mage	Female	19	4 months	35
45	49867e204722	2009-02-02 05:01:20	2009-02-02 05:02:40	Thunderhorn	United Sta	Houston	80	Male	Undead	Death Kn	Male	15	4 years	10
46	49867ed303f76	2009-02-02 05:04:19	2009-02-02 05:05:57	Zangermash	Australia	Perth	49	Male	Tauren	Shaman	Male	27	18 months	8 hours
47	49867ed303f76	2009-02-02 05:05:57	2009-02-02 05:06:36	Zangermash	Australia	Perth	49	Male	Tauren	Shaman	Male	27	18 months	8 hours
48	498682020170	2009-02-02 05:17:54	2009-02-02 05:19:47	Maelstrom	USA	Orlando	76	Male	Undead	Warlock	Male	25	4 years	10-15
49	4986840bdc12	2009-02-02 05:26:35	2009-02-02 05:28:15	Zul'jin	US	Glasgow	80	Female	Draenei	Shaman	Male	26	4 years, off	20
50	4986840bdc12	2009-02-02 05:28:15	2009-02-02 05:28:25	Zul'jin	US	Glasgow	80	Female	Draenei	Shaman	Male	26	4 years, off	20
51	49868650ae0b	2009-02-02 05:36:16	2009-02-02 05:38:04	Ursin	USA	Houston	80	Male	Night Elf	Hunter	Male	16	1 year	Many many
52	49868650ae0b	2009-02-02 05:38:04	2009-02-02 05:38:14	Ursin	USA	Houston	80	Male	Night Elf	Hunter	Male	16	1 year	Many many
53	498688d8a7c8	2009-02-02 05:47:04	2009-02-02 05:48:43	draenor	canada		80	Female	Night Elf	Druid	Male	28	4 years	9-10
54	498689230b10	2009-02-02 05:48:19	2009-02-02 05:49:39	Gorggonash	Australia	Sydney	80	Female	Undead	Mage	Female	21	1 year	20-30
55	49868a675930	2009-02-02 05:53:43	2009-02-02 05:55:45	Jubei'th	Australia	Toowoomba	80	Female	Blood El	Paladin	Male	15	3 Years? Al	About 10 a
56	49868a675930	2009-02-02 05:55:45	2009-02-02 05:55:57	Jubei'th	Australia	Toowoomba	80	Female	Blood El	Paladin	Male	15	3 Years? Al	About 10 a
57	498692b67446	2009-02-02 06:29:10	2009-02-02 06:30:49	Zul'jin	United Sta	Chicago	80	Female	Blood El	Paladin	Male	22	3.5 years	35
58	498692b67446	2009-02-02 06:30:49	2009-02-02 06:31:00	Zul'jin	United Sta	Chicago	80	Female	Blood El	Paladin	Male	22	3.5 years	35
59	498692b67446	2009-02-02 06:31:00	2009-02-02 06:31:15	Zul'jin	United Sta	Chicago	80	Female	Blood El	Paladin	Male	22	3.5 years	35
60	4986944f91532	2009-02-02 06:35:59	2009-02-02 06:38:19	Dentarg	United Sta	Salt Lake City	72	Female	Gnome	Mage	Female	22	2.75 years	5. forums:
61	498697fad1f17	2009-02-02 06:51:38	2009-02-02 06:52:11	Dreadmaul	Australia	Wollongong	80	Male	Night Elf	Priest	Male	17	3 Years	35+
62	498697fad1f17	2009-02-02 06:52:11	2009-02-02 06:52:28	Dreadmaul	Australia	Wollongong	80	Male	Night Elf	Priest	Male	17	3 Years	35+
63	498697fad1f17	2009-02-02 06:52:28	2009-02-02 06:52:34	Dreadmaul	Australia	Wollongong	80	Male	Night Elf	Priest	Male	17	3 Years	35+
64	498697fad1f17	2009-02-02 06:52:34	2009-02-02 06:52:43	Dreadmaul	Australia	Wollongong	80	Male	Night Elf	Priest	Male	17	3 Years	35+
65	498697fad1f17	2009-02-02 06:52:43	2009-02-02 06:52:51	Dreadmaul	Australia	Wollongong	80	Male	Night Elf	Priest	Male	17	3 Years	35+
66	498697fad1f17	2009-02-02 06:52:51	2009-02-02 06:52:59	Dreadmaul	Australia	Wollongong	80	Male	Night Elf	Priest	Male	17	3 Years	35+
67	498699a058c8	2009-02-02 06:58:40	2009-02-02 07:02:35	Azjol-Nerub	Australia	Perth	80	Female	Human	Paladin	Female	24	3years	~20
68	49869a500045	2009-02-02 07:01:36	2009-02-02 07:03:02	darrowmere	USA	Staten Island	80	Male	Blood El	Warlock	Male	14	3 years	8
69	498699f0d0064	2009-02-02 07:00:00	2009-02-02 07:03:31	Dalaran	America	Ann Arbor	71	Female	Night Elf	Druid	Female	22	4 years? (I	perhaps 20
70	498699f0d0064	2009-02-02 07:03:31	2009-02-02 07:03:41	Dalaran	America	Ann Arbor	71	Female	Night Elf	Druid	Female	22	4 years? (I	perhaps 20
71	49869a6a1431	2009-02-02 07:02:02	2009-02-02 07:03:48	Area 52	United Sta	Oklahoma City	80	Male	Night Elf	Hunter	Female	14	Two years	At least 8 h
72	49869a1870a8	2009-02-02 07:00:40	2009-02-02 07:03:56	Khaz'gc	Australia	Melbourne	80	Male	Undead	Druid	Male	16	4 years	167
73	49869cd8de6f4	2009-02-02 07:12:24	2009-02-02 07:14:17	Rivendare	Singapore	Singapore	79	Female	Tauren	Druid	Female	26	4 months	30 - 40
74	49869cd8de6f4	2009-02-02 07:14:17	2009-02-02 07:14:34	Rivendare	Singapore	Singapore	79	Female	Tauren	Druid	Female	26	4 months	30 - 40
75	49869ce6ed08	2009-02-02 07:12:38	2009-02-02 07:19:50	Windrunner	United Sta	Rialto	73	Female	Night Elf	Hunter	Male	20	3.5 years	Varies ~15
76	4986ac64e25c	2009-02-02 08:18:44	2009-02-02 08:19:59	Garrosh	The Unite	Unclear	80	Male	Night Elf	Priest	Male	15	3 years	4
77	4986ac64e25c	2009-02-02 08:19:59	2009-02-02 08:20:08	Garrosh	The Unite	Unclear	80	Male	Night Elf	Priest	Male	15	3 years	4
78	4986ac984e59	2009-02-02 08:19:36	2009-02-02 08:21:49	Blackrock	United Sta	Chicago	80	Male	Orc	Shaman	Male	19	A year	20-60
79	4986ade22bcb	2009-02-02 08:25:06	2009-02-02 08:29:07	Lethon	United Sta	Citrus Heights	80	Male	Orc	Hunter	Male	23	Since Dece	about 20
80	4986cf79ec995	2009-02-02 10:48:26	2009-02-02 10:50:37	Thunderlord	United Sta	Laredo	40	Female	Night Elf	Mage	Female	18	On and off	during sch
81	4986e00eac82	2009-02-02 11:59:10	2009-02-02 11:59:57	Alleria	United Sta	Seattle	75	Male	Draenei	Paladin	Male	20	<1 year	>10

Student	40	Wrath of the Lich King	N	N		2009-02-02 04:11:59
Student	40	Wrath of the Lich King	Y	Y		2009-02-02 04:11:59
marketing	40	Wrath of the Lich King	Y	Y		2009-02-02 04:56:20
College Student	40	Wrath of the Lich King	Y	Y	I wish I could	2009-02-02 04:59:13
Student	35	Wrath of the Lich King	Y	Y		2009-02-02 05:01:20
Parts Interpreter	45	Wrath of the Lich King	N	N	Good luck wit	2009-02-02 05:04:19
Parts Interpreter	45	Wrath of the Lich King	Y	Y	Good luck wit	2009-02-02 05:04:19
Private Investigator	40	Wrath of the Lich King	Y	Y		2009-02-02 05:17:54
Unemployed		Wrath of the Lich King	N	N		2009-02-02 05:26:35
Unemployed		Wrath of the Lich King	Y	Y		2009-02-02 05:26:35
Student	60	Wrath of the Lich King	N	N		2009-02-02 05:36:16
Student	60	Wrath of the Lich King	Y	Y		2009-02-02 05:36:16
fulltime managerial	40+	Wrath of the Lich King	Y	Y		2009-02-02 05:47:04
Student	30 hours	Wrath of the Lich King	Y	Y		2009-02-02 05:48:19
Student	30 Exactly Lol	Wrath of the Lich King	N	N		2009-02-02 05:53:43
Student	30 Exactly Lol	Wrath of the Lich King	Y	Y		2009-02-02 05:53:43
student	25	Wrath of the Lich King	N	N		2009-02-02 06:29:10
student	25	Wrath of the Lich King	N	N		2009-02-02 06:29:10
student	25	Wrath of the Lich King	Y	Y		2009-02-02 06:29:10
student: comm/math	40	Wrath of the Lich King	Y	Y	If the forums	2009-02-02 06:35:59
Student		Not-Specified	N	N		2009-02-02 06:51:38
Student	40-50	Wrath of the Lich King	N	N		2009-02-02 06:51:38
Student	40-50	Wrath of the Lich King	N	N		2009-02-02 06:51:38
Student	40-50	Wrath of the Lich King	N	N		2009-02-02 06:51:38
Student	40-50	Wrath of the Lich King	N	N		2009-02-02 06:51:38
Student	40-50	Wrath of the Lich King	Y	Y		2009-02-02 06:51:38
Accountant	36	Wrath of the Lich King	Y	Y		2009-02-02 06:58:40
student	40	Wrath of the Lich King	Y	Y		2009-02-02 07:01:36
none atm	NA	Wrath of the Lich King	N	N	Very interestii	2009-02-02 07:00:00
none atm	NA	Wrath of the Lich King	Y	Y	Very interestii	2009-02-02 07:00:00
ırs a day...		Wrath of the Lich King	Y	Y	Homeschoole	2009-02-02 07:02:02
I work in the home of the	1	Wrath of the Lich King	Y	N	If you want di	2009-02-02 07:00:40
Multimedia Producer	8 - 9	Wrath of the Lich King	N	N		2009-02-02 07:12:24
Multimedia Producer	8 - 9	Wrath of the Lich King	Y	Y		2009-02-02 07:12:24
Student	~45	Wrath of the Lich King	Y	Y	Regarding the	2009-02-02 07:12:38
Student	~50	Wrath of the Lich King	N	N		2009-02-02 08:18:44
Student	~50	Wrath of the Lich King	Y	Y		2009-02-02 08:18:44
College student, YMCA	30	Wrath of the Lich King	Y	Y		2009-02-02 08:19:36
Electrician	40	Wrath of the Lich King	Y	Y		2009-02-02 08:25:06
Student	16	Burning Crusade	Y	N	WoW is ftw =)	2009-02-02 10:48:26
College student	All of them	Wrath of the Lich King	N	N		2009-02-02 11:59:10

82	4986e00eac82!	2009-02-02 11:59:57	2009-02-02 12:00:05	Alleria	United Sta	Seattle	75	Male	Draenei	Paladin	Male	20	<1 year	>10
83	4986e105f1e70	2009-02-02 12:03:17	2009-02-02 12:06:18	Llane	Ireland	Dublin	80	Male	Night Elf	Priest	Male	23	2 years	10
84	4986e2dfd303f	2009-02-02 12:11:11	2009-02-02 12:12:49	Kael'tha	United Sta	Covina	80	Male	Tauren	Druid	Male	22	3 1/2 years	20
85	4986e34f8fb25!	2009-02-02 12:13:03	2009-02-02 12:15:20	The Venture (	United States		80	Female	Orc	Rogue	Female	26	About 2 yea	20ish
86	4986ee3241c1l	2009-02-02 12:59:30	2009-02-02 13:01:12	Eonar	Australia	Brisbane	80	Male	Troll	Hunter	Male	15	3yrs	15
87	498732227081(	2009-02-02 17:49:22	2009-02-02 17:50:47	Ghostlands	New Zeala	Christchurch	29	Female	Blood Elf	Rogue	Female	15	3 years	20ish
88	49874174626b	2009-02-02 18:54:44	2009-02-02 18:56:04	Scarlet Crusa	US	Seattle	80	Male	Orc	Shaman	Male	32	5 years	25
89	498772e75358	2009-02-02 22:25:43	2009-02-02 22:26:45	Illidan	USA	Lansing	80	Male	Tauren	Druid	Male	26	4	20
90	498794fd2d49e	2009-02-03 00:51:09	2009-02-03 00:52:10	Moonrunner	Canada	Vancouver	80	Male	Human	Mage	Male	15	2 years	10
91	4987954279ca8	2009-02-03 00:52:18	2009-02-03 00:53:34	Bleeding Holl	USA	Monroe	80	Male	Orc	Death Kn	Male	16	3 years	20-30 hour
92	498795312b8b	2009-02-03 00:52:01	2009-02-03 00:53:36	Khaz Modan	USA	San Diego	80	Male	Undead	Death Kn	Male	56	3 yrs	20
93	498795549240	2009-02-03 00:52:36	2009-02-03 00:53:54	FireTree	Canada	Africa	40000	Female	Blood Elf	Paladin	Female	-19	-80	400
94	498795329f9df!	2009-02-03 00:52:02	2009-02-03 00:53:55	turylon	usA	baltimore	80	Female	Human	Paladin	Not Speci	32	3 years	35
95	4987957c49a6	2009-02-03 00:53:16	2009-02-03 00:54:49	Draka	USA	Norfolk	76	Female	Night Elf	Warrior	Female	20	1 year	40 hours
96	498799d1631e!	2009-02-03 01:11:45	2009-02-03 01:11:49					Not Specif	Not-Spe	Not-Spec	Not Specified			
97	498799d1631e!	2009-02-03 01:11:49	2009-02-03 01:13:14	Coilfang	USA	Laguna Niguel	24	Female	Troll	Shaman	Male	22	10 days	40
98	4987b0c30b678	2009-02-03 02:49:39	2009-02-03 02:51:19	Drak'Th	USA	Sacramento	80	Male	Blood Elf	Mage	Female	30	a little over	40+
99	4987b79d967c	2009-02-03 03:18:53	2009-02-03 03:20:25	Wildhammer	U.S	Tallahassee	80	Male	Troll	Shaman	Male		3 years	10
100	4987ce34ed248	2009-02-03 04:55:17	2009-02-03 04:57:10	Frostmourne	Australia	Melbourne	63	Male	Night Elf	Rogue	Male	23	1 year	30
101	4987fc19298a3	2009-02-03 08:11:05	2009-02-03 08:13:00	Aman'T	New Zeala	Timaru	80	Male	Night Elf	Druid	Male	15	~1 year	20
102	4987fd5a54d4c	2009-02-03 08:16:26	2009-02-03 08:18:20	Shadowsong	United Sta	Phoenix	80	Male	Troll	Shaman	Male	17	2 Years	20-36 Hour
103	4987ffa97b18a!	2009-02-03 08:26:17	2009-02-03 08:28:09	antonidas	usa	sparta	80	Female	Tauren	Hunter	Female	24	3 years	25
104	4987ffa97b18a!	2009-02-03 08:28:09	2009-02-03 08:28:35	antonidas	usa	sparta	80	Female	Tauren	Hunter	Female	24	3 years	25
105	4987ffa97b18a!	2009-02-03 08:28:35	2009-02-03 08:28:50	antonidas	usa	sparta	80	Female	Tauren	Hunter	Female	24	3 years	25
106	498800011ded8	2009-02-03 08:27:45	2009-02-03 08:29:55	Dath'Re	US, currently in Japan		80	Male	Blood Elf	Paladin	Male	27	Less than 6	2-4, I'
107	4988026e9a39	2009-02-03 08:38:06	2009-02-03 08:39:22	Shu'Hal	United Sta	Indianapolis	80	Female	Night Elf	Druid	Female	35	1 year	50 or so
108	4988033a0cf51	2009-02-03 08:41:30	2009-02-03 08:43:19	Khaz'Go	New Zeala	Wellington	80	Male	Draenei	Paladin	Male	15	About 2 yea	From 15-25
109	49880868b241(	2009-02-03 09:03:37	2009-02-03 09:05:30	VashJ	USA	Las Vegas	80	Female	Troll	Death Kn	Male	17	1.5 years	24+
110	498808c4b55fa	2009-02-03 09:05:08	2009-02-03 09:06:47	skywall	australia	sydney	79	Male	Human	Warlock	Male	42	3 yrs	20
111	498808c7b0f12	2009-02-03 09:05:11	2009-02-03 09:06:53	Scarlet Crusa	USA	Albany	80	Female	Human	Rogue	Female	32	4 years	42
112	49881b120a798	2009-02-03 10:23:14	2009-02-03 10:38:32	saurfang	australia	sydney	80	Male	Night Elf	Not-Spec	Male	33	4yrs	50
113	49889e19df3cb	2009-02-03 19:42:18	2009-02-03 19:44:51	Suramon?	Australia	Melbourne	48	Male	Orc	Rogue	Male	28	12mths	0
114	4988e4eb50aa!	2009-02-04 00:44:27	2009-02-04 00:46:35	Moonglade	Holland	Tilburg	80	Male	Troll	Rogue	Male	18	About 4 yea	About 35?
115	4988ea2ee909(	2009-02-04 01:06:54	2009-02-04 01:10:39	Dentarg	United Sta	Gaithersburg (\	70	Male	Night Elf	Hunter	Male	16 (a	1 1/2 years	no more th
116	4988edb59fef5	2009-02-04 01:21:57	2009-02-04 01:23:14	moonrunner	USA	Iowa City	80	Male	Undead	Mage	Male	32	2 years	10-15
117	4988f515d639e	2009-02-04 01:53:25	2009-02-04 01:55:22	spinebreaker	US	New york	80	Male	Night Elf	Warrior	Not Speci	24	3 years	12
118	498a4e935b44	2009-02-05 02:24:23	2009-02-05 02:27:31	Bladefist	USA	Los Angees	22	Male	Human	Paladin	Not Speci	37	2months	10
119	498a4e935b44	2009-02-05 02:27:31	2009-02-05 02:27:55	Bladefist	USA	Los Angees	22	Male	Human	Paladin	Not Speci	37	2months	10
120	498a4e935b44	2009-02-05 02:27:55	2009-02-05 02:28:17	Bladefist	USA	Los Angees	22	Male	Human	Paladin	Not Speci	37	2months	10
121	498a89615779	2009-02-05 06:35:47	2009-02-05 06:38:25	Aegent Dawn	USA	Spokane WA	25	Female	Tauren	Hunter	Female	18	1.5 years	40+
122	498a89615779	2009-02-05 06:38:25	2009-02-05 06:38:49	Aegent Dawn	USA	Spokane WA	25	Female	Tauren	Hunter	Female	18	1.5 years	40+

