

December 2002

Analysis of Stakeholder Concerns with a View to Avoid Organisational Conflict in B2B Systems

Jacob Cybulski

School of Information Systems, Faculty of Business and Law, Deakin University

Follow this and additional works at: <http://aisel.aisnet.org/bled2002>

Recommended Citation

Cybulski, Jacob, "Analysis of Stakeholder Concerns with a View to Avoid Organisational Conflict in B2B Systems" (2002). *BLED 2002 Proceedings*. 4.

<http://aisel.aisnet.org/bled2002/4>

This material is brought to you by the BLED Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in BLED 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Analysis of Stakeholder Concerns with a View to Avoid Organisational Conflict in B2B Systems

Pradipta K. Sarkar

School of Information Systems, Faculty of Business and Law, Deakin University
Australia
PKS1@deakin.edu.au

Jacob L. Cybulski

School of Information Systems, Faculty of Business and Law, Deakin University
Australia
JLCybulski@deakin.edu.au

Abstract

In recent years there has been a remarkable increase in information exchange between organizations due to changes in market structures and new forms of business relationships. The increase in the volume of business-to-business (B2B) transactions has contributed significantly to the expanding need for electronic systems that could effectively support communication between collaborating organizations. Examples of such collaborating systems include those that offer various types of business-to-business services, e.g. electronic commerce, electronic procurement systems, electronic links between legacy systems, or outsourced systems providing data processing services via electronic media. Development and running of B2B electronic systems has not been problem free. One of the most intractable issues found in B2B systems is the prevalence of inter-organisational conflict reported to exist and persists between the participants of inter-organisational electronic networks. There have been very few attempts, however, to prescribe any practical method of detecting the antecedents of such conflict early in B2B development to facilitate smooth construction and the subsequent operation of B2B services. The research reported in this paper focuses on the identification and analysis of antecedent conflict in a joint process involving different organizations in a B2B venture. The proposed method involves identification of domain

stakeholders, capturing and packaging their views and concerns into a reusable form, and the application of captured domain experience in B2B systems development. The concepts and methods introduced in this paper have been illustrated with examples drawn from our study of six web-enabled payroll systems.

1. Introduction

Organisational conflict and its pervasive nature can be observed in any organisation (Schermerhorn Jr., Hunt *et al.* 1997). Such conflict may occur at personal, group, and organizational levels, and it could manifest itself as intra-personal, inter-personal, intra-group, inter-group, intra-organizational, and inter-organizational conflict. Business *stakeholders*, or organisational members participating in a common business process (Freeman 1984), may disagree on matters such as the setting of a group, organizational goals and values, allocation of resources, distribution of rewards, policies, procedures, and task assignments (Putnam and Poole 1987). In the extreme situations, unless it is detected early and dealt with promptly, intra-organisational conflict could lead to express struggle between stakeholders who have incompatible goals, who share scarce resources and compete for few rewards, and who are the source of mutual interference (Hocker and Wilmot 1985).

As B2B systems transcend the boundaries of and are beyond the control of a single enterprise, hence, the degree of complexity inherent in these systems is much higher than in intra-organizational systems. One of the prime aspects contributing to this complexity is the presence of a heterogeneous set of multiple stakeholders. While the most apparent stakeholders in a B2B system include sponsors and adopters, also known as "hubs" and "spokes" in Webster's model of inter-organisational stakeholders (1995), stakeholders also include system developers, telecommunications companies, financiers, industry associations, regulatory agencies (both national, and international), and trade unions. Considering the variety of organisational stakeholders, their competition for market resources, the structure of the market, its inherent processes, and the associated legal-political constraints, B2B systems also suffer from *inter-organizational conflicts* (Chatov 1981). Examples of such conflicts include situations where one organisation actively seeks dominance over other stakeholders in the control of access into the electronic alliance and its resources (Cavaye 1995, Cavaye and Cragg 1995), in its influence over design of software facilitating interaction of the participant firms (Hørlück 1995), or in monitoring and controlling trading relationships with suppliers, who on the other hand, could be deprived of any strategic benefits (Webster 1995). Schermerhorn *et al.* (1997) also exemplify disputes between trade unions and management, between government bodies and organizations who have to comply with their legislation, and, between organizations in the supply/value chain. Unless detected and eliminated, inter-organisational conflict can set

unsurpassable barriers to inter-firm collaboration through a B2B system (Li and Williams 1999).

Conflicts commonly develop over an extended period of time, and they follow a number of recognisable stages (Pondy 1967), i.e. latent accumulation of *conflict antecedents*, feeling or perception of early signs of conflict, manifestation of conflicting behaviour, and the conflict aftermath upon its suppression or resolution. Since conflict antecedents are directly responsible for the development of manifest conflict, the focus of our study is directed towards gaining better understanding of these conflict-conducive conditions, of which there exist three main types, i.e. socio-political, structural and technological (Kumar and van Dissel 1996, Li and Williams 1999). However, regardless of the nature of the conflict antecedents, it is the process of B2B system *requirements determination* that provides an opportunity to identify and eliminate the potential B2B conflicts at the earliest stage of developing technological infrastructure for the collaborating or competing organisations.

