EVALUATING PROCEDURAL ALTERNATIVES. A CASE STUDY IN E-VOTING

Volha Bryl[1], Fabiano Dalpiaz[1], Roberta Ferrario[2], Andrea Mattioli[3], Adolfo Villafiorita[3]

[1] University of Trento, DIT,
via Sommarive 14, Povo (TN) 38050, Italy,
{bryl,dalpiaz}@dit.unitn.it
[2]LOA,ISTC-CNR,
via alla Cascata 56C, , Povo (TN) 38050, Italy,
ferrario@loa-cnr.it
[3]FBK-irst,
Via Sommarive, 18, Povo (TN) 38050, Italy,
{amattioli,adolfo}@itc.it

Abstract –This paper describes part of the work carried out within the ProVotE project and describes the approach we are taking in order to provide both precise models of the electoral processes of an electronic voting, and mechanisms for documenting and reasoning on the possible alternative implementations of the procedures to support the provincial elections of 2008. In particular, the approach is based on defining an alternating sequence of models, written using UML and Tropos. The former is used to represent the electoral processes (both existing and future), while the latter is meant to provide design rationale for the taken decisions about the future procedures. The choice is made evaluating the available alternatives against non-functional requirements with the help of Tropos goal analysis technique.

1. – Introduction

Art. 84 of PAT (Autonomous Province of Trento) Law 2/2003 promotes the introduction of forms of e-voting for the next provincial elections (to be held in 2008). To actuate the law, the Province is sponsoring the ProVotE project, that has the goal of providing a smooth transition to the new technologies. The project develops along different lines, among which is the process/logistical line: it aims at defining the procedural, organizational, and normative framework that will regulate an electronic election.

Electoral procedures involve different organizations, several people over periods of months, and have strict security and traceability requirements. This paper describes the approach we are taking in order to provide precise models of the electoral processes, while, at the same time, providing mechanisms for documenting and reasoning on the possible alternative implementations of the procedures to support the elections of 2008. In particular, in order to cope with the complexity of the domain, we defined a methodology based on the UML for modelling the electoral processes and we show how we are complementing such a methodology with the use of Tropos approach [2], in order to reason about process alternatives and therefore provide means to trace choices in devising the electoral “to be” processes and understand their rationales.

Broadly speaking, the idea of integrating UML and Tropos is not new (see e.g. [16]). In the literature, however, most of the efforts have been directed towards the combination of the two approaches to provide a uniform methodology to support software development (e.g. start from early requirements in Tropos and move to UML when it is time to do the “concrete” design). Our approach differs in the sense that UML and Tropos are used independently to achieve different and complementing goals; we stick to UML as a notation to formalize procedures and processes (both “as is” and “to be”), whereas Tropos is used to explain the passage from the “as is” (paper voting) procedures to the “to be” (electronic voting) process. In particular, Tropos allows to depict on the same model the different alternative ways of implementing the “to be” processes, together with the non-functional requirements each alternative helps to or prevent from satisfying. The goal-oriented features of Tropos allow to reason about possible alternatives and thus provide a visual way of formalizing why the specific ways of implementing the to be procedures was chosen.

In Sect. 2 we will briefly present the project under which scope this work has been developed; in Sect. 3 the main features of our proposal are explained, while in Sect. 4 these same features are illustrated in more details with the help of an example. Finally, in Sect. 5 we draw the conclusions and sketch some possible future developments.

2. – The Scenario: e-Voting and ProVotE

2.1 – The ProVotE Project and Motivations

ProVotE has the goal of ensuring a smooth transition to e-voting in Trentino, eliminating risks of digital divide and providing technological solutions which support, with legal value, the phases ranging from voting to the publication of the elected candidates.

The project includes partners from the public administration (Provincia autonoma di Trento, Regione Trentino/Alto-Adige, Consorzio dei Comuni Trentini, Comune di Trento, IPRASE), research centers and academia (IRST, Faculty of Sociology of the University of Trento, Fondazione Graphitech), and local industries (Informatica Trentina) and is co-led by the Electoral Service of the Autonomous Province of Trento and by IRST. Project leadership by the Public Sector, in our opinion, among other advantages, helps tackling the issue of potential conflicts of interests by private industries, see e.g. [14].

The project is multi-phased and is organized in various lines of activities which strictly interact. For instance, in the first phase of the project, some functional and non-functional requirements of the e-voting prototype were built with a strict round-trip between the sociological and the technological line, with the normative line ensuring compatibility with the Italian electoral laws. See [20,3] for more details and [17] for some considerations related to the sociological aspects of e-voting.

Various trials have been conducted to assess the results of the first phase of the project. The trials have had the goals of testing prototypes, evaluating acceptance by citizens, ease of use, etc. So far more than 11.000 citizens have tried the systems, either with experimental value (in four trials conducted in parallel to local elections) or with legal value (election of the representatives of the students in a local high school, involving about 1.000 students).

For the second phase of the project, which will lead to a large-scale introduction of the new voting system, aspects related to procedures, organization, processes become more relevant, as they will serve both as the basis for the deployment of the solution and for the definition of the laws that will govern the electronic election.

With respect to scope, population, and participation, ProVotE is among the largest, if not the largest, e-voting project in Italy.

2.2 – Voting Procedures in Italy and e-voting Experimentations

Simplifying both on the law and on the procedures for the sake of presentation, voting in Italy happens as follows:

1. **Identification and registration of the voter.** At the polling station the voter is usually required to show his/her ID card and the electoral card. If the name of the voter is present in the electoral list of the polling station, the voter is registered, the electoral card stamped, and the voter is admitted to voting.