College student	All of them	Wrath of the Lich King	Y	Y		2009-02-02 11:59:10
IT technicial support	40	Wrath of the Lich King	Y	Y	Good survey	2009-02-02 12:03:17
Customer Service Rep	40	Wrath of the Lich King	Y	Y		2009-02-02 12:11:11
Typist	Not sure	Wrath of the Lich King	Y	Y		2009-02-02 12:13:03
stuedent	35hrs	Wrath of the Lich King	Y	Y		2009-02-02 12:59:30
high school student	30	Wrath of the Lich King	Y	Y		2009-02-02 17:49:22
database developer	50	Wrath of the Lich King	Y	Y		2009-02-02 18:54:44
student	40	Wrath of the Lich King	Y	Y		2009-02-02 22:25:43
student	35	Wrath of the Lich King	Y	Y		2009-02-03 00:51:09
student	7-8 hours	Wrath of the Lich King	Y	Y		2009-02-03 00:52:18
Retired Military		Wrath of the Lich King	Y	Y		2009-02-03 00:52:01
Hooker	i play and do m	Wrath of the Lich King	N	N	LOL	2009-02-03 00:52:36
radiology	36	Wrath of the Lich King	Y	Y		2009-02-03 00:52:02
Student and computer a	40 hours	Wrath of the Lich King	Y	Y		2009-02-03 00:53:16
		Not-Specified	N	N		2009-02-03 01:11:45
student	30	Classic	Y	Y	I've only	2009-02-03 01:11:45
technical writer	40+	Wrath of the Lich King	Y	Y	Planning on p	2009-02-03 02:49:39
student	40	Wrath of the Lich King	Y	N		2009-02-03 03:18:53
Student	20	Burning Crusade	Y	Y		2009-02-03 04:55:17
Student, work in toolstor	10	Wrath of the Lich King	Y	Y	Buff feral drui	2009-02-03 08:11:05
Student	42 Hours	Wrath of the Lich King	Y	Y		2009-02-03 08:16:26
student	15	Wrath of the Lich King	N	N		2009-02-03 08:26:17
student	15	Wrath of the Lich King	N	N		2009-02-03 08:26:17
student	15	Wrath of the Lich King	Y	Y		2009-02-03 08:26:17
High school teacher	40+	Wrath of the Lich King	Y	Y	I don't lil	2009-02-03 08:27:45
Homemaker	20 or so	Wrath of the Lich King	Y	Y		2009-02-03 08:38:06
Student at highschool.	Uh highschool?	Wrath of the Lich King	Y	Y	Nice survey =	2009-02-03 08:41:30
Student	10.5 Hours	Wrath of the Lich King	Y	Y		2009-02-03 09:03:37
chef	30	Wrath of the Lich King	Y	Y		2009-02-03 09:05:08
Disabled	0	Wrath of the Lich King	Y	Y		2009-02-03 09:05:11
techie	30	Wrath of the Lich King	Y	Y	I take occasic	2009-02-03 10:23:14
Accountant	50	Classic	Y	Y	Don't pl	2009-02-03 19:42:18
Finishing up 'colleg	Close to 35 hou	Wrath of the Lich King	Y	Y		2009-02-04 00:44:27
student	40	Wrath of the Lich King	Y	N	Even though	2009-02-04 01:06:54
Neonatal Fellow	60-80 hours	Wrath of the Lich King	Y	Y		2009-02-04 01:21:57
student	35	Wrath of the Lich King	Y	Y	Wtf is this	2009-02-04 01:53:25
theatre	38	Burning Crusade	N	N	I have about	2009-02-05 02:24:23
theatre	38	Burning Crusade	N	N	I have about	2009-02-05 02:24:23
theatre	38	Burning Crusade	N	N	I have about	2009-02-05 02:24:23
Mother	....all?	Wrath of the Lich King	N	N	I love this gar	2009-02-05 06:35:47
Mother	....all?	Wrath of the Lich King	N	N	I love this gar	2009-02-05 06:35:47

123	498a89615779	2009-02-05 06:38:49	2009-02-05 06:39:04	Aegent Dawn	USA	Spokane WA	25	Female	Tauren	Hunter	Female	18	1.5 years	40+
124	498b7c4e3f916	2009-02-05 23:54:54	2009-02-05 23:56:43	nordrassil	USA	Minneapolis	80	Female	Night Elf	Hunter	Female	38	2 years	12
125	498bcf6b69a4a	2009-02-06 05:49:31	2009-02-06 05:52:11	dentarg	united stat	shippemsburg	72	Not Specif	Night Elf	Hunter	Not Speci	20	3 months	7o hours
126	498c9c549a66f	2009-02-06 20:23:36	2009-02-06 20:23:48					Not Specif	Not-Spe	Not-Spec	Not Specified			
127	498cad675f571	2009-02-06 21:36:39	2009-02-06 21:40:03	Silverhand	USA	New Meadows	80	Not Specif	Human	Rogue	Not Speci	34	5	15
128	498dfc6bc1c6e	2009-02-07 21:23:43	2009-02-07 21:26:03	Auchindoun (	Finland		80	Female	Draenei	Warrior	Not Speci	35	2 years	21
129	498dfc6bc1c6e	2009-02-07 21:26:03	2009-02-07 21:26:25	Auchindoun (	Finland		80	Female	Draenei	Warrior	Not Speci	35	2 years	21
130	498dfc6bc1c6e	2009-02-07 21:26:25	2009-02-07 21:26:48	Auchindoun (	Finland		80	Female	Draenei	Warrior	Not Speci	35	2 years	21
131	498dfc6bc1c6e	2009-02-07 21:26:48	2009-02-07 21:27:14	Auchindoun (	Finland		80	Female	Draenei	Warrior	Not Speci	35	2 years	21
132	498dfc6bc1c6e	2009-02-07 21:27:14	2009-02-07 21:28:01	Auchindoun (	Finland	Ã¥bo	80	Female	Draenei	Warrior	Female	35	2 years	21
133	498f9270d0b28	2009-02-09 02:18:25	2009-02-09 02:23:54	Jaedenar	USA	nope	75	Female	Orc	Death Kn	Male	39	1.5 yrs	15
134	49909548a6a1	2009-02-09 20:41:10	2009-02-09 20:42:48	Ysera US	United Sta	Hendersonville	72 as	Male	Tauren	Warrior	Male	14	Four Years	Atleast 20.
135	49909548a6a1	2009-02-09 20:42:48	2009-02-09 20:43:04	Ysera US	United Sta	Hendersonville	72 as	Male	Tauren	Warrior	Male	14	Four Years	Atleast 20.
136	49909548a6a1	2009-02-09 20:43:04	2009-02-09 20:43:13	Ysera US	United Sta	Hendersonville	72 as	Male	Tauren	Warrior	Male	14	Four Years	Atleast 20.
137	49909548a6a1	2009-02-09 20:43:13	2009-02-09 20:43:25	Ysera US	United Sta	Hendersonville	72 as	Male	Tauren	Warrior	Male	14	Four Years	Atleast 20.
138	4990ae578374	2009-02-09 22:29:43	2009-02-09 22:31:01	Onyxia	USA	Chicago	80	Male	Troll	Shaman	Male	19	Two years	10
139	4997456d4496	2009-02-14 22:27:57	2009-02-14 22:28:59	Destromath	United Sta	West Friendshi	80	Male	Orc	Shaman	Male	18	2 years	35-30
140	4997456d4496	2009-02-14 22:28:59	2009-02-14 22:29:04	Destromath	United Sta	West Friendshi	80	Male	Orc	Shaman	Male	18	2 years	35-30
141	499745c50ca3	2009-02-14 22:29:25	2009-02-14 22:32:18	Garithos	USA	Addison	21	Male	Human	Paladin	Male	14	2 years	30?
142	499745c50ca3	2009-02-14 22:32:18	2009-02-14 22:32:32	Garithos	USA	Addison	21	Male	Human	Paladin	Male	14	2 years	30?
143	4997465f9b0d7	2009-02-14 22:31:59	2009-02-14 22:33:14	Blackhand	Canada	Vancouver	80	Male	Gnome	Warrior	Male	23	4 years	40-50 hour
144	4997465f9b0d7	2009-02-14 22:33:14	2009-02-14 22:33:21	Blackhand	Canada	Vancouver	80	Male	Gnome	Warrior	Male	23	4 years	40-50 hour
145	4997461746a6	2009-02-14 22:30:47	2009-02-14 22:33:38	steamwheedl	usa	shreveport	80	Female	Human	Warlock	Female	32	since relea	5
146	4997461746a6	2009-02-14 22:33:38	2009-02-14 22:34:07	steamwheedl	usa	shreveport	80	Female	Human	Warlock	Female	32	since relea	5
147	4997470ae597	2009-02-14 22:34:50	2009-02-14 22:36:13	Quel'do	United Sta	Yorba Linda	80	Male	Blood El	Paladin	Male	20	1 year	20 hours
148	4997471c47e9	2009-02-14 22:35:08	2009-02-14 22:36:41	Ner'Zhu	USA	Erie, Pennyslv	80	Female	Draenei	Priest	Female	19	4 years	40+
149	49974731e5d7	2009-02-14 22:35:29	2009-02-14 22:37:30	Earthen Ring	USA	Eugene, OR	80	Female	Troll	Priest	Female	31	4 Years	40
150	499747d84cd7	2009-02-14 22:38:16	2009-02-14 22:38:21					Not Specif	Not-Spe	Not-Spec	Not Specified			
151	499747c4a75d	2009-02-14 22:37:56	2009-02-14 22:38:49	Ravenholdt	United Sta	Lakeville	80	Male	Gnome	Mage	Male	15	2 Years	50+
152	4997472933dfe	2009-02-14 22:35:21	2009-02-14 22:39:06	shattered har	usa	baltimore	80	Male	Troll	Shaman	Male	23	since beta	30
153	49974609b5e4	2009-02-14 22:30:33	2009-02-14 22:39:24	Bonechewer	USA	Gainesville	72	Female	Night Elf	Druid	Male	18	5 years	10-14
154	49974918d45el	2009-02-14 22:43:36	2009-02-14 22:44:41	Alterac Moun	United Sta	Lowell, MA	80	Male	Night Elf	Druid	Male	22	2 years	10-20
155	49974932adbd	2009-02-14 22:44:02	2009-02-14 22:45:21	Kargath	Canada	Toronto	80	Male	Blood El	Mage	Male	16	3 years, wi	21 hours
156	499749803a48	2009-02-14 22:45:20	2009-02-14 22:46:31	Boulderfist	USA	Dallas	75	Male	Undead	Priest	Male	15	1-2 years	Not sure
157	4997490fa3091	2009-02-14 22:45:35	2009-02-14 22:46:50	Laughing Sku	USA	Igncaio	80	Male	Tauren	Death Kn	Male	23	6 months	10-12
158	499749b087a1	2009-02-14 22:46:08	2009-02-14 22:47:31	Gnomeregan	USA	Atlanta	80	Male	Human	Mage	Male	16	1 year 6 m	15-25
159	499749d59fe6c	2009-02-14 22:46:45	2009-02-14 22:50:00	Area 52/Burn	USA	Honolulu	80	Female	Tauren	Druid	Female	23	3.5 years	15-20
160	49974a5a157e	2009-02-14 22:48:58	2009-02-14 22:50:36	Duskwood	United Sta	Scranton	80	Male	Orc	Death Kn	Male	15	3 years	~15
161	49974a5a157e	2009-02-14 22:50:36	2009-02-14 22:50:44	Duskwood	United Sta	Scranton	80	Male	Orc	Death Kn	Male	15	3 years	~15
162	49974a475559	2009-02-14 22:48:39	2009-02-14 22:51:32	Dunemaul	United Sta	Portland	70	Female	Draenei	Shaman	Female	23	Since relea	~20
163	49974ae83c4e	2009-02-14 22:51:20	2009-02-14 22:53:19	Cenarion Circ	USA	Ketchikan	80	Female	Night Elf	Death Kn	Male	18	2005	35

Mother	....all?	Wrath of the Lich King	N	N	I love this gar	2009-02-05 06:35:47
daycare teacher	36	Wrath of the Lich King	Y	Y		2009-02-05 23:54:54
ssi	none	Wrath of the Lich King	Y	Y		2009-02-06 05:49:31
		Not-Specified	Y	N		2009-02-06 20:23:36
it	40	Wrath of the Lich King	Y	Y		2009-02-06 21:36:39
nurse	40	Wrath of the Lich King	N	N		2009-02-07 21:23:43
nurse	40	Wrath of the Lich King	N	N		2009-02-07 21:23:43
nurse	40	Wrath of the Lich King	N	N		2009-02-07 21:23:43
nurse	40	Wrath of the Lich King	N	N		2009-02-07 21:23:43
nurse	40	Wrath of the Lich King	N	N		2009-02-07 21:23:43
office amin	55	Wrath of the Lich King	Y	Y	boo	2009-02-09 02:18:25
Student	Five	Wrath of the Lich King	N	N		2009-02-09 20:41:10
Student	Five	Wrath of the Lich King	N	N		2009-02-09 20:41:10
Student	Five	Wrath of the Lich King	N	N		2009-02-09 20:41:10
Student	Five	Wrath of the Lich King	N	N		2009-02-09 20:41:10
Student	15	Wrath of the Lich King	Y	Y		2009-02-09 22:29:43
Senior in high school	50	Wrath of the Lich King	N	N		2009-02-14 22:27:57
Senior in high school	50	Wrath of the Lich King	Y	Y		2009-02-14 22:27:57
Student	30	Burning Crusade	N	N	I'm not e	2009-02-14 22:29:25
Student	30	Burning Crusade	Y	Y	I\'m not	2009-02-14 22:29:25
student	30	Wrath of the Lich King	N	N		2009-02-14 22:31:59
student	30	Wrath of the Lich King	Y	Y		2009-02-14 22:31:59
full time Mom	endless	Wrath of the Lich King	N	N		2009-02-14 22:30:47
full time Mom	endless	Wrath of the Lich King	Y	Y		2009-02-14 22:30:47
full time student	25 hours	Wrath of the Lich King	Y	Y		2009-02-14 22:34:50
Student (international bu	many. full time	Wrath of the Lich King	Y	Y	good luck witl	2009-02-14 22:35:08
CNA	30	Wrath of the Lich King	Y	Y		2009-02-14 22:35:29
		Not-Specified	N	N		2009-02-14 22:38:16
Student	30	Wrath of the Lich King	Y	Y		2009-02-14 22:37:56
student/full time tech su	40	Wrath of the Lich King	Y	Y	id like to know	2009-02-14 22:35:21
College Student & Janit	20	Wrath of the Lich King	Y	Y		2009-02-14 22:30:33
Student / Other	30-35	Wrath of the Lich King	Y	Y		2009-02-14 22:43:36
student	25 hours	Wrath of the Lich King	Y	Y	Blah	2009-02-14 22:44:02
Student	Alot	Wrath of the Lich King	Y	Y		2009-02-14 22:45:20
operator in the oil field	60	Wrath of the Lich King	Y	Y		2009-02-14 22:45:35
student	40 hours	Wrath of the Lich King	Y	Y		2009-02-14 22:46:08
Student/Production Coo	School: 30-40+	Wrath of the Lich King	Y	Y	Also just start	2009-02-14 22:46:45
High school student	35	Wrath of the Lich King	N	N		2009-02-14 22:48:58
High school student	35	Wrath of the Lich King	Y	Y		2009-02-14 22:48:58
Student of Criminal Justi	60+	Wrath of the Lich King	Y	Y	Hope that hel	2009-02-14 22:48:39
Student	30	Wrath of the Lich King	Y	Y	loggin my key	2009-02-14 22:51:20