In the following sections, we describe the theoretical framework for the detection of antecedent conflict in B2B systems development, which looks specifically at stakeholders, their views and their concerns, and the ability to capture and apply past experience in dealing with the stakeholder issues in a given application domain. We illustrate our framework with examples drawn from web-enabled payroll systems, which we have recently studied in Melbourne (Australia).

2. Framework for Detection of Antecedent Conflict

The methods of identifying system inconsistencies during requirements engineering, and thus avoiding the potential risks and conflicts, include *stakeholder analysis* (Smith 2000) and *viewpoint analysis* (Finkelstein, Nuseibeh *et al.* 1994, Leite 1996, Sommerville and Kotonya 1996). Stakeholder analysis focuses on determining system stakeholders, their interests, influences, their similarities and differences, and their impact on the project. Viewpoints, on the other hand, represent the organisational and developmental knowledge as perceived by individual stakeholders. Such knowledge commonly provides a reference model for the requirements of the newly constructed systems. Comparison of viewpoints leads to discovery of inconsistencies in the stakeholders' understanding of the inter-organisational processes or in the policies of their hosting organisations. The two approaches are complementary, and not surprising, the elements of both methods can be found integrated into a single method in practice, e.g. in Andersen Consulting Foundation Method via the QFD technique and its associated "user context" and "user voice" analysis (Andersen Consulting 1994).

Sommerville and Kotonya (1996) further extend the viewpoint approach to dealing with the potential sources of inter- and intra-organisational conflict by explicitly identifying and analysing stakeholder *concerns*. In general, stakeholder concerns

impose constraints on system requirements and themselves are treated as obligatory requirements for all systems in a given application domain. Compliance with organisational stakeholder concerns is critical to developing systems, which are less likely to cause or support inter-organisational conflict (Easterbrook 1994). In web-based e-commerce systems, stakeholder concerns are usually related to the systems' accessibility, scalability, reliability, performance, security, and attractiveness of the web presentation (Li 2000, Abolhassani 2000). The majority of such general concerns in B2B systems focus on the system effectiveness in facilitating interactions between participating firms. For example, in payroll systems that we have investigated, departmental managers are concerned with the confidentiality of the employee pay records. This concern is commonly attributed to any application in the payroll domain. It is related to the data "access" which is typical for this domain, hence, it should be incorporated as part of a security component in the design of all payroll systems.

Stakeholders' concerns, together with their needs and requirements, are part of domain knowledge and thus can be *reused* across projects in a common domain (Cybulski, Neal *et al.* 1998, Arango 1994, Prieto-Diaz 1990). Research in software engineering suggests that reuse of software artefacts could lead to higher quality of software products and could be more cost-effective, as compared with systems produced from scratch (Tracz 1987, Prieto-Diaz 1990, Basili, Dieter *et al.* 1987). This research is of the view that experience gained by expert designers with projects can also be reused in new projects leading to a significant reduction of stakeholder concerns. In this regard, we suggest to capture and reuse developers experience in dealing with stakeholders, their requirements and their concerns, with a view to reduce the possibility of future conflict as manifested in an implemented B2B system.

Experience in dealing with conflict can be collected in a series of interviews, then be formalised, packaged and stored in an experience repository (Basili, Caldiera *et al.* 1994). Our *conflict experience base* for the web-enabled payroll domain consisted of models and examples of observed situation that led to stakeholders concerns or even express conflict, which we initially categorised as follows (Wood, Wallace *et al.* 1998, Schermerhorn Jr., Hunt *et al.* 1997):

- Task interdependency among stakeholders;
- Scarcity of resources, both tangible and intangible, e.g. funds, equipment, facilities, knowledge, expertise and influence;
- Role ambiguities, e.g. due to the lack of clarity of the stakeholder's responsibilities or poorly defined boundaries of their influence;
- Organizational differentiation, which occurs when the stakeholders follow different time horizons for their projects, have different goals and perspectives, and use distinct language and terminology;
- Structural differentiation, which may exist due to the difference in stakeholders' rules, procedures, and policies; and,

- Unresolved prior conflicts, or residue of past conflicts among the stakeholders.

The overall conceptual framework for our research is shown in Figure 1. The model shows two stakeholders, namely A and B, represented by stick figures. The stakeholders are two organisations in a B2B relationship. The number of stakeholders may vary depending upon their direct participation in a B2B process. The stakeholders experience in dealing with problems associated with some class of shared processes may be generalised and subsequently transferred to other stakeholders in the same application domain – in our case we focussed on the transfer of such expertise to B2B systems developers. In the context of this research, the common application domain was web-enabled payroll and the processes that we investigated involved communication of payroll data in various forms between client organisations, payroll outsourcer, the banks and financial institutions, and finally various government agencies.

As the participating stakeholders may have differing viewpoints on the shared process, there exist a potential of conflict developing in respect to that process. To counter such development, we capture the stakeholder viewpoints by identifying the stakeholder concerns, and subsequently compare these concerns against the patterns collected in a conflict experience base. Should the match occur, we then can use the matching patterns to guide B2B systems developers in the analysis of conflict antecedents, possible renegotiation of process requirements, and adopting a proven solution that is likely to reduce the stakeholders' concerns.