164	49974b3cec55(	2009-02-14 22:52:44	2009-02-14 22:53:54	Frostmane - I	England	Swindon	80	Male	Draenei	Shaman	Male	17	2 Years	15ish
165	49974b1e4aa5:	2009-02-14 22:52:14	2009-02-14 22:53:56	Winterhoof	America F	West Lawn, PA	80	Male	Human	Mage	Male	17	2 Years	7x12=How
166	49974b1e4aa5:	2009-02-14 22:53:56	2009-02-14 22:54:04	Winterhoof	America F	West Lawn, PA	80	Male	Human	Mage	Male	17	2 Years	7x12=How
167	49974ca113f19	2009-02-14 22:58:41	2009-02-14 22:59:36	Uldum	US	Sacramento	80	Male	Tauren	Druid	Male	36	3 Years	15
168	49974d4a2efb7	2009-02-14 23:01:30	2009-02-14 23:05:22	Dethecus	Australia	Melbourne	73	Male	Undead	Warlock	Male	18	2 years	Roughly 2(
169	49974de833b3	2009-02-14 23:04:08	2009-02-14 23:05:56	Sen'jin	U.S	Honolulu	80	Male	Orc	Warlock	Male	13	3 years	35
170	49974eebb562(	2009-02-14 23:08:27	2009-02-14 23:09:14	Terenas	Canada	Toronto	80	Female	Tauren	Druid	Female	19	5 Years	30
171	49974f4f1b5fa4	2009-02-14 23:10:07	2009-02-14 23:11:16	Skullcrusher	USA	Winston-Salem	73	Male	Troll	Mage	Male	16	a year off a	20-30
172	49974f6065f81:	2009-02-14 23:10:24	2009-02-14 23:11:25	[redacted]				Female	Gnome	Priest	Male	[redacted]		
173	49974f3364716	2009-02-14 23:09:39	2009-02-14 23:11:28	Sisters of Elu	Russia	St. Petersburg	80	Male	Blood El	Priest	Male	17	4~ years	~30
174	49974f3e97d2c	2009-02-14 23:09:50	2009-02-14 23:12:25	Zangarmarsh	United Sta	Inver Grove He	80	Male	Not-Spe	Not-Spec	Male	14 ar	almost 5 ye	Varies from
175	49974f41a8a6d	2009-02-14 23:09:53	2009-02-14 23:13:10	Shadow Cou	USA	Toms River	80	Female	Undead	Priest	Female	a lad	4 years	how ever m
176	49974f41a8a6d	2009-02-14 23:13:10	2009-02-14 23:13:23	Shadow Cou	USA	Toms River	80	Female	Undead	Priest	Female	a lad	4 years	how ever m
177	49975015cdf50	2009-02-14 23:13:25	2009-02-14 23:14:52	Vashj	United Sta	Cecil	39	Male	Undead	Warrior	Male	16	3 years	7
178	49974b7ce3da	2009-02-14 22:54:41	2009-02-14 23:15:11	Archimonde	USA	Pittsburgh	80	Female	Undead	Warlock	Female	17	3 1/2 years	20-30
179	4997502a2947	2009-02-14 23:13:46	2009-02-14 23:15:13	Korgath	United Sta	St. Cloud	80	Male	Night Elf	Warrior	Male	20	6 months	5-10
180	4997506a1aa5(	2009-02-14 23:14:50	2009-02-14 23:17:12	Stormrage	Canada	Mississauga	61	Female	Human	Mage	Male	25	8 months	2 hours
181	499750a2baec(	2009-02-14 23:15:46	2009-02-14 23:18:28	Zul'jin	USA	Veneta	51	Female	Blood El	Hunter	Female	17	7 months	5
182	499751282df36	2009-02-14 23:18:00	2009-02-14 23:19:08	Anetheron	us	Ann Arbor	80	Male	Undead	Mage	Male	23	3 years	15
183	4997511e5b84(	2009-02-14 23:17:50	2009-02-14 23:19:23	Spirestone	South Afri	Pretoria	80	Male	Blood El	Death Kn	Male	18	3 years	3
184	4997514aa1b2	2009-02-14 23:18:34	2009-02-14 23:20:07	Barthilas	Australia	Ingham	80	Male	Human	Paladin	Male	15	3 years	too many
185	499751edd4e2	2009-02-14 23:21:17	2009-02-14 23:22:35	ysera	US	Colorado Sprin	80	Male	Undead	Warlock	Male	34	2 years	16+
186	499751edd4e2	2009-02-14 23:22:35	2009-02-14 23:22:48	ysera	US	Colorado Sprin	80	Male	Undead	Warlock	Male	34	2 years	16+
187	4997521eb81e	2009-02-14 23:22:06	2009-02-14 23:24:06	Malygos	United Sta	Laurinburg	72	Male	Human	Mage	Male	24	4 Years	35 or more
188	4997520215b2	2009-02-14 23:21:38	2009-02-14 23:24:33	The Venture (	Canada	Kamloops	80	Male	Not-Spe	Not-Spec	Male	19	2 years	20
189	4997520215b2	2009-02-14 23:24:33	2009-02-14 23:25:23	The Venture (	Canada	Kamloops	80	Male	Not-Spe	Not-Spec	Male	19	2 years	20
190	499752a04299(	2009-02-14 23:24:16	2009-02-14 23:25:32	Burning Blad	USA	Granger	80	Male	Gnome	Mage	Male	15	5 years	10-15
191	499752e6c7bff	2009-02-14 23:25:26	2009-02-14 23:26:43	Alterac moun	US	Melrose	80	Male	Tauren	Death Kn	Not Speci	14	2years	21
192	499753436d24	2009-02-14 23:26:59	2009-02-14 23:28:00	Thaurissan	Australia	Launceston	80	Female	Troll	Warrior	Female	17	since 2004	not sure. 2
193	4997533e6724	2009-02-14 23:26:54	2009-02-14 23:29:21	Bronzebeard	United Sta	Columbia	80	Female	Night Elf	Druid	Female	15	3 years	54
194	499753ee385c	2009-02-14 23:29:50	2009-02-14 23:30:53	Echo Isles	United Sta	Las Vegas	80	Male	Gnome	Druid	Male	16	1yr exactly	prob like 2(
195	499753ee385c	2009-02-14 23:30:53	2009-02-14 23:31:06	Echo Isles	United Sta	Las Vegas	80	Male	Gnome	Druid	Male	16	1yr exactly	prob like 2(
196	499754a1429c	2009-02-14 23:32:49	2009-02-14 23:34:53	Dawnbringer	Canada	Calgary	80	Male	Orc	Warrior	Male	15	year and a	too many, r
197	499754face8eb	2009-02-14 23:34:18	2009-02-14 23:35:53	Arthas	Canada	Calgary	80	Male	Night Elf	Druid	Female	19	3 years	25+
198	499755734f856	2009-02-14 23:36:19	2009-02-14 23:37:45	Azuremyst	United Sta	Columbus	30	Male	Human	Mage	Male	19	3 months	20+
199	4997558b0611(	2009-02-14 23:36:43	2009-02-14 23:37:59	Duskwood	America	Pekin	80	Female	Human	Death Kn	Male	15	4 years, 1 r	20
200	499755e26e4f	2009-02-14 23:38:10	2009-02-14 23:39:29	Bladefist	Israel	Tel Aviv	50	Not Specif	Blood El	Rogue	Male	20	A year, on	8
201	499755e00500	2009-02-14 23:38:08	2009-02-14 23:39:34	Arygos	USA	Boise	80	Female	Human	Rogue	Female	47	3.5 years	20-25
202	4997560bbc97	2009-02-14 23:38:51	2009-02-14 23:40:08	Exodar	US	manchester	80	Not Specif	Draenei	Paladin	Male	20	2 years	10-18
203	499757282754	2009-02-14 23:43:36	2009-02-14 23:44:58	draka	america	twin falls idaho	71	Male	Dwarf	Paladin	Male	26	4	8
204	4997573a8ed8	2009-02-14 23:43:54	2009-02-14 23:45:35	spinebreaker	united stat	st. paul	80	Male	Undead	Warrior	Male	25	about a yea	40+

College Student	10	Wrath of the Lich King	Y	Y	None :)	2009-02-14 22:52:44
Student	1-2, home scho	Wrath of the Lich King	N	N	[redacted]	2009-02-14 22:52:14
Student	1-2, home scho	Wrath of the Lich King	Y	Y		2009-02-14 22:52:14
Network Engineer	50	Wrath of the Lich King	Y	Y		2009-02-14 22:58:41
Plumber's Apprenti	40	Wrath of the Lich King	Y	Y		2009-02-14 23:01:30
Middle School	25	Wrath of the Lich King	Y	Y	Prepare to be	2009-02-14 23:04:08
Student	30	Wrath of the Lich King	Y	Y		2009-02-14 23:08:27
student	50-60	Wrath of the Lich King	Y	Y		2009-02-14 23:10:07
[redacted]		Classic	Y	N	[redacted]	2009-02-14 23:10:24
Student	60	Wrath of the Lich King	Y	Y	Ð–Ð¸Ð²Ð¸‚Ð	2009-02-14 23:09:39
Student	50-60	Wrath of the Lich King	Y	N	Best game ev	2009-02-14 23:09:50
nany i have free		Wrath of the Lich King	N	N		2009-02-14 23:09:53
nany i have free		Wrath of the Lich King	Y	N		2009-02-14 23:09:53
student	high school	Wrath of the Lich King	Y	Y	i liek mudkipz	2009-02-14 23:13:25
High school senoir	40	Wrath of the Lich King	Y	Y		2009-02-14 22:54:41
Full-time Student	30-35 hours	Wrath of the Lich King	Y	Y	Good luck wit	2009-02-14 23:13:46
Software engineer / Gan	8-12 hours	Wrath of the Lich King	Y	Y	N/A	2009-02-14 23:14:50
student	7	Burning Crusade	Y	Y		2009-02-14 23:15:46
student	40	Wrath of the Lich King	Y	Y		2009-02-14 23:18:00
Student	30	Wrath of the Lich King	Y	Y	I dislike aussi	2009-02-14 23:17:50
student	30+ hours	Wrath of the Lich King	Y	Y		2009-02-14 23:18:34
Consultant	40+	Wrath of the Lich King	N	N	Don't lo	2009-02-14 23:21:17
Consultant	40+	Wrath of the Lich King	Y	Y	Don\'t lo	2009-02-14 23:21:17
Hotel Manager	40+	Wrath of the Lich King	Y	Y		2009-02-14 23:22:06
student/sales associate	24	Wrath of the Lich King	N	N		2009-02-14 23:21:38
student/sales associate	24	Wrath of the Lich King	Y	Y		2009-02-14 23:21:38
student	7 hrs/day 5 day	Wrath of the Lich King	Y	Y	i want to do th	2009-02-14 23:24:16
student	35	Wrath of the Lich King	Y	Y		2009-02-14 23:25:26
student	40+	Wrath of the Lich King	Y	Y		2009-02-14 23:26:59
Student	40	Wrath of the Lich King	Y	Y		2009-02-14 23:26:54
student	20 hours	Wrath of the Lich King	N	N		2009-02-14 23:29:50
student	20 hours	Wrath of the Lich King	Y	N		2009-02-14 23:29:50
High school student	30	Wrath of the Lich King	Y	Y	[redacted]	2009-02-14 23:32:49
Full time student	30+	Wrath of the Lich King	Y	Y		2009-02-14 23:34:18
Student	40+	Wrath of the Lich King	Y	Y		2009-02-14 23:36:19
student	35	Wrath of the Lich King	Y	Y		2009-02-14 23:36:43
Logistics and Economics	40	Burning Crusade	Y	Y		2009-02-14 23:38:10
Chef	40+	Wrath of the Lich King	Y	Y	Love the gam	2009-02-14 23:38:08
student	35	Wrath of the Lich King	Y	Y		2009-02-14 23:38:51
tech support	40	Wrath of the Lich King	Y	Y		2009-02-14 23:43:36
retail	30-40	Wrath of the Lich King	N	N		2009-02-14 23:43:54

205	4997573a8ed8:	2009-02-14 23:45:35	2009-02-14 23:45:44	spinebreaker	united stat	st. paul	80	Male	Undead	Warrior	Male	25	about a yea	40+
206	4997573a8ed8:	2009-02-14 23:45:44	2009-02-14 23:45:57	spinebreaker	united stat	st. paul	80	Male	Undead	Warrior	Male	25	about a yea	40+
207	4997580478f62	2009-02-14 23:48:58	2009-02-14 23:50:00	Undermine	Scotland	St. Andrews	80	Male	Draenei	Death Kn	Male	38	2 Years	40+
208	4997580478f62	2009-02-14 23:50:00	2009-02-14 23:50:16	Undermine	Scotland	St. Andrews	80	Male	Draenei	Death Kn	Male	38	2 Years	40+
209	499758a93848	2009-02-14 23:50:01	2009-02-14 23:51:18	Kaelthas	Usa	Cincinatti	80	Female	Night Elf	Druid	Male	24	3 years	~20
210	499758a93848	2009-02-14 23:51:18	2009-02-14 23:51:25	Kaelthas	Usa	Cincinatti	80	Female	Night Elf	Druid	Male	24	3 years	~20
211	499758898bb2(	2009-02-14 23:49:29	2009-02-14 23:51:54	Greymane	United Sra	Lake Forest Ca	80	Male	Human	Rogue	Male	15	4 years	14 hours i
212	499758898bb2(	2009-02-14 23:51:54	2009-02-14 23:52:00	Greymane	United Sra	Lake Forest Ca	80	Male	Human	Rogue	Male	15	4 years	14 hours i
213	499758898bb2(	2009-02-14 23:52:00	2009-02-14 23:52:11	Greymane	United Sra	Lake Forest Ca	80	Male	Human	Rogue	Male	15	4 years	14 hours i
214	49975903b44e	2009-02-14 23:51:31	2009-02-14 23:52:27	Alterac Moun	Iran	Tehran	1	Female	Dwarf	Druid	Female	4	4 years	40+
215	499759336a10(	2009-02-14 23:52:19	2009-02-14 23:53:54	The Forgotter	The Unitec	Godfrey	80	Male	Human	Paladin	Male	19	Three year	MANY
216	49975a5e7431	2009-02-14 23:57:18	2009-02-14 23:59:11	thunderlord	America	Ventura	74	Male	Human	Paladin	Male	14	3 years	14 hours
217	49975a6b89b2l	2009-02-14 23:57:31	2009-02-14 23:59:46	Frostmane	Australia	Brisbane	60	Male	Orc	Warrior	Male	15	2 years	3
218	49975ab82016	2009-02-14 23:58:48	2009-02-15 00:00:09	Echo Isles	United Sta	Chicago	75	Male	Human	Death Kn	Male	14	half a year	10-20
219	49975ac95e0d	2009-02-14 23:59:05	2009-02-15 00:00:45	BlackWing La	Canada	Midland	69	Male	Tauren	Death Kn	Male	17	Since BC	15-25
220	49975ac1d5ac	2009-02-14 23:58:57	2009-02-15 00:01:29	Gnomeregan	Canada	Trenton	74	Male	Human	Rogue	Male	15	2 years	3
221	49975add9bc6	2009-02-14 23:59:25	2009-02-15 00:01:37	Ravenholdt	USA	Springfield	80	Male	Blood El	Mage	Male	20	since dece	20-40
222	49975b632834l	2009-02-15 00:01:39	2009-02-15 00:01:50	Doomhamme	United States			Not Specif	Not-Spe	Not-Spec	Not Specified			
223	49975b3c3b4b	2009-02-15 00:01:00	2009-02-15 00:02:44	Azjol-Nerub	America	San Francisco	80	Female	Dwarf	Paladin	Male	49	4 years	10-20
224	49975b63d31d	2009-02-15 00:01:39	2009-02-15 00:03:08	Thunderhorn	Australia	Broome	80	Female	Gnome	Mage	Male	15	4 years	Too many.
225	49975b632834l	2009-02-15 00:01:50	2009-02-15 00:03:19	Doomhamme	United Sta	Columbus	70	Male	Dwarf	Priest	Male	18	About 2 1/2	0, I quit
226	49975b63d31d	2009-02-15 00:03:08	2009-02-15 00:03:26	Thunderhorn	Australia	Broome	80	Female	Gnome	Mage	Male	15	4 years	Too many.
227	49975ba598a1	2009-02-15 00:02:45	2009-02-15 00:04:18	Aggramar	USA	Dallas	80	Female	Draenei	Mage	Male	15	Since Rele	About 15-2
228	49975b888cc2	2009-02-15 00:02:16	2009-02-15 00:04:58	Illidan	USA	Monterey	80	Female	Blood El	Mage	Female	42	4yrs	15 to 20
229	49975bf753083	2009-02-15 00:04:07	2009-02-15 00:05:58	Kael'Th	Canada	Kingston	80	Male	Night Elf	Warrior	Male	21	3 years	11 hours
230	49975bf753083	2009-02-15 00:05:58	2009-02-15 00:06:09	Kael'Th	Canada	Kingston	80	Male	Night Elf	Warrior	Male	21	3 years	11 hours
231	49975bf753083	2009-02-15 00:06:09	2009-02-15 00:06:18	Kael'Th	Canada	Kingston	80	Male	Night Elf	Warrior	Male	21	3 years	11 hours
232	49975c2f5df6d	2009-02-15 00:05:03	2009-02-15 00:06:49	Thaurissan	New Zeala	Nelson	73	Male	Tauren	Shaman	Male	18	4 years	0
233	49975cbed1d2	2009-02-15 00:07:26	2009-02-15 00:09:02	Winterhoof	Canada	Edmonton	80	Male	Night Elf	Druid	Male	20	4 years	10
234	49975e000a7d	2009-02-15 00:12:48	2009-02-15 00:14:37	Scilla	Belgium	Chicago	80	Male	Human	Death Kn	Male	17	3 years	too many
235	49975e000a7d	2009-02-15 00:14:37	2009-02-15 00:14:48	Scilla	Belgium	Chicago	80	Male	Human	Death Kn	Male	17	3 years	too many
236	49975e000a7d	2009-02-15 00:14:48	2009-02-15 00:14:55	Scilla	Belgium	Chicago	80	Male	Human	Death Kn	Male	17	3 years	too many
237	49975e68a5f56	2009-02-15 00:14:32	2009-02-15 00:16:51	[redacted]	7th circle c	:)	99	Not Specif	Troll	Shaman	Not Speci	69	eighteen ye	none
238	49975e68a5f56	2009-02-15 00:16:51	2009-02-15 00:17:01	[redacted]	7th circle c	:)	99	Not Specif	Troll	Shaman	Not Speci	69	eighteen ye	none
239	49975e9b31ffd	2009-02-15 00:15:23	2009-02-15 00:17:21	Cairne	United Sta	Jeer Koff, Ohio	21	Male	Night Elf	Druid	Male	17	Three weel	20+
240	49975ece2443f	2009-02-15 00:16:14	2009-02-15 00:17:45	[redacted]				Male	Gnome	Death Kn	Male	[redacted]		
241	49975ff8002f01	2009-02-15 00:21:12	2009-02-15 00:22:20	Tichondrius	New Zeala	None	71	Male	Troll	Mage	Male	16	4 years	100
242	49975f196376f	2009-02-15 00:17:29	2009-02-15 00:22:57	Caelestasz	Australia	Perth	80	Female	Night Elf	Hunter	Female	13	1 Year ther	16-20
243	4997605532950	2009-02-15 00:22:45	2009-02-15 00:24:56	dunemaul	canada	coquitlam	80	Female	Human	Paladin	Male	18	since may :	4-10
244	4997606e611da	2009-02-15 00:23:10	2009-02-15 00:25:37	Shadowmoor	Ukraine	Simferopol	80	Male	Tauren	Druid	Male	15	1 year	10
245	4997606e611da	2009-02-15 00:25:37	2009-02-15 00:25:52	Shadowmoor	Ukraine	Simferopol	80	Male	Tauren	Druid	Male	15	1 year	10

retail	30-40	Wrath of the Lich King	N	N		2009-02-14 23:43:54
retail	30-40	Wrath of the Lich King	Y	Y		2009-02-14 23:43:54
Business owner	60	Wrath of the Lich King	N	N		2009-02-14 23:48:58
Business owner	60	Wrath of the Lich King	Y	Y		2009-02-14 23:48:58
Student/ Retail	60	Wrath of the Lich King	N	N		2009-02-14 23:50:01
Student/ Retail	60	Wrath of the Lich King	Y	Y		2009-02-14 23:50:01
student	40 hours	Wrath of the Lich King	N	N		2009-02-14 23:49:29
student	40 hours	Wrath of the Lich King	N	N		2009-02-14 23:49:29
student	40 hours	Wrath of the Lich King	Y	Y		2009-02-14 23:49:29
Student	2	Wrath of the Lich King	Y	N	Hai	2009-02-14 23:51:31
Unemployed	--------	Wrath of the Lich King	Y	Y	Good Luck!	2009-02-14 23:52:19
middleschool	6 hours	Wrath of the Lich King	Y	Y	no!	2009-02-14 23:57:18
Student	40	Classic	Y	Y		2009-02-14 23:57:31
Student	35	Wrath of the Lich King	Y	Y		2009-02-14 23:58:48
Student	No idea	Wrath of the Lich King	Y	Y		2009-02-14 23:59:05
student	35	Wrath of the Lich King	Y	Y	not really	2009-02-14 23:58:57
freeloader	zero	Wrath of the Lich King	Y	Y	wow rules	2009-02-14 23:59:25
		Not-Specified	N	N		2009-02-15 00:01:39
Commercial Pilot	40	Wrath of the Lich King	Y	Y	I hope this isr	2009-02-15 00:01:00
Apprentice Chef.	38	Wrath of the Lich King	N	N	Honestly, I&#	2009-02-15 00:01:39
Student	All of my time	Wrath of the Lich King	Y	Y	[redacted]	2009-02-15 00:01:39
Apprentice Chef.	38	Wrath of the Lich King	N	N	Honestly, I\&#	2009-02-15 00:01:39
Student	35	Wrath of the Lich King	Y	Y	Nope	2009-02-15 00:02:45
therapy for the criminally	40	Wrath of the Lich King	Y	Y	hubby, son ar	2009-02-15 00:02:16
computer networking stu	25	Wrath of the Lich King	N	N		2009-02-15 00:04:07
computer networking stu	25	Wrath of the Lich King	N	N		2009-02-15 00:04:07
computer networking stu	25	Wrath of the Lich King	Y	Y		2009-02-15 00:04:07
Lab Technician	30	Wrath of the Lich King	Y	Y		2009-02-15 00:05:03
student	40	Wrath of the Lich King	N	N		2009-02-15 00:07:26
senior in high school	too many	Wrath of the Lich King	N	N	WoW is more	2009-02-15 00:12:48
senior in high school	too many	Wrath of the Lich King	N	N	WoW is more	2009-02-15 00:12:48
senior in high school	too many	Wrath of the Lich King	Y	Y	WoW is more	2009-02-15 00:12:48
candy striper	not much, i go '	Wrath of the Lich King	N	N	Internet surve	2009-02-15 00:14:32
candy striper	not much, i go '	Wrath of the Lich King	Y	N	Internet surve	2009-02-15 00:14:32
Student (failing every cla	over 9,000	Classic	Y	Y	[redacted]	2009-02-15 00:15:23
[redacted]		Wrath of the Lich King	Y	N	[redacted]	2009-02-15 00:16:14
Student	6 hours a day r	Wrath of the Lich King	Y	Y	nope	2009-02-15 00:21:12
Student	40+	Wrath of the Lich King	Y	Y		2009-02-15 00:17:29
electrican	48	Wrath of the Lich King	Y	Y		2009-02-15 00:22:45
student	30?	Wrath of the Lich King	N	N		2009-02-15 00:23:10
student	30?	Wrath of the Lich King	Y	Y		2009-02-15 00:23:10

246	499761019569l	2009-02-15 00:25:37	2009-02-15 00:26:03					Not Specif	Not-Spe	Not-Spec	Not Specified			
247	499761019569l	2009-02-15 00:26:03	2009-02-15 00:26:17					Not Specif	Not-Spe	Not-Spec	Not Specified			
248	4997618bc27fa	2009-02-15 00:27:55	2009-02-15 00:29:15	tichondrius	united stat	Houston	80	Male	Orc	Death Kn	Male	17	3 years	7
249	4997639a3710	2009-02-15 00:36:42	2009-02-15 00:38:38	Dragonblight	North Ame	Fort Walton Be	80	Female	Blood El	Paladin	Male	17	Three Year	20-25
250	499764b04b50	2009-02-15 00:41:20	2009-02-15 00:42:17	Nazjatar	US	Gig Harbor	80	Male	Tauren	Druid	Male	14	5 Years	20
251	499764fb833fe	2009-02-15 00:42:35	2009-02-15 00:43:31	Draenor	Russia	New York	80	Male	Blood El	Death Kn	Male	5555	555555555	555555555
252	499766736ebbl	2009-02-15 00:48:51	2009-02-15 00:50:24	Fenris	USA	Toms River	80	Male	Draenei	Paladin	Male	16	3 years	15
253	499768c2f2a1f	2009-02-15 00:58:42	2009-02-15 01:04:26	The Forgotte	Japan	Osaka	70	Female	Night Elf	Druid	Female	20	2 1/2 years	Between 9
254	499768c2f2a1f	2009-02-15 01:04:26	2009-02-15 01:04:53	The Forgotte	Japan	Osaka	70	Female	Night Elf	Druid	Female	20	2 1/2 years	Between 1
255	499768c2f2a1f	2009-02-15 01:04:53	2009-02-15 01:05:09	The Forgotte	Japan	Osaka	70	Female	Night Elf	Druid	Female	20	2 1/2 years	Between 1
256	49976affd054f4	2009-02-15 01:08:15	2009-02-15 01:10:51	Jupiter	China	Mexico City	Over 9	Not Specif	Draenei	Warlock	Not Speci	Unde	Forever	169
257	49976affd054f4	2009-02-15 01:10:42	2009-02-15 01:11:05	Jupiter	China	Mexico City	Over 9	Not Specif	Draenei	Warlock	Not Speci	Unde	Forever	169
258	49976affd054f4	2009-02-15 01:11:05	2009-02-15 01:11:13	Jupiter	China	Mexico City	Over 9	Not Specif	Draenei	Warlock	Not Speci	Unde	Forever	169
259	49976affd054f4	2009-02-15 01:10:42	2009-02-15 01:13:47	Jupiter	China	Mexico City	Over 9	Not Specif	Draenei	Warlock	Not Speci	Unde	Forever	169
260	49976affd054f4	2009-02-15 01:08:15	2009-02-15 01:13:49	Jupiter	China	Mexico City	Over 9	Not Specif	Draenei	Warlock	Not Speci	Unde	Forever	169
261	49976dd8eb09	2009-02-15 01:20:24	2009-02-15 01:23:08	malfurion	USA	mesa	70	Female	Night Elf	Rogue	Male	16	2 years	15
262	49976eb77227	2009-02-15 01:24:07	2009-02-15 01:26:04	Uther	US	WIlliston, ND	80	Male	Orc	Warrior	Male	27	a little over	~20-25
263	49976f2c59287	2009-02-15 01:26:04	2009-02-15 01:27:36	Madoran	United Sta	Dublin	80	Male	Orc	Warrior	Male	15	2.5 years	10-12
264	49976f42d790b	2009-02-15 01:26:26	2009-02-15 01:27:43	Ner'zhu	USA	Rochester	80	Male	Dwarf	Hunter	Male	15	~2 years	30ish.
265	49977188c2c1	2009-02-15 01:36:08	2009-02-15 01:38:40	tich	canada		20	Male	Dwarf	Hunter	Male	18	1	2
266	49977666adee	2009-02-15 01:56:54	2009-02-15 01:58:34	Akama	Venezuela	Caracas	80	Male	Night Elf	Warrior	Male	15	About a ye	3 or 2 depe
267	499777385bfac	2009-02-15 02:00:24	2009-02-15 02:24:11	Proudmoore/	USA	Phoenix	80	Male	Undead	Mage	Male	15	3 and a hal	like 4-8 but
268	49977cef6b333	2009-02-15 02:24:47	2009-02-15 02:25:52	Gurubashi	USA	Texas City	80	Female	Tauren	Warrior	Male	17	3 years	10ish
269	49977fb70cd40	2009-02-15 02:36:39	2009-02-15 02:39:01	y0	momm@h	bish	fucking	Not Specif	Draenei	Death Kn	Not Speci	i hae	shut up u f	dumb ass
270	4997893b188b	2009-02-15 03:17:15	2009-02-15 03:19:04	thaurissan	australia	tully	80	Male	Dwarf	Hunter	Male	18	4 years	168
271	49979bfd9f4dc	2009-02-15 04:37:17	2009-02-15 04:39:40	Moon Guard	USA		80	Female	Troll	Hunter	Female	32	Since laun	30
272	49987335ca8c	2009-02-15 19:55:34	2009-02-15 20:00:35	Argent Dawn	Canada	St-Vianney	80	Female	Tauren	Druid	Male	17	2 years 7 n	50
273	49987335ca8c	2009-02-15 20:00:35	2009-02-15 20:00:49	Argent Dawn	Canada	St-Vianney	80	Female	Tauren	Druid	Male	17	2 years 7 n	50
274	49987390eacc	2009-02-15 19:57:04	2009-02-15 20:01:01	Area 52	United Sta	Martinez, CA	80	Male	Undead	Warrior	Male	23	1.5 years	30
275	49987335ca8c	2009-02-15 20:00:49	2009-02-15 20:01:05	Argent Dawn	Canada	St-Vianney	80	Female	Tauren	Druid	Male	17	2 years 7 n	50
276	49987335ca8c	2009-02-15 20:01:05	2009-02-15 20:01:20	Argent Dawn	Canada	St-Vianney	80	Female	Tauren	Druid	Male	17	2 years 7 n	50
277	4998748e5bad	2009-02-15 20:01:18	2009-02-15 20:02:28	Bloodhoof	America	Moore	80	Female	Tauren	Druid	Female	21	4 years	9 hours
278	4998748e5bad	2009-02-15 20:02:28	2009-02-15 20:02:37	Bloodhoof	America	Moore	80	Female	Tauren	Druid	Female	21	4 years	9 hours
279	4998750cb581	2009-02-15 20:03:24	2009-02-15 20:04:31	Alleria	US	Athens, Alaban	80	Male	Tauren	Druid	Male	17	4 years.	20+
280	49987792c042	2009-02-15 20:14:10	2009-02-15 20:16:06	misha	US	blacksburg	80	Female	Draenei	Shaman	Male	NA	4 years	20
281	4998796ea106	2009-02-15 20:22:06	2009-02-15 20:29:51	Saurfang	Indonesia	Jakarta	80	Male	Blood El	Warlock	Male	32	3 years	about 20 h
282	49987b0380d2	2009-02-15 20:28:51	2009-02-15 20:31:31	Uldum	United Sta	New York	80	Male	Human	Paladin	Male	15	3 years	15-20
283	49987d2189fe0	2009-02-15 20:37:53	2009-02-15 20:41:00	Dethecus	Canada	Milton	80	Male	Tauren	Druid	Male	18	4 years	28
284	499883d9e582l	2009-02-15 21:06:33	2009-02-15 21:08:14	Scarlet Crusa	USA		80	Female	Night Elf	Hunter	Female	23	4 Years	too many
285	499885117194	2009-02-15 21:11:45	2009-02-15 21:12:37	Durotan	Canada	QuÃ©bec	80	Male	Troll	Priest	Male	19	4	20
286	499885117194	2009-02-15 21:12:37	2009-02-15 21:12:46	Durotan	Canada	QuÃ©bec	80	Male	Troll	Priest	Male	19	4	20

		Not-Specified	N	N	[redacted]	2009-02-15 00:25:37
		Not-Specified	Y	N		2009-02-15 00:25:37
student	60-70	Wrath of the Lich King	Y	Y		2009-02-15 00:27:55
College Student	35-37	Wrath of the Lich King	Y	Y	Good luck ga	2009-02-15 00:36:42
Student	30	Wrath of the Lich King	Y	Y		2009-02-15 00:41:20
555555555555555555	555555555555	Wrath of the Lich King	Y	N	555555555h	2009-02-15 00:42:35
student	40	Wrath of the Lich King	Y	Y		2009-02-15 00:48:51
College, double majoring	College + Job =	Wrath of the Lich King	N	N	Ganbatte! ^-^	2009-02-15 00:58:42
College, double majoring	College + Job =	Wrath of the Lich King	N	N	Ganbatte! ^-^	2009-02-15 00:58:42
College, double majoring	College + Job =	Wrath of the Lich King	Y	Y	Ganbatte! ^-^	2009-02-15 00:58:42
Homeless	0	Wrath of the Lich King	N	N	no	2009-02-15 01:08:15
Homeless	0	Wrath of the Lich King	N	N	no	2009-02-15 01:08:15
Homeless	0	Wrath of the Lich King	Y	N	no	2009-02-15 01:08:15
Homeless	0	Wrath of the Lich King	N	N	no	2009-02-15 01:08:15
Homeless	0	Wrath of the Lich King	N	N	no	2009-02-15 01:08:15
student	36	Burning Crusade	Y	Y		2009-02-15 01:20:24
Meterologist	40	Wrath of the Lich King	Y	Y		2009-02-15 01:24:07
student	EVERY SINGL	Wrath of the Lich King	Y	Y	no	2009-02-15 01:26:04
Student	35	Wrath of the Lich King	Y	Y		2009-02-15 01:26:26
school	30	Wrath of the Lich King	Y	Y		2009-02-15 01:36:08
Student	8hrs + Homew	Wrath of the Lich King	Y	Y	CwutIdidthar?	2009-02-15 01:56:54
hustler	however many	Wrath of the Lich King	Y	Y	MAI KEYZ MA	2009-02-15 02:00:24
Student	Too many	Wrath of the Lich King	Y	Y		2009-02-15 02:24:47
ass hole	ur ugly	Wrath of the Lich King	Y	N	I would like to	2009-02-15 02:36:39
wow player	168	Wrath of the Lich King	Y	Y	what year is it	2009-02-15 03:17:15
Quality Assurance	40+	Wrath of the Lich King	Y	Y		2009-02-15 04:37:17
High School Senior	40	Wrath of the Lich King	N	N		2009-02-15 19:55:34
High School Senior	40	Wrath of the Lich King	N	N	a 9 Q x S i	2009-02-15 19:55:34
Student	30	Wrath of the Lich King	Y	Y		2009-02-15 19:57:04
High School Senior	40	Wrath of the Lich King	N	N		2009-02-15 19:55:34
High School Senior	40	Wrath of the Lich King	Y	Y		2009-02-15 19:55:34
College Student	12 hours	Not-Specified	N	N		2009-02-15 20:01:18
College Student	12 hours	Not-Specified	Y	Y		2009-02-15 20:01:18
Student	35+	Wrath of the Lich King	Y	Y	:3	2009-02-15 20:03:24
freelancer	30 - 40	Wrath of the Lich King	Y	N	nope	2009-02-15 20:14:10
it security consultant	50	Wrath of the Lich King	Y	Y		2009-02-15 20:22:06
Student	30	Wrath of the Lich King	Y	Y		2009-02-15 20:28:51
Student/Cashier	30	Wrath of the Lich King	Y	Y		2009-02-15 20:37:53
Graphic Artist	40	Wrath of the Lich King	Y	Y		2009-02-15 21:06:33
Student	45	Wrath of the Lich King	N	N		2009-02-15 21:11:45
Student	45	Wrath of the Lich King	Y	Y		2009-02-15 21:11:45

Appendix B - Online Implementation Second Phase Dataset Sample

This appendix consists of the survey data from the second phase, which is post-implementation of GARDEN to the process. This is the phase that shows the improvement GARDEN made to the process. The data is presented in the same structure and format as Appendix A. Also, like in Appendix A this data is censored in some places due to the vulgar language used by some of the participants, however it has not been altered in any way.