3. Study of Web-Enabled Payroll B2B Systems

In order to select an application domain suitable for our study of B2B systems and subsequently to gain better understanding of the chosen domain, elite interviews (Marshall and Rossman 1989) were conducted with two consultants. These interviews suggested web-based payroll systems as potentially of interest to this research, which was due to the recent adoption of web technology in outsourced and in-house payroll services in Australia, the significant scale of deployed systems, and the “true” B2B nature of such systems.

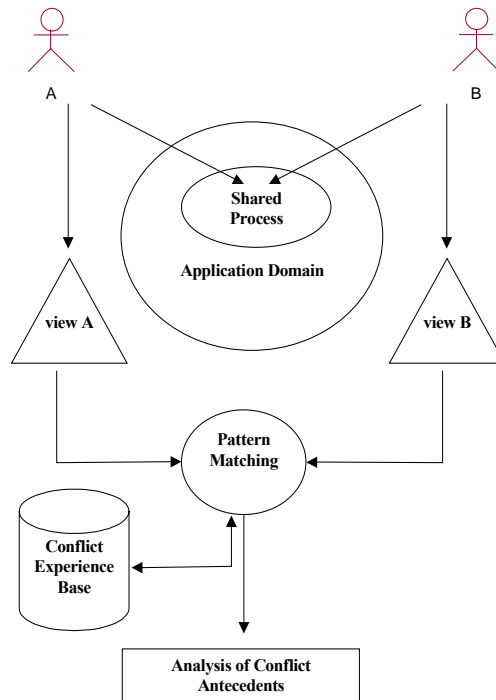


Figure 1: The Model of Conflict Detection in the B2B Systems Development

Subsequently, we have carried out a study that investigated six web-based applications in the payroll domain. Its main aim was to evaluate the proposed method of capturing and using conflict experience. Our method of study comprised systematic collection, analysis, and packaging of design experience related to stakeholder concerns into the conflict experience base. **Figure 2** illustrates all of the conducted activities and provides a methodological framework for our study. It shows the stakeholders (shown as stick figures), activities (ellipses), and the output of each activity (rectangular boxes). A brief description of the methodological steps follows.

3.1 Collection of Conflict Experience

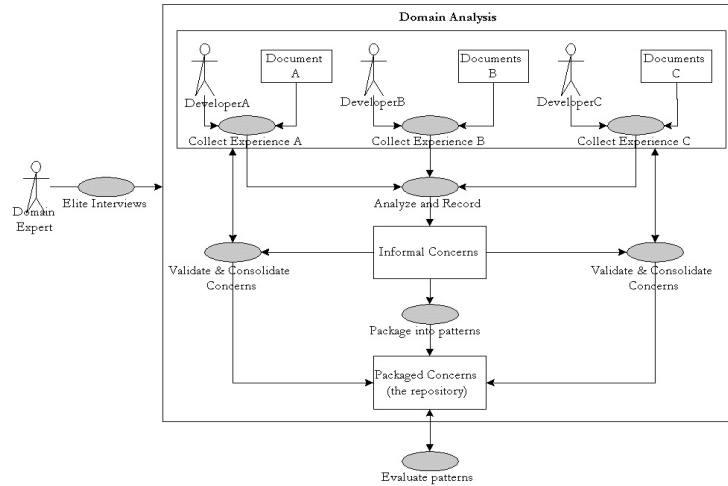


Figure 2: Conducted Study in Capturing Conflict Experience

Experiences in dealing with conflict situations was collected via qualitative interviews with developers, and examination of application documents, which included manuals, requirements and design specification reports, developers’ notes and charts, and importantly, software demos. Some stakeholder concerns were explicitly stated by the interview participants, while others were revealed during the analysis and interpretation of documented data by the researcher.

3.2 Analysis and Recording of Concerns

To analyse the collected data across the entire payroll domain and its multiple systems, we employed a well-known method of feature-oriented domain analysis, as exemplified by the FORM (Kang, Kim *et al.* 1998) and FODA (Kang, Cohen *et al.* 1990) methodologies. Our analysis focussed on discovering stakeholder concerns, their commonalities and variations. **Figure 2** shows the collection of relevant data, and its subsequent analyses, from several applications (i.e. A, B, and C). Each application involved a developer stakeholder (e.g. DeveloperA for application A) and a set of documents associated with the application (e.g. Documents A for application A). As a result of this analysis, we produced a large list of informally stated stakeholder concerns.

3.3 Validation and Consolidation of Concerns

An effective way of packaging experience related to solving frequently occurring problems in a certain domain is known as *patterns* (Buschman, Meunier *et al.* 1996, Rising 1999). A pattern can be thought of as a prescription for identifying and selecting a solution to a non-trivial class of problems. Patterns are known to be particularly effective in sharing and reusing expert knowledge in such applications as architectural design (Alexander 1979), education (Anthony 1996), design of organisational processes (Coplien 1995), software development (Fowler 1997, Buschman, Meunier *et al.* 1996, Gamma, Helm *et al.* 1995) and multimedia construction (Cybulski and Linden 1998, Rossi, Schwabe *et al.* 1997), to name just a few. In our study of six web-based payroll systems in Melbourne (Australia), we found out that patterns are also applicable to capturing stakeholder concerns and resolving antecedent conflicts. As all our patterns reflect developers' experience in dealing with B2B conflicts, we called them *conflict experience patterns*.