id	session	start time	end time	server	country	city	level	char genc	race	class	real gende	age	play tim
1	4990d43dd252c	2009-02-10 01:11:25	2009-02-10 01:12:16	Barthilas	Australia	Melbourne	80	Male	Night Elf	Druid	Male	27	4 years
2	4990f3df768d23.	2009-02-10 03:26:23	2009-02-10 03:28:17	Hyjal	United State	Honolulu	80	Female	Human	Death Knig	Male	15	1.5 years
3	49911f643947a7	2009-02-10 06:32:04	2009-02-10 06:33:43	Eldre'Th	USA	Dallas	80	Female	Human	Paladin	Male	15	4 years
4	499206e138be2	2009-02-10 22:59:45	2009-02-10 23:00:36	Sisters of Elu	USA	Los Angeles	80	Female	Draenei	Priest	Female	32	3 years
5	499229198226c	2009-02-11 01:25:45	2009-02-11 01:26:52	Kirin Tor	USA	Baltimore	80	Female	Night Elf	Druid	Female	19	4 years
6	49923ab8913d4	2009-02-11 02:40:56	2009-02-11 02:43:08	Antonidas	USA	Orchard	80	Male	Troll	Rogue	Male	16	Two yea
7	49923f6a4e8a81	2009-02-11 03:00:58	2009-02-11 03:02:45	Ursin	US	Colorado Sprin	74	Male	Tauren	Druid	Male	25	2.5 years
8	49924af1bbb878	2009-02-11 03:50:11	2009-02-11 03:51:31	proudmoore	USA	Rockford Il	80	Male	Not-Specifie	Not-Specifi	Male	22	3.5 years
9	499257ee93d9b	2009-02-11 04:45:34	2009-02-11 04:47:49	dalaran	United state	long beach	77	Female	Blood Elf	Death Knig	Male	18	2 years
10	49925a81ed25b	2009-02-11 04:56:33	2009-02-11 04:58:23	Archimonde	United State	La Crosse	66	Male	Tauren	Shaman	Male	17	3 weeks
11	49925c065cfef6.	2009-02-11 05:03:02	2009-02-11 05:05:17	Cairne	Uninted Sta	Louisville, KY	69	Female	Blood Elf	Hunter	Female	20	i'd
12	499261d83f9eb9	2009-02-11 05:27:52	2009-02-11 05:29:38	shu halo	canada		72	Not Speci	Night Elf	Druid	Not Specified		
13	49927524db740	2009-02-11 06:50:12	2009-02-11 06:51:24	Spirestone	USA		80	Male	Undead	Priest	Male	16	2.5 years
14	499275d1ab2e3	2009-02-11 06:53:05	2009-02-11 06:57:10	Blackrock	Canada	Vancouver	59	Female	Blood Elf	Hunter	Female	14	since it c
15	499275d1ab2e3	2009-02-11 06:57:10	2009-02-11 06:57:23	Blackrock	Canada	Vancouver	59	Female	Blood Elf	Hunter	Female	14	since it c
16	499356e92f3575	2009-02-11 22:53:29	2009-02-11 22:54:38	Draka	USA	Norfolk	78	Female	Night Elf	Warrior	Female	20	1 year
17	499356e9b6784	2009-02-11 22:53:29	2009-02-11 22:54:53	Gilneas	Canada	Saskatoon	80	Male	Human	Paladin	Male	19	1 Year.
18	499356e962a11	2009-02-11 22:53:29	2009-02-11 22:55:03	Terenas	United State	Los Angeles	80	Female	Gnome	Warrior	Male	27	2.5 years
19	499357debc4d8	2009-02-11 22:57:34	2009-02-11 22:59:08	Hydraxis	United State	Minneapolis	54	Female	Blood Elf	Hunter	Female	18	6 or 7 m
20	499357e09d781	2009-02-11 22:57:36	2009-02-11 23:00:28	Gurubashi	USA	Cypress	76	Male	Gnome	Mage	Male	33	from the
21	49935879ea353	2009-02-11 23:00:09	2009-02-11 23:01:15	Malorne	United State	Austin	80	Female	Undead	Priest	Male	18	2 years
22	49935909a786f	2009-02-11 23:02:33	2009-02-11 23:04:20	Rivendare	USA	Pensacola	72	Male	Blood Elf	Death Knig	Male	32	7 months
23	499358fda81437	2009-02-11 23:02:21	2009-02-11 23:05:58	alterac mount	US	Philadelphia	80	Female	Blood Elf	Hunter	Female	21	8 months
24	499359e922416	2009-02-11 23:06:17	2009-02-11 23:08:18	Detheroc	England	Leeds	80	Male	Gnome	Death Knig	Male	16	2.5 years
25	49935ae5c7f5d0	2009-02-11 23:10:29	2009-02-11 23:11:50	Backhand	USA	-.-	80	Male	Troll	Hunter	Male	17	3 year
26	49935afac35877	2009-02-11 23:10:50	2009-02-11 23:11:51	Uldaman	United State	Louisville	80	Male	Blood Elf	Paladin	Not Specifi	15	3 years
27	49935afac35877	2009-02-11 23:11:51	2009-02-11 23:12:17	Uldaman	United State	Louisville	80	Male	Blood Elf	Paladin	Not Specifi	15	3 years
28	49935b16dab31	2009-02-11 23:11:18	2009-02-11 23:14:35	Lothar	United State	Williamsport, P	80	Female	Night Elf	Druid	Female	24	4 years
29	49935b16dab31	2009-02-11 23:14:35	2009-02-11 23:14:43	Lothar	United State	Williamsport, P	80	Female	Night Elf	Druid	Female	24	4 years
30	49935c895b352	2009-02-11 23:17:29	2009-02-11 23:19:40	Proudmoore	United State	Costa Mesa	80	Female	Dwarf	Paladin	Male	30	5 years
31	49935d3fc200b9	2009-02-11 23:20:31	2009-02-11 23:22:32	gilneas	canada		80	Male	Night Elf	Warrior	Male		1 year
32	49935df6913590	2009-02-11 23:23:34	2009-02-11 23:25:50	Drak'Tha	U.S.A	Atlanta	80	Male	Undead	Rogue	Male	17	2.5 years
33	49935ea242936	2009-02-11 23:26:26	2009-02-11 23:27:25	Caelestrasz	NZ	Dunedin	80	Male	Night Elf	Hunter	Male	23	10 mont
34	49935f3dd714f3.	2009-02-11 23:29:01	2009-02-11 23:31:26	Turalyon	USA	Mead	80	Female	Blood Elf	Warlock	Male	16	Since Be
35	49935f872b4905	2009-02-11 23:30:15	2009-02-11 23:31:27	Perenolde	Canada	Port Hardy	80	Male	Tauren	Druid	Male	23	4
36	49935ec996ad9	2009-02-11 23:27:05	2009-02-11 23:32:44	Ghostlands	USA	North Caroina	67	Female	Blood Elf	Hunter	Female	21	1 year
37	4993606c1cda1	2009-02-11 23:34:04	2009-02-11 23:35:10	Windrunner	United State	Stamford	80	Male	Orc	Warlock	Male	17	2 years
38	4993618adfef94.	2009-02-11 23:38:50	2009-02-11 23:40:31	Whisperwind	United State	Schwenksville	80	Male	Blood Elf	Death Knig	Male	15	4 years
39	4993618a3662f3	2009-02-11 23:38:50	2009-02-11 23:40:41	Fizzcrank	Denmark	Svendborg	80	Male	Human	Paladin	Male	15	6 months
40	499361b8a372f3	2009-02-11 23:39:36	2009-02-11 23:41:13	Exodar	United State	Cullpepper Virg	80	Male	Undead	Death Knig	Male	14	3 years

hours week	occupation	job time	game settings	captch	valid	comments	real_start_time
10	web developer	40	Wrath of the Lich King	Y	Y		2009-02-10 01:11:25
15 hours	Student	30 hours a week	Wrath of the Lich King	Y	Y		2009-02-10 03:26:23
60ish	Student	8 hours	Wrath of the Lich King	Y	Y	I hate this ga	2009-02-10 06:32:04
25+	Illustrator	40+	Wrath of the Lich King	Y	Y		2009-02-10 22:59:45
7	Nursing major	A lot	Wrath of the Lich King	Y	Y		2009-02-11 01:25:45
Too many.	Student	Thirty five.	Wrath of the Lich King	Y	Y	Carthago de	2009-02-11 02:40:56
8-10 hours	Commercial Insurace	40 hours	Wrath of the Lich King	Y	Y	I	2009-02-11 03:00:58
20	graduate student	35	Wrath of the Lich King	Y	Y		2009-02-11 03:50:11
54	student	45	Wrath of the Lich King	Y	Y		2009-02-11 04:45:34
14ish	student/waiter	40 hours	Wrath of the Lich King	Y	Y		2009-02-11 04:56:33
it depends or	college student	too damn many	Burning Crusade	Y	Y	[redacted]	2009-02-11 05:03:02
75+	student	10 minutes	Wrath of the Lich King	Y	N	[redacted]	2009-02-11 05:27:52
30	Student	~35	Wrath of the Lich King	Y	Y		2009-02-11 06:50:12
21-24	student	56	Burning Crusade	N	N	plz dont b a	2009-02-11 06:53:05
21-24	student	56	Burning Crusade	Y	Y	plz dont b a	2009-02-11 06:53:05
40	Student, part time libra	40	Wrath of the Lich King	Y	Y		2009-02-11 22:53:29
~15 Hours.	Student	~25 Hours.	Wrath of the Lich King	Y	Y		2009-02-11 22:53:29
16	unemployed	0	Wrath of the Lich King	Y	Y		2009-02-11 22:53:29
about 4 hour	Student, ChildCare	40	Wrath of the Lich King	Y	Y		2009-02-11 22:57:34
20-30	shipping manger	40-60	Wrath of the Lich King	Y	Y		2009-02-11 22:57:36
12	student	45	Wrath of the Lich King	Y	Y		2009-02-11 23:00:09
20	food service operation	50	Wrath of the Lich King	Y	Y	LFG IRL	2009-02-11 23:02:33
20	student/IT technician	48	Wrath of the Lich King	Y	Y		2009-02-11 23:02:21
10-20	Junior High School Stu	35 hours a week	Not-Specified	Y	Y		2009-02-11 23:06:17
15+	droped out	0!	Wrath of the Lich King	Y	Y	man i love th	2009-02-11 23:10:29
about 15			Wrath of the Lich King	N	N		2009-02-11 23:10:50
about 15	Sometimes i volunteer to do stuff		Wrath of the Lich King	Y	Y		2009-02-11 23:10:50
20	Therapeutic Support S	40	Wrath of the Lich King	N	N		2009-02-11 23:11:18
20	Therapeutic Support S	40	Wrath of the Lich King	Y	Y		2009-02-11 23:11:18
10 - 20	Manager	45	Wrath of the Lich King	Y	Y		2009-02-11 23:17:29
	truck driver	40+	Wrath of the Lich King	Y	N		2009-02-11 23:20:31
~40	student	~30	Wrath of the Lich King	Y	Y		2009-02-11 23:23:34
10-15	IT Application Analyst	40	Wrath of the Lich King	Y	Y		2009-02-11 23:26:26
24	Student	Too many.	Wrath of the Lich King	Y	Y	I like small g	2009-02-11 23:29:01
40	Flight Service Speciali	35	Wrath of the Lich King	Y	Y		2009-02-11 23:30:15
20-30	US Military	60-85	Wrath of the Lich King	Y	Y		2009-02-11 23:27:05
20	student	42	Wrath of the Lich King	Y	Y		2009-02-11 23:34:04
3-4	student	over 35	Wrath of the Lich King	Y	Y	I don't	2009-02-11 23:38:50
About 6 now	High School	about 30 hours	Wrath of the Lich King	Y	Y	I don't	2009-02-11 23:38:50
40ish	student	40ish? Studying	Wrath of the Lich King	Y	Y		2009-02-11 23:39:36

41	499361967395e(	2009-02-11 23:39:02	2009-02-11 23:41:15	Silver Hand	USA	Fremont	80	Male	Draenei	Shaman	Male	26	2004
42	499361b32c72b2	2009-02-11 23:39:31	2009-02-11 23:41:23	Silvermoon	USA	Tracy	80	Female	Night Elf	Druid	Male	28	1 Year
43	499361967395e(	2009-02-11 23:41:15	2009-02-11 23:41:23	Silver Hand	USA	Fremont	80	Male	Draenei	Shaman	Male	26	2004
44	4993618d32771(	2009-02-11 23:38:53	2009-02-11 23:41:26	Gorefiend	America	Pittsburgh	67	Male	Undead	Mage	Male	28	3.5 years
45	499361d108c9e2	2009-02-11 23:40:01	2009-02-11 23:41:40	The Forgotten	US	Irvine	80	Male	Night Elf	Rogue	Male	31	2.5 years
46	49936238ab18e8	2009-02-11 23:41:44	2009-02-11 23:42:38	Lightbringer	USA	San Diego	80	Male	Night Elf	Druid	Male	25	Release
47	499362161ced51	2009-02-11 23:41:10	2009-02-11 23:43:26	Dunemaul	USA	Westminster	80	Female	Undead	Priest	Male	32	4+ years
48	4993629a964eb9	2009-02-11 23:43:45	2009-02-11 23:44:52	Dragonblight	United State	Latrobe	80	Male	Blood Elf	Paladin	Male	22	3 years
49	4993621274e85	2009-02-11 23:41:06	2009-02-11 23:45:00	Arygos	Canada	Ottawa	80	Male	Draenei	Shaman	Male	16	4 years
50	4993628ff186e8.	2009-02-11 23:43:11	2009-02-11 23:45:04	Dawnbringer	United State	Scott Depot	80	Female	Tauren	Death Knig	Female	15	2 Years,
51	499362aa9693c2	2009-02-11 23:43:38	2009-02-11 23:45:16	Akama	Canada	Regina	80	Male	Gnome	Warlock	Male	18	3 years
52	499361abe18c84	2009-02-11 23:43:12	2009-02-11 23:45:42	Eonar	US	Iowa City	80	Male	Human	Mage	Male	21	4 years
53	499362bf579513	2009-02-11 23:43:59	2009-02-11 23:46:15	Elune	United State	Maine	80	Male	Dwarf	Paladin	Male	16	4 years
54	499362a16af0f5.	2009-02-11 23:43:29	2009-02-11 23:46:29	Cenarius	US	Raleigh, NC	80	Male	Human	Paladin	Male	14	2 years
55	499362a16af0f5.	2009-02-11 23:46:29	2009-02-11 23:46:36	Cenarius	US	Raleigh, NC	80	Male	Human	Paladin	Male	14	2 years
56	499362bf807a02	2009-02-11 23:43:59	2009-02-11 23:47:02	Jubei'Th	Australia	Melbourne	80	Male	Orc	Shaman	Male	22	3 years
57	49936396e89d1	2009-02-11 23:47:34	2009-02-11 23:48:47	The Venture (	United State	Lake Charles	80	Male	Tauren	Hunter	Male	12	1 year
58	4993636e61dc9	2009-02-11 23:46:54	2009-02-11 23:48:53	Khadgar	United State	Louisville	48	Female	Tauren	Druid	Female	20	4 months
59	499363b21b9e54	2009-02-11 23:48:02	2009-02-11 23:49:04	Uther	US	Anchorage	72	Male	Orc	Warlock	Not Specifi	29	4 years
60	4993634686ea3(	2009-02-11 23:48:16	2009-02-11 23:49:07	Uldum	Australia	Melbourne	80	Female	Human	Paladin	Female	26	3 years (
61	499363a2e1d49(	2009-02-11 23:47:46	2009-02-11 23:49:28	Garithos	United State	New York	58	Male	Blood Elf	Paladin	Male	13	1 year`
62	499363e131966	2009-02-11 23:48:49	2009-02-11 23:49:33	Moonrunner	Canada	Vancouver	80	Male	Human	Mage	Male	15	2.5 years
63	499363d161f783	2009-02-11 23:48:33	2009-02-11 23:49:37	Daggerspine	USA	CHicago	80	Male	Undead	Rogue	Male	20	1 year
64	4993641b180e8	2009-02-11 23:48:12	2009-02-11 23:49:47	Nagrand	New Zealar	Taranaki	80	Male	Human	Warrior	Male	36	6 months
65	499363b91d275(	2009-02-11 23:48:09	2009-02-11 23:50:12	Gorefiend	United State	Lancaster, PA	75	Male	Tauren	Druid	Male	17	6
66	4993640bcc7b7	2009-02-11 23:49:31	2009-02-11 23:50:20	Trollbane	Denmark	AllerÃ¸d	80	Male	Undead	Mage	Male	14	4 years
67	499363fb977fd0.	2009-02-11 23:49:15	2009-02-11 23:50:33	Duskwood	United States		80	Male	Dwarf	Priest	Male	17	
68	49936453012be5	2009-02-11 23:50:43	2009-02-11 23:52:29		eonar	Calgary	80	Not Speci	Blood Elf	Hunter	Male	1	10 years
69	499364d06f72f8.	2009-02-11 23:52:48	2009-02-11 23:53:45	Khagdar	USA		80	Male	Human	Mage	Male	18	6 months
70	499364d06f72f8.	2009-02-11 23:53:45	2009-02-11 23:53:51	Khagdar	USA		80	Male	Human	Mage	Male	18	6 months
71	4993654af3afc0.	2009-02-11 23:54:50	2009-02-11 23:55:03	Mok'Nat	United States			Not Speci	Not-Specifie	Not-Specifi	Not Specified		
72	4993652ab8ed24	2009-02-11 23:54:18	2009-02-11 23:55:31	Whisperwind	United State	Olathe	80	Male	Human	Rogue	Male	13	3ish yea
73	49936501941a8	2009-02-11 23:53:37	2009-02-11 23:55:40	Dragonmaw	United State	Sausalito	61	Male	Human	Priest	Male	17	3 years
74	4993655d6b4fb7	2009-02-11 23:55:09	2009-02-11 23:56:13	Azjol-Nerub	USA	Albuquerque	80	Female	Dwarf	Paladin	Female	34	3.5 years
75	499365593822b2	2009-02-11 23:55:05	2009-02-11 23:56:17	Quel'dor	United State	Lewiston	80	Male	Human	Priest	Male	16	Four yea
76	4993655c81f887	2009-02-11 23:55:08	2009-02-11 23:56:20	Suramar	US	Salt Lake City	80	Male	Troll	Shaman	Male	17	3 years
77	4993656ec1050	2009-02-11 23:55:26	2009-02-11 23:56:22	Fenris	US	Goshen	80	Female	Human	Warrior	Male	17	2 years
78	4993656733a02(	2009-02-11 23:55:19	2009-02-11 23:56:31	Blackrock	United State	Los Angeles	80	Not Speci	Orc	Warrior	Male	18	2 years
79	49936567b060a(	2009-02-11 23:55:19	2009-02-11 23:56:46	Scarlet Crusa	United State	Anderson	80	Male	Dwarf	Paladin	Male	23	4 Years
80	49936566e7d47	2009-02-11 23:55:18	2009-02-11 23:56:46	firetree	usa	dc	80	Male	Draenei	Shaman	Male	25	since be
81	4993657029e22	2009-02-11 23:55:28	2009-02-11 23:56:47	Fizzcrank	USA	Corvallis, Oreg	80	Female	Undead	Warlock	Female	28	Release,

10-12	Database Administrato	45-50	Wrath of the Lich King	N	N		2009-02-11 23:39:02
22	Software Engineer	40+	Wrath of the Lich King	Y	Y		2009-02-11 23:39:31
10-12	Database Administrato	45-50	Wrath of the Lich King	Y	Y		2009-02-11 23:39:02
40-50	Network Engineer	40-50	Wrath of the Lich King	Y	Y	*Should be r	2009-02-11 23:38:53
5 - 10	Sr. HR Administrator	40	Wrath of the Lich King	Y	Y		2009-02-11 23:40:01
10	Finance	45	Wrath of the Lich King	Y	Y		2009-02-11 23:41:44
15-20	management	40-50	Wrath of the Lich King	Y	Y	good luck or	2009-02-11 23:41:10
20	PC Tech	40	Wrath of the Lich King	Y	Y		2009-02-11 23:43:45
32	student	30	Wrath of the Lich King	Y	Y		2009-02-11 23:41:06
6-18	Student	35	Wrath of the Lich King	Y	Y		2009-02-11 23:43:11
Too many	Poli Sci major (lololol	Including homew	Wrath of the Lich King	Y	Y	Add me on I	2009-02-11 23:43:38
18	Student	20	Wrath of the Lich King	Y	Y		2009-02-11 23:43:12
1-30 hours	Student	49 hours	Wrath of the Lich King	Y	Y		2009-02-11 23:43:59
maybe 30?	student	35 hours	Wrath of the Lich King	N	N		2009-02-11 23:43:29
maybe 30?	student	35 hours	Wrath of the Lich King	Y	Y		2009-02-11 23:43:29
40+	RFID Engineer	24	Wrath of the Lich King	Y	Y	Glad to help	2009-02-11 23:43:59
12-16	Student	40	Wrath of the Lich King	Y	Y		2009-02-11 23:47:34
8	Full-time nursing stude	35	Burning Crusade	Y	Y		2009-02-11 23:46:54
10	Net Admin	40-60	Wrath of the Lich King	Y	Y		2009-02-11 23:48:02
up to 40	Stay at home mum	24/7 :)	Wrath of the Lich King	Y	Y		2009-02-11 23:48:16
I don't e	Student	42+	Burning Crusade	Y	Y	Long Live W	2009-02-11 23:47:46
20	student	35	Wrath of the Lich King	Y	Y		2009-02-11 23:48:49
15	student	20	Wrath of the Lich King	Y	Y		2009-02-11 23:48:33
40	professional	40	Wrath of the Lich King	N	N		2009-02-11 23:48:12
10-12	Student in high school	8	Wrath of the Lich King	Y	Y	lolololololol	2009-02-11 23:48:09
80-90	Student	45	Wrath of the Lich King	Y	Y		2009-02-11 23:49:31
10-15	Student	40	Wrath of the Lich King	Y	Y		2009-02-11 23:49:15
168	[redacted]	20	Wrath of the Lich King	Y	N		2009-02-11 23:50:43
30	Unemployed	0	Wrath of the Lich King	N	N		2009-02-11 23:52:48
30	Unemployed	0	Wrath of the Lich King	Y	Y		2009-02-11 23:52:48
			Not-Specified	N	N		2009-02-11 23:54:50
20-30	Student	35 hours, exactly	Wrath of the Lich King	Y	Y		2009-02-11 23:54:18
15	Student	20	Wrath of the Lich King	Y	Y	New accour	2009-02-11 23:53:37
15-20	personal trainer	20-30	Wrath of the Lich King	Y	Y		2009-02-11 23:55:09
Used to play	Student	Too many.	Wrath of the Lich King	Y	Y	160 days pla	2009-02-11 23:55:05
20	salesman	7 1/2 hours	Wrath of the Lich King	Y	Y	nope.	2009-02-11 23:55:08
25+	student	30+	Wrath of the Lich King	Y	Y		2009-02-11 23:55:26
15	Student	8	Wrath of the Lich King	Y	Y		2009-02-11 23:55:19
35+ Hours	Sears RTV Associate	30+ Hours	Wrath of the Lich King	Y	Y		2009-02-11 23:55:19
8-10	photo restorer	55	Wrath of the Lich King	Y	Y		2009-02-11 23:55:18
14	Student	40	Wrath of the Lich King	Y	Y		2009-02-11 23:55:28

82	49936566e03d49	2009-02-11 23:55:18	2009-02-11 23:56:59	Gul'dan	USA	Lawrence, ks	80	Male	Blood Elf	Paladin	Male	25	about 2 y
83	4993654af3afc0.	2009-02-11 23:55:03	2009-02-11 23:57:09	Mok'Nat	United State	Killingly	80	Male	Orc	Hunter	Female	16	2 years
84	49936596e470a8	2009-02-11 23:56:06	2009-02-11 23:57:48	Lightbringer	USA	Denver	80	Female	Draenei	Paladin	Male	36	4 Years
85	499365ef639026	2009-02-11 23:57:35	2009-02-11 23:58:35	Malygos	US	Sacramento	80	Male	Human	Warrior	Male	18	2 1/2 yea
86	499365c4dcba45	2009-02-11 23:56:52	2009-02-11 23:59:12	Cho'Gal	USA	NRH TX	80	Male	Undead	Death Knig	Male	16	5 Years
87	4993663fd6d2b7	2009-02-11 23:58:55	2009-02-12 00:00:21	Vashj	Africa	Sydney	00	Female	Blood Elf	Shaman	Female	99	OVER 9(
88	4993663714f216	2009-02-11 23:58:47	2009-02-12 00:00:38	Destromath	USA	Tacoma	80	Male	Human	Paladin	Male	27	3 years
89	499366b5d52f58	2009-02-12 00:00:53	2009-02-12 00:01:06	Duskwood	United States			Not Speci	Not-Specifie	Not-Specifi	Not Specified		
90	499366929bb04(	2009-02-12 00:00:18	2009-02-12 00:01:25	Ghostlands	United State	Little Rock	80	Male	Undead	Priest	Male	16	3 years
91	4993668ff14726.	2009-02-12 00:00:15	2009-02-12 00:01:36	Daggerspine	US of A BAI	North Texas	80	Female	Night Elf	Priest	Male	17	3 years
92	4993664d4d8d38	2009-02-11 23:59:09	2009-02-12 00:01:53	Cenarius	United State	Pacific Palisad	80	Female	Human	Priest	Male	19	4 Years
93	499366b5d52f58	2009-02-12 00:01:06	2009-02-12 00:01:55	Duskwood	United State	Small Town	80	Male	Night Elf	Druid	Male	16	2-ish yea
94	499366c46b8e5(	2009-02-12 00:01:08	2009-02-12 00:02:06	Mal'Gan	USA	Waco, TX	80	Male	Tauren	Shaman	Male	20	~2 years
95	499366c0181a43	2009-02-12 00:01:04	2009-02-12 00:02:32	drenden	U.S.A	Williamsport	80	Male	Human	Paladin	Male	14	like a yea
96	499366b7024d5(	2009-02-12 00:00:55	2009-02-12 00:02:52	Ner'Zhul	Canada	Calgary	80	Female	Draenei	Shaman	Male	22	4 Years
97	49936702622fb3	2009-02-12 00:02:10	2009-02-12 00:03:22	genesis	United State	Nashua	80	Male	Night Elf	Druid	Male	16	3 years
98	4993670524c433	2009-02-12 00:02:13	2009-02-12 00:03:29	Area 52	United State	Clarksville	80	Male	Blood Elf	Paladin	Male	14	5 years
99	499366a5382ba5	2009-02-12 00:00:37	2009-02-12 00:03:53	Durotan	USA	Berkeley, CA	80	Male	Gnome	Death Knig	Male	24	5 years
100	499366eb2ba5b7	2009-02-12 00:01:47	2009-02-12 00:03:58	Blade's I	Canada	Crofton	80	Male	Human	Paladin	Male	12	2-3 years
101	4993679056366	2009-02-12 00:04:32	2009-02-12 00:05:29	NONE	LOL	Over 9000	90	Female	Tauren	Rogue	Female	90	900000
102	49936796e46f12	2009-02-12 00:04:38	2009-02-12 00:05:31	Skywall	US	Columbus	80	Female	Blood Elf	Rogue	Female	21	3.5 years
103	4993676e7623c5	2009-02-12 00:03:58	2009-02-12 00:05:40	Zuluhed	United State	Hartford	80	Female	Troll	Rogue	Male	19	4 years
104	4993677d3945d2	2009-02-12 00:04:13	2009-02-12 00:05:41	Burning Legio	Usa	Edmond	Va	Male	Blood Elf	Paladin	Male	17	1 year
105	499367a63e22c9	2009-02-12 00:04:54	2009-02-12 00:05:56	Frostmane	USA	CC,texas	80	Male	Human	Paladin	Male	17	4 years
106	499367b8b7cc43	2009-02-12 00:05:12	2009-02-12 00:05:59	Drak'Tha	USA	San Francisco	80	Female	Night Elf	Death Knig	Male	16	2 Years
107	499367cb88e8f9	2009-02-12 00:05:31	2009-02-12 00:06:27	Misha	United State	Houston	79	Male	Blood Elf	Rogue	Male	17	2 months
108	499367a63e22c9	2009-02-12 00:05:56	2009-02-12 00:06:36	Frostmane	USA	CC,texas	80	Male	Human	Paladin	Male	17	4 years
109	499367ca816409	2009-02-12 00:05:30	2009-02-12 00:06:44	Dentarg	Canada	Edmonton	80	Female	Human	Paladin	Male	20	2 years
110	499367e010b988	2009-02-12 00:05:52	2009-02-12 00:06:57	Dark Iron	Australia	Sydney	80	Female	Draenei	Shaman	Male	22	4 years
111	4993666e746ae7	2009-02-11 23:59:42	2009-02-12 00:07:17	Gilneas	US	Tucson	68	Female	Human	Paladin	Male	15	A whole
112	499367de024e75	2009-02-12 00:05:50	2009-02-12 00:08:18	Hellscream	Canada	Toronto	71	Male	Undead	Death Knig	Male	27	2005
113	499367418fb6e8	2009-02-12 00:03:13	2009-02-12 00:08:38	Nazjatar	United State	Philadelphia	70	Male	Undead	Warrior	Male	15	1.5 years
114	49936879e705d(	2009-02-12 00:08:25	2009-02-12 00:09:46	warsong	USA	williamsburg	80	Female	Blood Elf	Warlock	Male	23	since sta
115	499368608cd701	2009-02-12 00:08:00	2009-02-12 00:09:54	Thunderlord	Australia	Adelaide	70	Male	Tauren	Warrior	Male	18	4 years
116	499368989c005(	2009-02-12 00:08:56	2009-02-12 00:10:15	The Underbog	US	Sarasota	80	Female	Blood Elf	Paladin	Male	28	since lau
117	499368b02ba3f2	2009-02-12 00:09:20	2009-02-12 00:10:21	dentarg	usa	houston	80	Male	Tauren	Shaman	Male	25	4 years
118	499368605f8b37	2009-02-12 00:08:00	2009-02-12 00:10:44	Bonechewer	United State	Hebron	80	Male	Dwarf	Paladin	Male	17	4 years
119	499368a2594428	2009-02-12 00:09:06	2009-02-12 00:10:47	Thrall	USA	Atlanta	80	Male	Night Elf	Hunter	Male	21	4 years
120	499368295a9f06	2009-02-12 00:07:05	2009-02-12 00:11:04	Doomhamme	USA	Kansas City	80	Female	Night Elf	Druid	Male	21	Off and c
121	499368abb6bdb8	2009-02-12 00:09:15	2009-02-12 00:11:43	Frostmane	US	Boston	80	Female	Night Elf	Druid	Female	21	About 2-
122	49936901e47a27	2009-02-12 00:10:41	2009-02-12 00:11:51	Kael'Tha	United State	Pensacola	80	Male	Gnome	Death Knig	Male	19	4 years

15	Researcher	40ish	Wrath of the Lich King	Y	Y		2009-02-11 23:55:18
6 hours Mon	Student	45 hours	Wrath of the Lich King	Y	Y	No one knov	2009-02-11 23:54:50
15-20 Hours	Programmer/Analyst	50	Wrath of the Lich King	Y	Y		2009-02-11 23:56:06
7	Student	30	Wrath of the Lich King	Y	Y		2009-02-11 23:57:35
30-40	Paintball ref	30	Wrath of the Lich King	Y	Y	I bot mostly	2009-02-11 23:56:52
OVER 9000	I DUNNO LOL	OVER 900 HOU	Wrath of the Lich King	Y	N	IMA CHARG	2009-02-11 23:58:55
50	Military	40+	Wrath of the Lich King	Y	Y		2009-02-11 23:58:47
			Not-Specified	N	N		2009-02-12 00:00:53
10 hours	Student	8 hours	Wrath of the Lich King	Y	Y		2009-02-12 00:00:18
10	student	35	Wrath of the Lich King	Y	Y		2009-02-12 00:00:15
13	Student	37 hours	Wrath of the Lich King	Y	Y		2009-02-11 23:59:09
too many	student	35	Wrath of the Lich King	Y	Y		2009-02-12 00:00:53
6-8	Student / Retail Clerk	20	Wrath of the Lich King	Y	Y	Good luck o	2009-02-12 00:01:08
alot	highschool student	...all of it...	Wrath of the Lich King	Y	Y	HOPE I HAL	2009-02-12 00:01:04
32	Student	25	Wrath of the Lich King	Y	Y		2009-02-12 00:00:55
20-30	student	40	Wrath of the Lich King	Y	Y	comment	2009-02-12 00:02:10
40+	student	40	Wrath of the Lich King	Y	Y	I do have a	2009-02-12 00:02:13
25	Police Assistant/Prope	40	Wrath of the Lich King	Y	Y		2009-02-12 00:00:37
30-45 hours	Student	70	Wrath of the Lich King	Y	Y	nerp... besic	2009-02-12 00:01:47
90000	90000	90000	Classic	Y	N	Over 9000	2009-02-12 00:04:32
5-6	Student	15-20	Wrath of the Lich King	Y	Y		2009-02-12 00:04:38
15-20	Student and cashier	12 hours	Wrath of the Lich King	Y	Y		2009-02-12 00:03:58
12 hours	Studen	30 + hours	Wrath of the Lich King	Y	N		2009-02-12 00:04:13
			Not-Specified	N	N		2009-02-12 00:04:54
5-6	High School Student	40	Wrath of the Lich King	Y	Y		2009-02-12 00:05:12
30-40	Unemployed	None	Wrath of the Lich King	Y	Y		2009-02-12 00:05:31
alot	student	40	Wrath of the Lich King	Y	Y	i like wow	2009-02-12 00:04:54
10	Computer Repair/Sale	44	Wrath of the Lich King	Y	Y	No. lol.	2009-02-12 00:05:30
20	student	25	Wrath of the Lich King	Y	Y		2009-02-12 00:05:52
8-10 weekda	Student	5days * 7hours =	Wrath of the Lich King	Y	Y	My 'ma	2009-02-11 23:59:42
10	Manager	44-50	Wrath of the Lich King	Y	Y		2009-02-12 00:05:50
30	student	75	Burning Crusade	Y	Y	there has be	2009-02-12 00:03:13
14-35	college student	54	Wrath of the Lich King	Y	Y		2009-02-12 00:08:25
10	Student	30	Burning Crusade	Y	Y		2009-02-12 00:08:00
40	Editor	40	Wrath of the Lich King	Y	Y		2009-02-12 00:08:56
10-20	law enforcement	60+	Wrath of the Lich King	N	N		2009-02-12 00:09:20
Use to play 3	Student		Wrath of the Lich King	Y	Y	Vanilla wow	2009-02-12 00:08:00
2-3 currently	Full time student majo	18	Wrath of the Lich King	Y	Y		2009-02-12 00:09:06
varies greatl	Student; Student work	50-60	Wrath of the Lich King	Y	Y	I don't	2009-02-12 00:07:05
About 15	Hairstylist	20-40	Wrath of the Lich King	Y	Y		2009-02-12 00:09:15
50+	Sales Associate	40	Wrath of the Lich King	N	N		2009-02-12 00:10:41