Our patterns were formulated according to the most commonly used pattern schema, which consists of a number of distinct components (Gamma, Helm *et al.* 1995, p 3, 6-8), i.e. definition of a *problem* in some well-defined *context*, a set of *forces* preventing effective problem solving, the *solution* that describes a method of overcoming the forces, and finally, the *consequences* of applying the problem-solution in terms of the expected results and trade-offs.

In order to package the collected data into patterns, it was important to validate stakeholder concerns, and to formalise and consolidate them into a pattern form. This was done through a series of structured follow-up interviews with developers. In these interviews, developers were presented with the stakeholder concerns collected across the entire domain to ascertain the relevance of these concerns to their particular organisation, as well as to classify each concern statement as a pattern component, i.e. its problem, force, solution or consequence. Interviewees were also asked to provide some insight into their particular decisions associated with the concern's relevance and its classification. The rich insight information was used to derive the context of each pattern. The consolidation and the relevance ranking were also helpful in identifying similarities and variation between the stakeholder concerns.¹ At this stage we have also revealed some additional previously unknown concerns (as indicated by double arrows between the validation activity and developers in **Figure 2**).

3.4 Packaging Concerns into Patterns

The structured interviews enabled the packaging of concerns into patterns as per the classification of their features. For each feature, there may exist a set of high-level patterns, each of which was comprised of a number of lower-level patterns. These

¹ Variations are those features, which are not shared by all applications.

packaged experiences of developers formed our experience repository. The packaging of patterns was the final step in the analysis of the domain.

3.5 Evaluation of Collected Patterns

Collected conflict experience patterns can be used to better understand a domain of application and to disseminate development experience between B2B system developers. This aim has been achieved thus far. However, an equally important goal of the experience base repository is to directly support development of new B2B products and services, by aiding the identification and analysis of stakeholder concerns over the issues of organisational conflict. To this end, the effectiveness of patterns in related design tasks also needs to be evaluated. Such evaluation is currently being conducted and is undertaken on the basis of analysing past design solutions against concerns captured in domain patterns, as well as applying the collected patterns to solving new problems in controlled environment. This step will allow feedback on the usefulness of the collected patterns and will provide opportunities of further patterns enhancements.

4. Findings and Discussion

After stakeholder concerns have been collected, analysed, reconciled and packaged in patterns, we found out that, unlike traditional software applications, web-based B2B payroll systems involved great many stakeholders who had no knowledge of the inner workings of the payroll process, e.g. the members of the payroll's client firm. This aspect led to concerns that were typical of B2B data processing systems in-general, and it clearly drove the design of such applications. Ease-of-use and optimum online user-support were found to be key aspects of web-based payroll applications and workflow-based B2B applications in particular. The research also discovered that the main motivator behind the adoption of the web-based B2B application by payroll providers was to reduce overheads associated with the keying in data obtained from paper-based documents, such as employee personal data, timesheets, leave requests, etc. By delegating data entry to the "source" (i.e. to its client side), payroll could devote more attention to its core payroll tasks. Clients and employees in the client firm were motivated by the ability to have access, "anywhere" and "anytime", to relevant pay and employment history records at their fingertips, without having to directly contact the payroll provider.

The feature-oriented analysis enabled us to identify groups of stakeholder concerns, which we found to be unique for the payroll domain. The concerns have, thus, been classified into several categories (e.g. Data Entry & Validation, Online processes - see **Figure 3**). Each category (Security, Data Validation, etc. - in the figure),

represents numerous informal concerns derived either directly from the participants and the application documents, or interpreted from the collected data. Stakeholder concerns were uncovered on data entry and validation issues. All payroll providers were keen on ensuring that the data entered by clients and their employees were in accordance with the business rules, set forth in their contractual agreement, as well as data integrity rules. Being aware of the fact that the data will be entered in remote locations and possibly, in the absence of those who are familiar with the applications, payroll providers embedded these rules into the web interfaces to prevent data entered “incorrectly” from being processed. This, however, raised concerns from the clients and their employees, who dreaded the idea of data rejections and being overwhelmed with error messages while working with the applications. According to the conflict and stakeholder theories, the antecedents of conflict between the payroll providers and their clients, over the web-based application, are implicit in these concerns that they have voiced. Clients and their employees were also concerned about the transparency of the payroll process once all online documents had been submitted.

Some of the participants have already been followed-up with the structured interviews. Some of these interviews are still in progress, and expected to be completed by mid 2002. The concerns have been drafted as statements, and participants are being asked whether it is a problem, a situation that complicates the solution to the problem, or the solution itself, as well as some insights into the background of the phenomenon. This is in line with the form of a pattern (problem, context, forces, solutions). Two such patterns have been presented later in this section (in **Napaka! Vira sklicevanja ni bilo mogoče najti.** and **Napaka! Vira sklicevanja ni bilo mogoče najti.**).

4.1 The Method

Our method not only enabled us to conduct a thorough analysis of the web-based (B2B) payroll application domain, but it also allowed pattern discovery, or mining, based on the knowledge of multiple systems in a single domain and experience of direct competitors. This has not been reported so far in the patterns literature, where the pattern mining activities are customarily conducted using collaborative groups or in-depth qualitative interviews confined to a single application (DeLano 1997, Rising 1998). By conducting multiple case studies, we were able to uncover data from developers of “rival” applications. In this regard, any other means would have been difficult, if not impossible. Moreover, the qualitative mode adopted by us was instrumental in producing rich analyses of the obtained data.

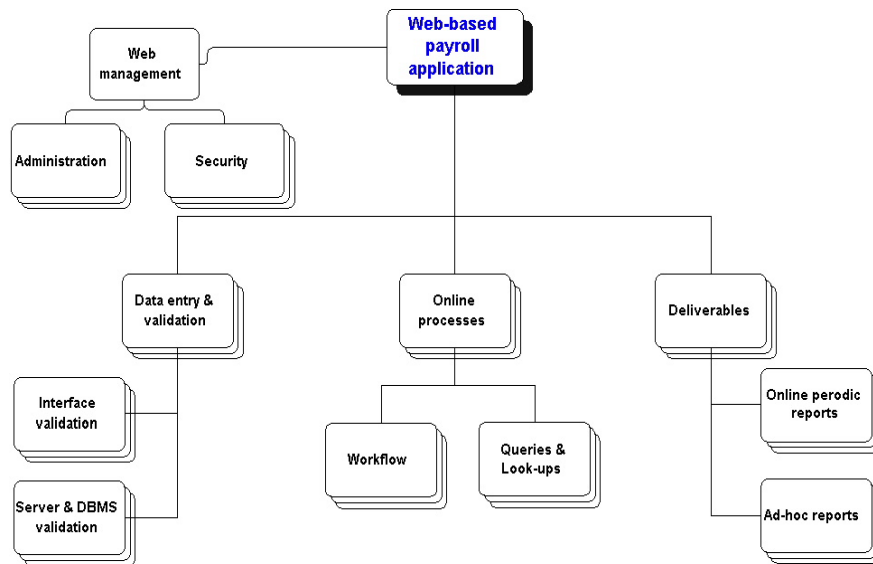


Figure 3: Classification of Domain Concerns

4.2 The Patterns

From our analysis of multiple yet competing applications in the domain, we were able to discover patterns of stakeholder concerns. Data collection took place via interviews with six Melbourne (Australia) organisations, which provide payroll services over the web. Based on the interviews in those organisations, stakeholder concerns have been classified into a number of similarity groups, and illustrated in **Table 1**. Here, for the sake of brevity, we show only stakeholder concerns related to data validation procedures, and collected from three organisations, codenamed O1, O2 and O3.

Table 1: Stakeholder Concerns

Concerns (T represents concerns present in organisation O1, O2 or O3)	O1	O2	O3
Clients are not supposed to enter timesheets of new employees without informing the payroll company.	T		
Clients and their employees may not be familiar with data integrity rules.	T	T	T
Clients request training in entering data into web forms in accordance with data integrity rules.	T	T	T
Not possible to ensure data integrity on each and every field on a web-based form.	T	T	T
Clients need to be informed of the errors in data entry, precisely.	T	T	T
Clients may not remember each and every feature of data integrity introduced during training sessions.	T	T	T

Concerns (T represents concerns present in organisation O1, O2 or O3)	O1	O2	O3
<i>Clients may delegate data entry tasks to employees who received no training.</i>	T	T	T
Individual employees may not be familiar with business rules.		T	T
Requires constant management of web servers on the part of payroll providers.	T	T	T
Redundant or inaccurate data reduces the quality of data stored on payroll databases	T	T	T
In the absence of the supervisor, another manager needs to be authorised for this purpose.	T		T
If data is not entered into a field marked with *, UI will flash error message "All field with an * require data to be entered".		T	T
Time entry is by hour (hours entered) or by start/finish times (in which case, hours can't be entered).		T	T
Employee will not be allowed to submit timesheets unless the Completed checkbox for each payday to be submitted is checked.			T
Days in the future can have time entered and saved, but submission will not be allowed.		T	T
If a supervisor rejects the timesheet hours for a particular employee without entering any comments, application will not allow the record to be saved.		T	T
Web server contains no data, so main validation undertaken by DBMS application.	T	T	T
An exception will be generated if there are irregularities with data integrity, by the DBMS. The issue then has to be addressed manually.	T	T	T
DBMS application will reject any data entry that does not correspond to an existing record in the database (DBMS application checks to see that the name and employee number on a timesheet match those on the corresponding record).	T	T	T
If at this stage, there are irregularities not spotted by the DBMS (business-related errors), the records are allowed to be rectified, additions, modifications, and deletions can be made.	T	T	T

We found out that each stakeholder, namely the payroll provider, the client firm, and the client's employees, held concerns over their involvement in work processes via a web-based system, the compliance with and investigation of which was critical in moderating conflict. Although, the repository of patterns we developed is by no means exhaustive (and it continually evolves, instead), it has shed significant light on the various conflict antecedents likely to exist in B2B payroll systems, as well as on the solutions that can be employed to prevent their manifestation. In **Napaka! Vira sklicevanja ni bilo mogoče najti.**, a pattern, *The Automated Input Validator*, has been illustrated. The substance of the pattern was derived from the data generated and qualitatively analyzed according to the proposed method, presented in **Figure 2**.