123	49936901e47a2	2009-02-12 00:11:51	2009-02-12 00:11:59	Kael'Tha	United State	Pensacola	80	Male	Gnome	Death Knig	Male	19	4 years
124	499368d014346	2009-02-12 00:09:52	2009-02-12 00:12:14	Dertheroc	America	Jellico	80	Male	Blood Elf	Paladin	Male	16	2 years.
125	4993691907aeb	2009-02-12 00:11:05	2009-02-12 00:12:17	Blackhand	United State	Los Angeles	80	Male	Night Elf	Druid	Male	14	Since rel
126	49936956e099c	2009-02-12 00:12:06	2009-02-12 00:13:28	mug'thol	US	-	80	Male	Blood Elf	Paladin	Male	20	4 years
127	4993686477f9e0	2009-02-12 00:08:04	2009-02-12 00:13:43	vek' nila	usa	washington dc	75	Female	Gnome	Warlock	Male	17	4 years.
128	499368dfd74b17	2009-02-12 00:10:07	2009-02-12 00:14:04	Eonar	US	Portsmouth	80	Male	Blood Elf	Paladin	Male	33	4 years
129	499369b0763d1	2009-02-12 00:13:36	2009-02-12 00:15:19	Light Bringer	USA	Rochester	80	Male	Human	Mage	Male	18	3 years
130	499369448c83b	2009-02-12 00:11:48	2009-02-12 00:15:19	Khadgar	U.S.A.	Des Moines	76	Male	Human	Warrior	Male	19	Since lau
131	49936a2b426da	2009-02-12 00:15:39	2009-02-12 00:18:47	Mug'thol	USA	Sanford, NC	80	Male	Human	Paladin	Male	16	3 years c
132	49936a562b849	2009-02-12 00:16:22	2009-02-12 00:19:41	Twisting Neth	Greece	Thessaloniki	80	Female	Dwarf	Paladin	Female	19	3 years
133	49936ae766eb5	2009-02-12 00:18:47	2009-02-12 00:20:13	bubble fart	my dick	anus	99	Female	Blood Elf	Warrior	Male	49	since rel
134	49936b3fc7db05	2009-02-12 00:20:15	2009-02-12 00:21:08	Cenarion Circ	USA	San Jose, CA	72	Male	Human	Priest	Male	43	4+ years
135	49936afe374991	2009-02-12 00:19:10	2009-02-12 00:21:31	Bleeding Hall	US		80	Female	Night Elf	Priest	Male	18	2 years
136	49936afe374991	2009-02-12 00:21:31	2009-02-12 00:21:40	Bleeding Hall	US		80	Female	Night Elf	Priest	Male	18	2 years
137	49936b37a055e	2009-02-12 00:20:07	2009-02-12 00:21:42	Nagrand	Australia	Geelong	80	Female	Draenei	Priest	Male	28	2 Years
138	49936b43ada9b	2009-02-12 00:20:19	2009-02-12 00:22:12	Twisting Neth	United State	Oklahoma City	80	Male	Tauren	Druid	Male	15	4 years
139	49936b43ada9b	2009-02-12 00:22:12	2009-02-12 00:22:24	Twisting Neth	United State	Oklahoma City	80	Male	Tauren	Druid	Male	15	4 years
140	49936c098111c0	2009-02-12 00:23:37	2009-02-12 00:23:47	Eredar	United States			Not Speci	Not-Specifie	Not-Specifi	Not Specified		
141	49936be4426bb	2009-02-12 00:23:00	2009-02-12 00:23:58	Medivh	US	Denver	80	Male	Orc	Warrior	Male	22	4 years
142	49936bf9bd7ce9	2009-02-12 00:23:21	2009-02-12 00:24:27	Galakrond	US	NYC	80	Male	Human	Death Knig	Male	17	2 years
143	49936c075c1ec	2009-02-12 00:23:35	2009-02-12 00:24:39	Dunemaul	USA	Seattle	80	Female	Undead	Priest	Female	31	4 yrs (Si
144	49936c075c1ec	2009-02-12 00:24:39	2009-02-12 00:24:46	Dunemaul	USA	Seattle	80	Female	Undead	Priest	Female	31	4 yrs (Si
145	49936c098111c0	2009-02-12 00:23:47	2009-02-12 00:25:03	Dalaran	United State	Leesburg	80	Female	Undead	Mage	Male	17	4 years
146	49936c1b39c50	2009-02-12 00:23:55	2009-02-12 00:25:09	Ravencrest	US		80	Male	Human	Warrior	Male	28	3 years
147	49936c3cbde76	2009-02-12 00:24:28	2009-02-12 00:25:12	[redacted]			99	Not Speci	Blood Elf	Hunter	Not Specifi	pe	[redacted]
148	49936c1cca49a	2009-02-12 00:23:56	2009-02-12 00:25:32	Jubei'Th	Australia	Melbourne	80	Female	Draenei	Paladin	Male	20	4 years
149	49936c3cca13f8	2009-02-12 00:24:28	2009-02-12 00:25:48	Khadgar	United State	Milwaukee	80	Not Speci	Undead	Rogue	Male	17	3 Years
150	49936c5d17d92	2009-02-12 00:25:01	2009-02-12 00:25:50	Tichondrius	United State	Baltimore	80	Male	Undead	Death Knig	Male	19	Since Re
151	49936c47f08e29	2009-02-12 00:24:39	2009-02-12 00:26:01	Demon Soul	United State	Des Plaines, Il	80	Male	Undead	Warrior	Male	18	3 years
152	49936c4d5fb5e9	2009-02-12 00:24:57	2009-02-12 00:26:06	Eonar	United State	Ormond Beach	72	Male	Tauren	Druid	Male	18	4 years
153	49936c4137ebb	2009-02-12 00:24:33	2009-02-12 00:26:12	Kalecgos	United State	Cleveland	61	Male	Undead	Rogue	Male	19	
154	49936c428a036	2009-02-12 00:24:34	2009-02-12 00:26:17	Draenor	USA	Girard	80	Male	Blood Elf	Paladin	Male	17	3 Years
155	49936c9ce7ba3	2009-02-12 00:26:04	2009-02-12 00:26:48	Kirin Tor	United State	Atlanta	73	Male	Tauren	Druid	Male	26	2.5 years
156	49936c8c9b045	2009-02-12 00:25:48	2009-02-12 00:26:52	laughing skull	usa	dallas	80	Male	Blood Elf	Death Knig	Male	15	2 years
157	49936c9ce7ba3	2009-02-12 00:26:48	2009-02-12 00:27:31	Kirin Tor	United State	Atlanta	73	Male	Tauren	Druid	Male	26	2.5 years
158	49936cb806481	2009-02-12 00:26:32	2009-02-12 00:27:59	Illidan	United State	Little Rock	80	Male	Orc	Hunter	Male	18	Four Yea
159	49936cc0c3070	2009-02-12 00:26:40	2009-02-12 00:29:02	Aerie Peak	United State	Charlotte	80	Male	Night Elf	Hunter	Male	16	5 to 6 mo
160	49936cd2e3017	2009-02-12 00:27:29	2009-02-12 00:29:15	Dalaran	USA	Chicago	80	Female	Night Elf	Hunter	Male	24	Since Op
161	49936d37bf60d8	2009-02-12 00:28:39	2009-02-12 00:29:33	Uldum	Australia	Adelaide	80	Male	Tauren	Druid	Male	23	Since Ma
162	49936d288b50b	2009-02-12 00:28:24	2009-02-12 00:29:33	Cho'gal	US	Eldersburg	80	Male	Tauren	Druid	Male	16	1 year 5
163	49936d2707307	2009-02-12 00:28:23	2009-02-12 00:29:36	Ravencrest	United State	Chicago	80	Male	Blood Elf	Paladin	Male	16	3 years

50+	Sales Associate	40	Wrath of the Lich King	Y	Y		2009-02-12 00:10:41
30-ish =/	Highschool	8	Wrath of the Lich King	Y	Y		2009-02-12 00:09:52
72	Student	30	Wrath of the Lich King	Y	Y		2009-02-12 00:11:05
30-40	student	20	Wrath of the Lich King	Y	Y		2009-02-12 00:12:06
50	student	30	Wrath of the Lich King	Y	Y	i have a 80b	2009-02-12 00:08:04
18	Telecom Specialist	45	Wrath of the Lich King	Y	Y	My wife play	2009-02-12 00:10:07
10	student	35+	Wrath of the Lich King	Y	Y	Getting bore	2009-02-12 00:13:36
Less than 5	college student	35+	Wrath of the Lich King	Y	Y	I have every	2009-02-12 00:11:48
depends on I	student	most of it	Wrath of the Lich King	Y	Y	Tank 95% o	2009-02-12 00:15:39
30	College Student	40 (class+studyi	Wrath of the Lich King	Y	Y	Probably, I&	2009-02-12 00:16:22
140	none	none	Wrath of the Lich King	Y	N	im over 350	2009-02-12 00:18:47
12-15	programmer	40	Wrath of the Lich King	Y	Y		2009-02-12 00:20:15
24	student	a lot	Wrath of the Lich King	N	N		2009-02-12 00:19:10
24	student	a lot	Wrath of the Lich King	Y	Y		2009-02-12 00:19:10
30+	Retail	15-22	Wrath of the Lich King	Y	Y	Hope you ge	2009-02-12 00:20:07
Too infreque	Student	38ish	Wrath of the Lich King	N	N	I play WoW	2009-02-12 00:20:19
Too infreque	Student	38ish	Wrath of the Lich King	Y	Y	I play WoW	2009-02-12 00:20:19
			Not-Specified	N	N		2009-02-12 00:23:37
10-15	Accountant	50+	Wrath of the Lich King	Y	Y		2009-02-12 00:23:00
Around 20.	Student	Near 30.	Wrath of the Lich King	Y	Y		2009-02-12 00:23:21
10-15	retired :)	none	Wrath of the Lich King	N	N		2009-02-12 00:23:35
10-15	retired :)	none	Wrath of the Lich King	Y	Y		2009-02-12 00:23:35
12+	Student	~40	Wrath of the Lich King	Y	Y	Good luck ^	2009-02-12 00:23:37
20	Software Engineer	40	Wrath of the Lich King	Y	Y		2009-02-12 00:23:55
[redacted]			Classic	Y	N	[redacted]	2009-02-12 00:24:28
10-20	Student	30	Wrath of the Lich King	Y	Y		2009-02-12 00:23:56
12 - 15 Hour	Student	40	Wrath of the Lich King	Y	Y	Good Luck!	2009-02-12 00:24:28
20	College Student		Wrath of the Lich King	Y	Y		2009-02-12 00:25:01
30+	Full time college stude	40+	Wrath of the Lich King	Y	Y		2009-02-12 00:24:39
30	High school Senoir	40	Wrath of the Lich King	Y	Y		2009-02-12 00:24:57
15	Unemployed	0	Burning Crusade	Y	Y		2009-02-12 00:24:33
15-20 Hours	Student	Too many/35	Wrath of the Lich King	Y	Y	I heard you	2009-02-12 00:24:34
20			Not-Specified	N	N		2009-02-12 00:26:04
10	student	alot	Wrath of the Lich King	Y	Y	[redacted]	2009-02-12 00:25:48
20	Veterenary Technichia	40	Wrath of the Lich King	Y	Y		2009-02-12 00:26:04
Seven	Student	~16-18	Wrath of the Lich King	Y	Y	I get sad if I	2009-02-12 00:26:32
about 25	student	35	Wrath of the Lich King	Y	Y		2009-02-12 00:26:40
15	Grad. Student	40	Wrath of the Lich King	Y	Y		2009-02-12 00:27:29
~10	Government data anal	~45-50	Wrath of the Lich King	Y	Y		2009-02-12 00:28:39
10-15?	Student :(	7 hours a day...e	Wrath of the Lich King	Y	Y		2009-02-12 00:28:24
10-30	student	50	Wrath of the Lich King	Y	Y		2009-02-12 00:28:23

164	49936d286deb3	2009-02-12 00:28:24	2009-02-12 00:29:40	Eonar	US	Los Angeles	70	Male	Orc	Hunter	Not Specifi	16	2 years
165	49936d3de338d	2009-02-12 00:28:45	2009-02-12 00:30:12	Archimonde	United State	Philadelphia	70	Male	Undead	Mage	Male	17	2 years
166	49936d4b28211	2009-02-12 00:28:59	2009-02-12 00:30:13	Medivh	United State	Bethel	80	Male	Night Elf	Warrior	Male	14	Since Re
167	49936d3baf3e93	2009-02-12 00:28:43	2009-02-12 00:30:25	Malfurion	USA	Windsor	80	Male	Human	Paladin	Male	16	1 1/2 yea
168	49936d492826e	2009-02-12 00:28:57	2009-02-12 00:30:50	Duskwood	U.S.A.	Altoona	80	Male	Draenei	Shaman	Male	18	2 Years
169	49936d4d2155b	2009-02-12 00:29:01	2009-02-12 00:31:15	Silver Hand	USA	San Francisco	80	Female	Night Elf	Hunter	Not Specifi	31	4 years
170	49936d9beebfe8	2009-02-12 00:30:19	2009-02-12 00:31:39	Aman'Th	Australia	Cairns	80	Male	Human	Warrior	Male	16	Just ove
171	49936dd6a54fc9	2009-02-12 00:31:18	2009-02-12 00:32:35	Staghealm	United State	Woodbridge	80	Female	Draenei	Priest	Female	29	4 years
172	49936ddc5ea26	2009-02-12 00:31:24	2009-02-12 00:33:12	Velen	United State	Tacoma	80	Male	Dwarf	Hunter	Not Specifi	15	1 year
173	49936e06c3a91	2009-02-12 00:32:06	2009-02-12 00:34:10	Akama	Canada	Toronto	46	Female	Night Elf	Druid	Male	21	4 years
174	49936e4c8d36c	2009-02-12 00:33:16	2009-02-12 00:34:39	Bronzebeard	Canada	Toronto	80	Male	Human	Paladin	Male	14	2 years
175	49936e4c8d36c	2009-02-12 00:34:39	2009-02-12 00:34:49	Bronzebeard	Canada	Toronto	80	Male	Human	Paladin	Male	14	2 years
176	49936e6d2cbc4	2009-02-12 00:33:49	2009-02-12 00:34:50	Arthas	United State	Columbus	80	Male	Orc	Hunter	Male	16	3 years
177	49936d2fdea5b2	2009-02-12 00:28:31	2009-02-12 00:35:32	blackhand	usa	berea,ohio	80	Male	Human	Paladin	Male	22	2ish yea
178	49936e9a8a91b	2009-02-12 00:34:34	2009-02-12 00:36:00	Sen'Jin	Unites State	Gig Harbor	76	Female	Blood Elf	Warlock	Female	16	Three ye
179	49936e5fa05858	2009-02-12 00:33:35	2009-02-12 00:36:15	Draka	US	Eugene	80	Male	Night Elf	Druid	Male	20	~1 year
180	49936edaba683	2009-02-12 00:35:38	2009-02-12 00:36:49	the forgotten	united state	baltimore	80	Female	Tauren	Druid	Female	20	2 years
181	49936efc734a53	2009-02-12 00:36:12	2009-02-12 00:37:21	Whisperwind	Canada	Windsor	80	Male	Gnome	Warrior	Male	14	3 years
182	49936ed7be52a	2009-02-12 00:35:35	2009-02-12 00:37:33	Firetree	United State	Madison	80	Female	Troll	Priest	Male	19	4 years
183	49936ece6964d	2009-02-12 00:35:26	2009-02-12 00:37:38	Anvilmar	Canada	Burlington	80	Male	Draenei	Priest	Male	19	2 years
184	49936efba6d347	2009-02-12 00:36:11	2009-02-12 00:37:38	Moonrunner	United State	Elkhart	80	Male	Night Elf	Death Knig	Male	19	August 2
185	49936f024648b2	2009-02-12 00:36:18	2009-02-12 00:37:42	Icecrown	USA	Boston	80	Female	Blood Elf	Paladin	Male	31	4.5 years
186	49936ee60f2a15	2009-02-12 00:35:50	2009-02-12 00:37:56	Blackrock	United State	Kodiak	80	Female	Night Elf	Priest	Male	17	4 years,
187	49936ee5865a6	2009-02-12 00:35:49	2009-02-12 00:38:06	Shadowsong	United State	Spokane	80	Male	Tauren	Druid	Male	18	Three ye
188	49936f2e7fd687	2009-02-12 00:37:02	2009-02-12 00:38:58	Malygos	U.S	Joliet	80	Male	Dwarf	Paladin	Male	19	4 years(i
189	49936f643f6d82	2009-02-12 00:37:56	2009-02-12 00:39:46	Anub'Ar	Germany	Bonn	80	Male	Tauren	Warrior	Male	20	Since rel
190	49936e0eedfec8	2009-02-12 00:32:14	2009-02-12 00:40:08	Proudmoore	Australia	Canberra	80	Male	Human	Paladin	Male	29	4 Years
191	49936ffcaacfb3.	2009-02-12 00:40:28	2009-02-12 00:41:35	shu'halo	usa	brooklyn	80	Female	Night Elf	Druid	Male	31	4 years
192	49936fefb59631	2009-02-12 00:40:15	2009-02-12 00:41:45	Thrall	United State	Raleigh	80	Female	Tauren	Hunter	Female	14	3 years
193	4993700aca7b1	2009-02-12 00:40:42	2009-02-12 00:42:03	Boulderfist	Canada	Kamloops	64	Male	Orc	Death Knig	Male	27	1 year
194	49937013311fd4	2009-02-12 00:40:51	2009-02-12 00:42:05	Tichondrius	United State	Long Beach	80	Male	Draenei	Death Knig	Male	18	4 years
195	49937004cc841	2009-02-12 00:40:36	2009-02-12 00:42:15	Perenolde	USA	Denver	80	Female	Tauren	Druid	Male	28	4 years,
196	49937013311fd4	2009-02-12 00:40:51	2009-02-12 00:43:39	Tichondrius	United State	Long Beach, C	80	Male	Draenei	Death Knig	Male	18	4 years
197	49937013311fd4	2009-02-12 00:43:39	2009-02-12 00:43:50	Tichondrius	United State	Long Beach, C	80	Male	Draenei	Death Knig	Male	18	4 years
198	49937105471aa	2009-02-12 00:44:53	2009-02-12 00:46:29	Moon Guard	Canada	St John's	80	Male	Blood Elf	Mage	Male	15	Three ye
199	499371814e695	2009-02-12 00:46:57	2009-02-12 00:48:05	Terenas	Canada	Toronto	70	Male	Gnome	Warlock	Male	17	2000 hou
200	499371eaa396e	2009-02-12 00:48:42	2009-02-12 00:49:35	Dethecus	US	Columbus	80	Male	Blood Elf	Rogue	Male	17	2 years
201	499371eaa396e	2009-02-12 00:49:35	2009-02-12 00:49:44	Dethecus	US	Columbus	80	Male	Blood Elf	Rogue	Male	17	2 years
202	4993729a0ec3d	2009-02-12 00:51:38	2009-02-12 00:52:51	Gorefiend	America		80	Male	Undead	Warrior	Male	16	2 years
203	4993729eabed5	2009-02-12 00:51:42	2009-02-12 00:52:54	Moon Guard	USA	Newpor News,	80	Female	Blood Elf	Death Knig	Male	34	4 years
204	4993740cd5515	2009-02-12 00:57:48	2009-02-12 00:58:57	Sargeras	Canada	Ottawa	80	Male	Dwarf	Priest	Male	17	3-4 years