The data, on which this pattern is entrenched on, can be referred to in **Table 1**. One of the main concerns of any IT service provider is the integrity of data. Likewise, payroll providers are also concerned with the matter, and this forms the *problem* component of the pattern. The forces, solution, and consequences, applicable to the problem, were initially generated from the data gathered, and then cast into statements. These statements were then presented in a structured interview to the participants. On the basis of their responses, we were able to validate the pattern components. Our study of the context in which the problem existed revealed that the training given by payroll providers to their clients was not a sufficient measure

to ensure that the latter will always use the web application in accordance with data integrity rules. The context of the problem is formed by the concern of the payroll provider, which is that “clients may not remember each and every feature of data integrity introduced during training sessions”, identified as a concern for all three organisations in **Table 1**. The forces which drove the solution to the problem in different directions were actually the concerns of all three groups of stakeholders, namely the payroll providers, the clients, and their employees. These concern “statements“ were deduced during data analysis, and subsequently, had undergone the process of validation and consolidation. This revealed that some of these statements were actual forces to the particular problem and context. For example, the statements, “clients and their employees may not be familiar with data integrity rules“, and “clients may not remember each and every feature of data integrity introduced during training sessions”, in **Table 1**, were identified as forces when we returned to the interview participants and presented these statements to them. Hence, the statements appear as forces (numbers 1 and 3) in the pattern presented in **Napaka! Vira sklicevanja ni bilo mogoče najti.** To accomodate these concerns, the embedding of data integrity rules into the application was found to be a optimum solution, even though it could have some drawbacks. Ignoring any of these forces would have brought forth a solution which was capable of causing dissatisfaction in one of the stakeholders, thereby increasing the likelihood of conflict. Often the negative consequence of a pattern forms the problem and context of a subsequent pattern. One such consequence, in (2), is dealt with in the next pattern, i.e. *Precise Errors* (see **Napaka! Vira sklicevanja ni bilo mogoče najti.**), which suggests that the rejection of incorrect data, enforced by data integrity rules embedded into the web application, should be accompanied by the popping up of precise error messages on screen. The flashing of error messages solves the problem in light of the concerns of clients as well as the payroll provider itself. The consequences of the application of the pattern, were also gathered from the concern statements.

Table 2: Sample Pattern

Pattern Name	The Automated Input Validator
Problem	How to ensure the integrity of data entered by clients into the database through the web interface?
Context	Even though training provided by the payroll outsourcer will familiarize client users with the various features of the web application, it may not sufficiently ensure data integrity on its own, due to various reasons, which are described in the forces.
Forces	<p>Clients may not be familiar with the data integrity rules of the application.</p> <p>Payroll staff may not be present while clients are entering data. In such situations, complex data-entry procedures may force clients to consult the on-line Help of the application, which they may be averse to.</p> <p>Clients may receive training in proper data-entry, but still enter records in violation of data integrity rules. Moreover, client contacts and administration staff may attend training sessions, but the actual data-entry task might be assigned to someone employed as a casual/temporary staff. Thus, training may not guarantee that clients follow the rules.</p>

	The data entered by clients through the web application is directed to the database held by the payroll legacy system. In view of this, redundant or inaccurate data entered by clients may result in poor data management by the payroll company.
Solution	<i>Embedding data integrity rules into the application</i> Data integrity rules could be enforced as non-functional requirements or constraints into the applications. For example, each time a client enters an employee record incorrectly or more than once, the application should reject the data.
Consequences	Embedding data integrity rules into the application will further enforce integrity by preventing data inaccuracy and redundancy. Rejecting data without informing clients of what aspects of the data were entered incorrectly, will cause frustration with the usage of the application. The embedding of data integrity rules into every field on a web form requires significant development hours spent by the company's IT staff.
Known Uses	Most data processing systems, both web and non-web, have business and data integrity rules embedded into them.

Our studies revealed that these problems prevailed in the majority of web-based payroll systems. However, it was the identification of the stakeholder concerns that can lead to the reuse of relevant solutions. Thus, if the problems and the contexts pertaining to a new application in the domain match with that of the corresponding patterns in the repository, we will have an idea of what stakeholder concerns may be present implicitly as well as how the solutions of the patterns can be adapted and applied in the new situation. Since, patterns represent system developmental experience in a manner that is aimed at its reuse, and thus such patterns should assist in the identification of antecedent conflicts as well as formulating a solution to minimize the manifestation of conflict.

Table 3: Another Sample Pattern

Pattern Name	Precise Errors
Problem	How to effectively inform client users of erroneous data entry?
Context	This is a follow –up to the Automated Validator pattern, whereby the embedded data integrity rules reject incorrect data entry.
Forces	Clients need to be informed of the errors. Web applications should minimize the need for clients to consult the payroll company with regards to routine usage via conventional modes of communication. Vague error messages or error statements that are not specific enough could cause confusion and subsequent frustration with the usage of the application.
Solution	Error messages should clearly indicate which fields on a web form were entered incorrectly.
Consequences	Precise error messages inform users not only of the incorrectness of the data entered, but also which fields were entered incorrectly. Even though the client user is informed of the fields entered incorrectly, no (online) suggestions, to rectify the errors, are provided.
Known Uses	Warning messages generated by well-designed applications, both web and non-web.

It should be noted that the Known Uses section of the patterns was not gathered empirically, but rather was based on the authors' experience with general applications development.

5. Future Work

In accordance with the method, our work necessitates evaluation of the effectiveness of the patterns in the repository in communicating (and teaching) and reusing the packaged experience. The evaluation will take place in the form of a controlled experiment, whereby participants, in this case, developers, would be presented with a series of case studies describing requirements for a proposed web-based payroll application, and aimed at their identification of developmental risks from the statements of requirements. The case study will be provided by a domain expert. The material will be based on an actual project that had been completed.

6. Summary and Conclusion

Inter-organisational conflict poses a serious problem for the development of business-to-business (B2B) electronic systems. The difficulty in dealing with such a conflict is two-fold. In the first instance, it is the conflict intractability, which is due to our ineffectiveness in identifying conflict antecedents. Secondly, it is our inability to determine and reconcile the incompatible views and concerns of stakeholders interacting across the inter-organisational networks.

As there exist few practical solutions to dealing with this situation, we thus proposed a novel method of predicting the possibility of inter-organisational conflict. Our method focuses on recording and collecting developers experience in dealing with conflict antecedents across a certain application domain. Should such experience be properly generalised, formalised and packaged, it can then be reused in new projects in the same domain. In our work, it is the stakeholders' concerns that are the basis of our experience base. They are captured from multiple domain sources in the form of patterns and they represent possible stakeholder viewpoints that may occur in some future projects. The conflict experience patterns are used to guide B2B system developers in determining inconsistencies of stakeholder viewpoints and concerns, and they provide typical solutions to alleviate such inconsistencies, reduce stakeholder concerns, and minimise the possibility of manifest conflict developing later in the system implementation or its operation.

We conducted a study of six payroll systems provided by six competing organisations. To arrive at a collection of useful patterns, we had to devise a novel pattern mining procedure, based on a series of exploratory, consolidating and

validating in-depth interviews and qualitative data analysis. In the process, we have discovered our method to be effective in collecting and packaging stakeholder concerns. Using a feature-based domain analysis, we have also discovered the similarities and variations in the stakeholder concerns, which enabled us to construct the classification of domain concerns. Such classification is useful in dealing with new domain concerns by instantiating them into high-level or low-level patterns.

The proposed method is a significant advancement in detecting and preventing inter-organisational conflict in B2B systems. Although purely manual at this point of time, the method is simple and practical. At the same time, we believe that the method can be easily automated, by assisting business analyst in recording stakeholder concerns, in the classification of concern features and their packaging into patterns, in pattern indexing and their matching with stakeholder viewpoint, etc. Such an automated system could then be incorporated as an integral part of industry-standard requirements management systems.

References

1. Abolhassani, M. (2000): "Reviewing the Requirements of Traditional E-Commerce". in *CollectR*.
2. Alexander, C. (1979): *A timeless way of building*. New York: Oxford University Press.
3. Andersen Consulting (1994): *FOUNDATION Methods Version 2.0*, Manual V 9.0, Andersen Consulting, Arthur Andersen & Co.
4. Anthony, D.L.G. (1996): "Patterns for Classroom Education", in *Pattern Languages of Program Design*, N.L. Kerth (Editor). Addison-Wesley Publishing Company: Reading, MA. p. 391-406.
5. Arango, G. (1994): "Domain analysis methods", in *Software Reusability*, W. Schafer, R. Prieto-Diaz, and M. Matsumoto (Editors). Ellis Horwood: London. p. 17-49.
6. Basili, V.R., G. Caldiera, and H.D. Rombach (1994): "The Experience Factory", in *Encyclopedia of Software Engineering*. John Wiley & Sons, Inc. p. 469-476.
7. Basili, V.R., R. Dieter, J. Bailey, and B.J. Joo (1987): "Software Reuse: A Framework for Research". in *the 10th Minnowbrook Workshop on Software Performance Evaluation*. Blue Mountain Lake, NY.
8. Buschman, F., R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal (1996): *Pattern-Oriented Software Architecture: A System of Patterns*: John Wiley & Sons.