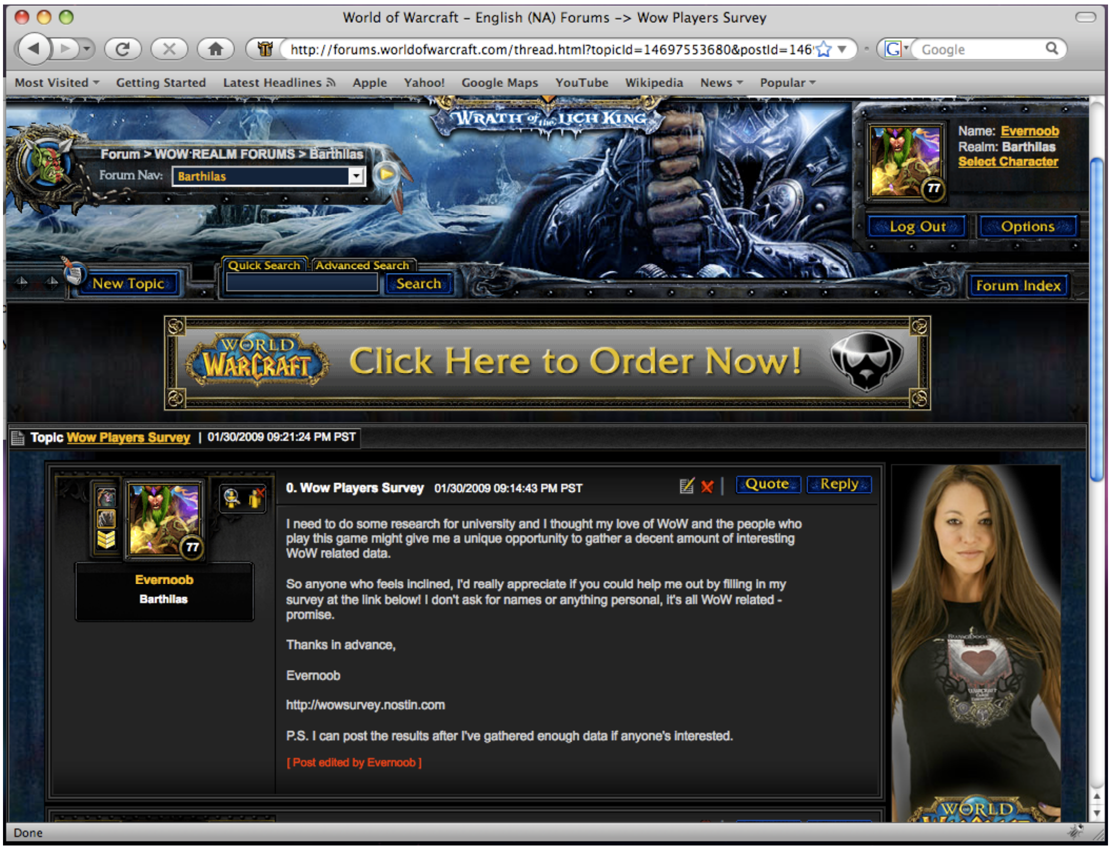
10	student	40	Burning Crusade	Y	Y		2009-02-12 00:28:24
15 hours	student	32	Wrath of the Lich King	Y	Y		2009-02-12 00:28:45
5-8	Student	6	Wrath of the Lich King	Y	Y	Goog Luck	2009-02-12 00:28:59
10	Student (junior)	~35 hours	Wrath of the Lich King	Y	Y		2009-02-12 00:28:43
50	Student	40	Wrath of the Lich King	Y	Y		2009-02-12 00:28:57
40+	Customer Service	40	Wrath of the Lich King	Y	Y		2009-02-12 00:29:01
A lot - not su	Student	Normal school h	Wrath of the Lich King	Y	Y		2009-02-12 00:30:19
20+	Teacher	60+	Wrath of the Lich King	Y	Y	Strange to t	2009-02-12 00:31:18
21-25 hours	Student	40	Wrath of the Lich King	Y	Y		2009-02-12 00:31:24
20	Student	38	Classic	Y	Y	I used to ha	2009-02-12 00:32:06
20 ish	Student	30	Wrath of the Lich King	N	N	Ret pally pw	2009-02-12 00:33:16
20 ish	Student	30	Wrath of the Lich King	Y	Y	Ret pally pw	2009-02-12 00:33:16
37	Student	35	Wrath of the Lich King	Y	Y		2009-02-12 00:33:49
atleast 30	student	25ish? i think	Wrath of the Lich King	Y	Y		2009-02-12 00:28:31
Depends on	High school student	Maybe 30 - 40	Wrath of the Lich King	Y	Y		2009-02-12 00:34:34
0-10	Graphic Designer/Stu	20	Wrath of the Lich King	Y	Y		2009-02-12 00:33:35
12 hours	student	none	Wrath of the Lich King	Y	Y		2009-02-12 00:35:38
120	Student	30 approx.	Wrath of the Lich King	Y	Y		2009-02-12 00:36:12
Around 30 h	Student	Around 40 hours	Wrath of the Lich King	Y	Y	Odd survey.	2009-02-12 00:35:35
20-30	unemployed atm	0	Wrath of the Lich King	Y	Y		2009-02-12 00:35:26
30	Student, Part Time W	N/A	Wrath of the Lich King	Y	Y		2009-02-12 00:36:11
20	Research Scientist	40	Wrath of the Lich King	Y	Y		2009-02-12 00:36:18
1/3 of my life	Student/shitty minimu	40h work, 30 sch	Wrath of the Lich King	Y	Y	Release pla	2009-02-12 00:35:50
56	Student	50	Wrath of the Lich King	Y	Y		2009-02-12 00:35:49
50ish	construction	45	Wrath of the Lich King	Y	Y	dwarfs are c	2009-02-12 00:37:02
50	University Student	30	Wrath of the Lich King	Y	Y		2009-02-12 00:37:56
25	Database Developer	40	Wrath of the Lich King	Y	Y		2009-02-12 00:32:14
15	manager	55	Wrath of the Lich King	Y	Y		2009-02-12 00:40:28
~35	Honors Highschool St	~40	Wrath of the Lich King	Y	Y		2009-02-12 00:40:15
6	Business Analyst	40	Wrath of the Lich King	Y	Y		2009-02-12 00:40:42
			Not-Specified	N	N		2009-02-12 00:40:51
30	programmer	40	Wrath of the Lich King	Y	Y		2009-02-12 00:40:36
7	Student	55	Wrath of the Lich King	N	N		2009-02-12 00:40:51
7	Student	55	Wrath of the Lich King	Y	Y		2009-02-12 00:40:51
50	None.	None.	Wrath of the Lich King	Y	Y	Nope.	2009-02-12 00:44:53
24	student	50	Burning Crusade	Y	Y	i like pie	2009-02-12 00:46:57
20+	Student	35	Wrath of the Lich King	N	N		2009-02-12 00:48:42
20+	Student	35	Wrath of the Lich King	Y	Y		2009-02-12 00:48:42
um around 4	11th grade	more than 40 ho	Wrath of the Lich King	Y	Y	Not really	2009-02-12 00:51:38
10-15	Help Desk Anylist	40-50	Wrath of the Lich King	Y	Y		2009-02-12 00:51:42
15-25 hours	Student	15-20 hours	Wrath of the Lich King	Y	Y		2009-02-12 00:57:48

205	499374abea10f8	2009-02-12 01:00:27	2009-02-12 01:01:36	Sargeras	U.S.	St. Louis	80	Male	Night Elf	Druid	Male	17	4 years
206	499376964c05f2	2009-02-12 01:08:38	2009-02-12 01:09:50	Gurubashi	USA	Columbus	71	Male	Undead	Death Knig	Male	16	2 years
207	4993770884c92	2009-02-12 01:10:32	2009-02-12 01:11:27	Stormrage	US	Greensboro	80	Male	Human	Death Knig	Male	16	2 years
208	4993770884c92	2009-02-12 01:11:27	2009-02-12 01:11:32	Stormrage	US	Greensboro	80	Male	Human	Death Knig	Male	16	2 years
209	499377ddddd76	2009-02-12 01:14:05	2009-02-12 01:15:47	Dath'Re	New Zealar	Auckland	80	Female	Human	Mage	Female	21	A little le
210	49937a262716a	2009-02-12 01:23:50	2009-02-12 01:27:00	Kargath	USA	Goochland	78	Not Speci	Night Elf	Druid	Male	21	3.5
211	49938afba668b8	2009-02-12 02:35:39	2009-02-12 02:36:56	Echo Isles	Australia	Melbourne	80	Male	Human	Mage	Male	23	4 years c
212	49939685a2f448	2009-02-12 03:24:53	2009-02-12 03:26:38	gnomeregan	usa	ashland	71	Male	Gnome	Death Knig	Male	27	one year
213	4993999729d1a	2009-02-12 03:37:59	2009-02-12 03:39:22	Stormscale	Canada	Toronto	80	Female	Draenei	Paladin	Male	27	4 years
214	4993a869bbb76	2009-02-12 04:41:13	2009-02-12 04:43:35	Drenden	United Stat	Dalton	73	Female	Draenei	Hunter	Female	20	3 years
215	4993e17655e0e	2009-02-12 08:44:38	2009-02-12 08:46:33	Nagrand	Australia	Perth	74	Female	Blood Elf	Mage	Male	29	3 years
216	4993e7e6c1816	2009-02-12 09:12:06	2009-02-12 09:13:56	Kil'Jaed	US	San Francisco	78	Female	Human	Mage	Male	22	Since rel
217	499462b346109	2009-02-12 17:56:03	2009-02-12 17:57:37	Moonrunner	USA	Illinois	80	Male	Night Elf	Druid	Male	18	3 years

10	student	n/a	Wrath of the Lich King	Y	Y	I do not nee	2009-02-12 01:00:27
10-15	Student	40	Wrath of the Lich King	Y	Y	Good luck w	2009-02-12 01:08:38
40 maybe	Student	40	Wrath of the Lich King	N	N		2009-02-12 01:10:32
40 maybe	Student	40	Wrath of the Lich King	Y	Y		2009-02-12 01:10:32
10	(taking a year gap fror	40	Wrath of the Lich King	Y	Y		2009-02-12 01:14:05
30	Wal-mart Loader	40	Wrath of the Lich King	Y	Y		2009-02-12 01:23:50
20-30	Paramedic	42	Wrath of the Lich King	Y	Y		2009-02-12 02:35:39
30	chemical process mar	40+	Wrath of the Lich King	Y	Y		2009-02-12 03:24:53
20+	bartender	40	Wrath of the Lich King	Y	Y		2009-02-12 03:37:59
5-10	Stay at Home Wife	Every hour of Ev	Wrath of the Lich King	Y	Y		2009-02-12 04:41:13
30	Farmer	50	Wrath of the Lich King	Y	Y		2009-02-12 08:44:38
1-2	Crap job	20 avg	Wrath of the Lich King	Y	Y		2009-02-12 09:12:06
~14	Freshman in college	~22	Wrath of the Lich King	Y	Y		2009-02-12 17:56:03

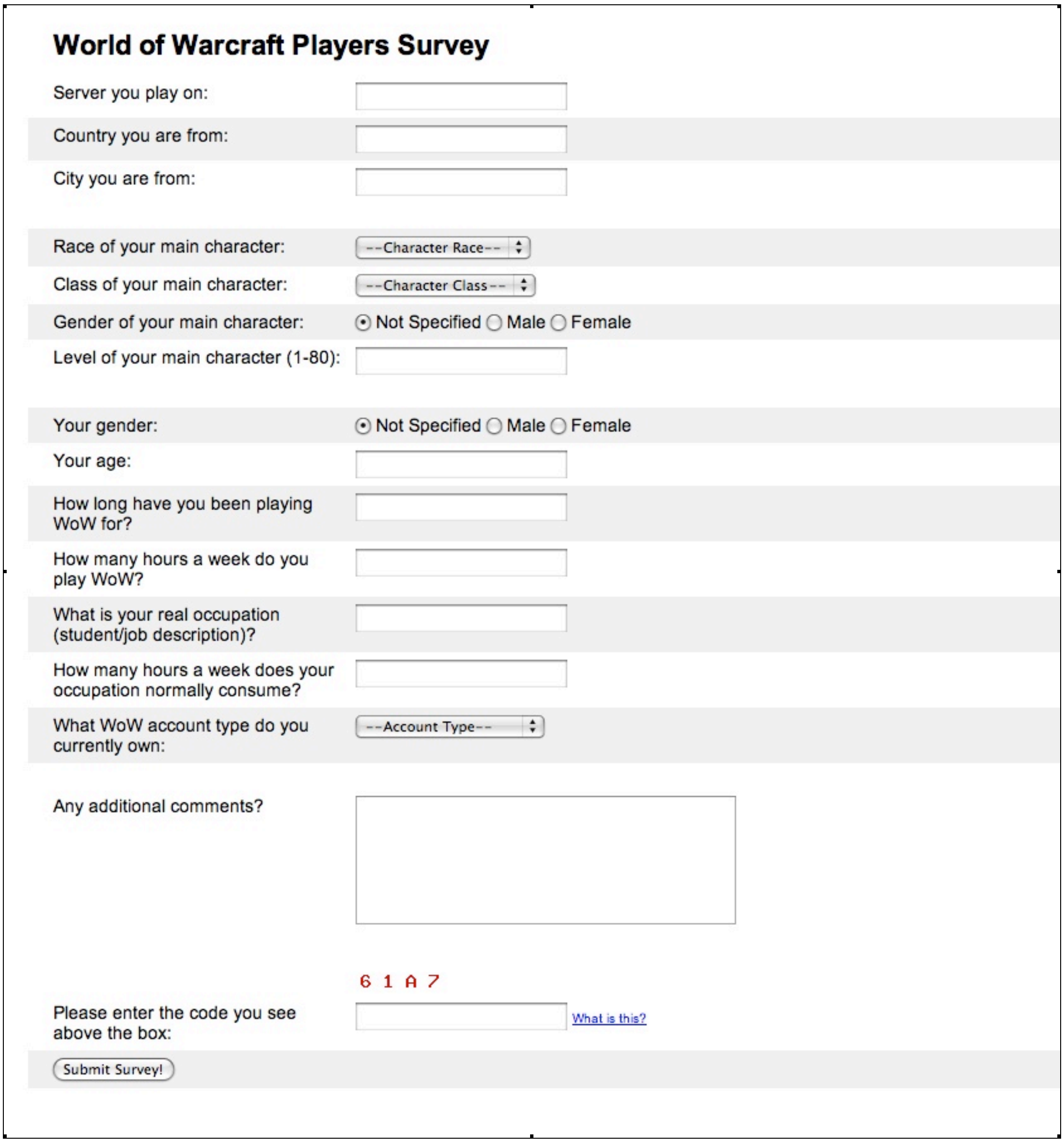
Appendix C – Online Implementation Participants Source

Appendix C shows the World of Warcraft users form, which was the source of gathering participants to test the GARDEN approach. These forums can be perused at http://us.battle.net/wow/en/forum/. There is also a screenshot showing an example of the thread posted on these forums enticing users to participate in the research. The URL for these forums has changed since this experiment was conducted, which is why the URL shown in the screenshot is different from the above.



Appendix D – Online Implementation Source

Appendix D shows the process form used in the testing of the GARDEN process non-conformance detection solution. A working demo of this form is still online and available at http://wowsurvey.nostin.com. A screenshot of this survey is also provided here for reader convenience.



World of Warcraft Players Survey

Server you play on:

Country you are from:

City you are from:

Race of your main character: --Character Race--

Class of your main character: --Character Class--

Gender of your main character: Not Specified Male Female

Level of your main character (1-80):

Your gender: Not Specified Male Female

Your age:

How long have you been playing WoW for?

How many hours a week do you play WoW?

What is your real occupation (student/job description)?

How many hours a week does your occupation normally consume?

What WoW account type do you currently own: --Account Type--

Any additional comments?

6 1 A 7

Please enter the code you see above the box: What is this?

Submit Survey!