9. Cavaye, A.L.M. (1995): "The sponsor-adopter gap - Differences between promoters and potential users of Information Systems that link organizations". *International Journal of Information Management*. **15**(2): p. 85 - 96.
10. Cavaye, A.L.M. and P.B. Cragg (1995): "Factors contributing to the success of customer oriented interorganizational systems". *Journal of Strategic Information Systems*. **4**(1): p. 13 - 30.
11. Chatov, R. (1981): "Cooperation between government and business", in *Handbook of Organizational Design*. Oxford University Press: Oxford. p. 409-437.
12. Coplien, J.O. (1995): "A Generative Development-Process Pattern Language", in *Pattern Languages of Program Design*, J.O. Coplien and D. Schmidt (Editors). Addison-Wesley. p. 204.
13. Cybulski, J.L. and T. Linden (1998): "Composing Multimedia Artefacts for Reuse", in *Pattern Languages of Program Design*, N. Harris, F. B., and R. H. (Editors). Addison-Wesley Longman (1999). p. 461-488.
14. Cybulski, J.L., R.D.B. Neal, A. Kram, and J.C. Allen (1998): "Reuse of Early Life-Cycle Artifacts: Workproducts, Methods and Tools". *Annals of Software Engineering*. **5**: p. 227-251.
15. DeLano, D.E. (1997): "Patterns Mining", in *Best Practices: A Patterns Handbook*. SIGS Books Inc.: New York. p. 87-96.
16. Easterbrook, S. (1994): "Resolving Requirements Conflicts with Computer-Supported Negotiation In Requirements Engineering", in *Requirements Engineering: Social and Technical Issues*, M. Jirotko and J. Goguen (Editors). Academic Press: London. p. 41-65.
17. Finkelstein, A., B. Nuseibeh, and J. Kramer (1994): "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification". *IEEE Transactions on Software Engineering*. **20**(10): p. 760 - 773.
18. Fowler, M. (1997): *Analysis Patterns*. Massachusetts: Addison-Wesley Publishing Group.
19. Freeman, R.E. (1984): *Strategic Management: A Stakeholder Approach*. Boston: Pitman.
20. Gamma, E., R. Helm, R. Johnson, and J. Vlissides (1995): *Design patterns : elements of reusable object-oriented software*. Reading, Mass.: Addison-Wesley.
21. Hocker, J.L. and W.W. Wilmot (1985): *Interpersonal Conflict*. 2nd ed: William C. Brown, Dubuque, IA.
22. Hørluck, J. (1995): "User participation in the development of Interorganizational Systems". in *the 18th Information Systems Research Seminar in Scandinavia*.

23. Kang, K.C., J.A. Cohen, W.E. Hess, and A.S. Peterson (1990): *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, CMU/SEI-90-TR-21, ADA 235785, Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA.
24. Kang, K.C., S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh (1998): "FORM: A feature-oriented reuse method with domain-specific reference architectures". *Annals of Software Engineering*: p. 143-168.
25. Kumar, K. and H. van Dissel (1996): "Sustainable collaboration: Managing conflict and cooperation in Interorganizational Systems". *MIS Quarterly*. **20**(3): p. 279 - 300.
26. Leite, J.C.S.P. (1996): "Viewpoints on Viewpoints". in *the 2nd International Software Architecture Workshop (ISAW-2) and the International Workshop on Multiple Perspectives in Software Development (Viewpoints '96) on SIGSOFT '96 Workshops*, p. 285-288.
27. Li, F. and H. Williams (1999): "New Collaboration between Firms: The Role of Interorganizational Systems". in *the 32nd Hawaii International Conference on System Sciences*. Hawaii.
28. Li, X. (2000): "Perspectives in Engineering Web-Enabled Electronic Commerce Systems". in *Pacific Asia Conference in Information Systems (PACIS)*. Hong Kong, p. 444-491.
29. Marshall, C. and G.B. Rossman (1989): *Designing Qualitative Research*: SAGE Publications Inc.
30. Pondy, L.R. (1967): "Organizational Conflict: Concepts and Models". *Administrative Science Quarterly*. **12**: p. 269-320.
31. Prieto-Diaz, R. (1990): "Domain Analysis: An Introduction". *Software Engineering Notes*. **15**(2): p. 47-54.
32. Putnam, L. and M. Poole (1987): "Conflict and Negotiation", in *Handbook of Organizational Communication: An Interdisciplinary Perspective*, L. Porter (Editor). p. 549-599.
33. Rising, L. (1998): "Patterns Mining", in *Handbook of Object Technology*. CRC Press.
34. Rising, L. (1999): "Patterns: A Way to Reuse Expertise". *IEEE Communications Magazine*. **37**(4).
35. Rossi, G., D. Schwabe, and A. Garrido (1997): "Design Reuse in Hypermedia Application Development". in *8th ACM Conference on Hypertext Technology (HyperText '97)*. Southampton, Inglaterra: ACM Press.
36. Schermerhorn Jr., J.R., J.G. Hunt, and R.N. Osborn (1997): "Conflict and Negotiation", in *Organization Behavior*. John Wiley & Sons, Inc. p. 377 - 398.

37. Smith, L.W. (2000): "Project Clarity Through Stakeholder Analysis". *CROSSTALK: The Journal of Defense Software Engineering*: p. 4 - 9.
38. Sommerville, I. and G. Kotonya (1996): "Requirements Engineering With Viewpoints". *BCS/IEE Software Engineering Journal*. **1**(11): p. 2 - 26.
39. Tracz, W. (1987): "Software Reuse: Motivators and Inhibitors". in *COMPCON S'87*, p. 358-363.
40. Webster, J. (1995): "Networks of collaboration or conflict? Electronic data interchange and power in the supply chain". *Journal of Strategic Information Systems*. **4**(1): p. 31 - 42.
41. Wood, J., J. Wallace, R.M. Zeffane, D.J. Kennedy, J.R. Schermerhorn Jr., J.G. Hunt, and R.N. Osborn (1998): "Conflict and Negotiation", in *Organization Behavior: An Asia-Pacific Perspective*. Jacaranda Wiley Ltd.: Brisbane. p. 578-605.