2. **Casting a vote.** The voter is given a ballot and a pencil and is shown a cabin where the vote can be cast in secrecy. Secrecy is both a right and a duty. The Italian law and procedures are aimed at ensuring that a voter cannot make his/her vote manifest to other people.

At the end of the voting day, the ballot boxes are opened and the counting procedure starts:

3. **Counting.** Votes are counted and the results tabulated in special registers.

4. **Transmission of the results.** When all the ballots have been tabulated, the results are transcribed in various paper documents and transmitted to the offices responsible of aggregating all the data.

5. **Sum and proclamation of the elected representatives.** All the data coming from the different polling stations are counted and seats assigned according to algorithms defined by the law. Data are then made available to the general public.

Various experimentations have been conducted in Italy to introduce new technologies in the polling stations. The largest trial, so far, was sponsored by the central government, and concerned a system for automating steps 3 and 4 above. The system, operated by specially appointed technicians, was installed in 47 precincts at the last European elections and repeated at the last political elections (2006). Little, however, is known about the results of the experimentation. See [10] for some more details.

Proper e-voting experimentations (i.e. including step 2) have been conducted at the local level, usually on a small scale, in experimentations which seem to have had little continuity and/or on which information is scarce. We mention San Benedetto del Tronto (2000), trials sites in Avellino (2001), Campobasso (2001), Cremona (2002, 2006), Ladispoli (2004), Specchia (2005) [5,6]. Other experimentations have been conducted in Valle D'Aosta, Friuli Venezia Giulia, and Milan.

3. – Transition to Electronic Elections

The introduction of new technologies in the polling stations not only changes the way in which votes are cast, but also roles and responsibilities, often in subtle ways (see e.g. [14]). For instance, the introduction of voting machines may change the tools polling officers and representatives of the parties can use to verify the tabulation of data (think for instance of voting machines with no printed trails, in use in some countries). In such a scenario, to maintain the same security/verifiability requirements of a paper election, it may be necessary to introduce various changes to the procedures (e.g. allow the parties and polling officers to test the machines long before the election; provide ways to verify what software is installed on the machines used during the election day).

To mitigate the risk of creeping security "holes" in the electronic procedures, it was decided to provide extensive modelling of processes. The model of the existing procedures provides a baseline for the definition of the new procedures, which describe the electoral process after the introduction of electronc procedures. Basic requirements for the system "to be" are to ensure the same security level of paper elections, to deal with new threats introduced by electronic systems, and to introduce as few changes as possible in the way of voting.

The modeling of the current electoral processes has been performed by devising a specific methodology [12], based on UML, to support an analyst in modelling the applicative domain. The use of UML, in our case, was an essential requirement for various reasons, among which: expertise, tool support and ease of understanding by the domain experts. Furthermore, the definition of the methodology allows performing (semi)automated analysis on the models. Among the supported functions, there is the possibility of extracting information on which actors are responsible for which artifacts produced in an election. The modeling of the "to be" procedures is a more complex activity, because there are different ways of modifying the existing procedures to support an electronic election. The definition of the exact voting procedure to be followed, should take into consideration not only the basic requirements of the system "to be", but also other non-functional requirements (e.g. economicity, efficiency, etc.), and should be based on mechanisms to weigh and evaluate the different choices.

The UML, however, is weak in providing means of describing alternatives and, therefore, the methodology devised in [12] falls short in providing ways to describe the *why* of the transition from the "as is" to a specific "to be". Hence there is a need to complement the UML modelling with some other approach more suited to face these issues. Our proposal is to fill the gap that is left after the application of the UML approach with Tropos [2], an agent-oriented software development methodology, which is requirements driven, i.e. it is based on concepts used during early phases of requirements analysis process. The main entities that populate models in Tropos come from the *i** modelling approach [21], and are actors, goals and dependencies (among actors and among goals); Tropos models both humans and information systems as networks of interdependent actors endowed with goals.

There are some features of Tropos that make it suitable to solve the problems left open by the UML modelling activity. Namely, in Tropos the analysis of system requirements starts from modelling the organizational environment, that is, identifying the stakeholders, their strategic goals, and social relations between them, which preexist the software system. Thus, Tropos modelling helps to to understand and motivate the changes that should incur to the organizational structure and procedures when the software is introduced. On the contrary, the UML model of the system "to be" shows *how* the voting scenario changes with respect to the paper-based system, but it cannot explain *why* such changes have been introduced.

Moreover, one of the analytical tools Tropos suggests is goal analysis [9], that is put forward by modelling goal dependencies, namely, positive or negative contribution a goal can have to the achievement of another, and goals decomposition into subgoals, that can be either an and-decomposition (all the connected subgoals must be fulfilled in order to fulfill the root goal) or an or-decomposition (the subgoals are alternatives: it is enough to fulfill one of them to achieve the parent goal). Some works, e.g. [9,11], use goal analysis to model the choice among "to be" alternatives by representing functional alternatives with or-decomposition, and then analyzing their contribution to non-functional requirements, which are represented as softgoals (which are goals for which it is not straightforward to determine whether they have been achieved or not, e.g. a goal of *having a secure system*). Such kind of analysis helps to understand which choices favored more the satisfaction of a requirement.

Finally, an extension of Tropos, called Secure Tropos [8], has been proposed, which specializes the Tropos dependencies in more security specific relations, such as trust and delegation, within the same framework. This is also relevant with respect to the present work, as security concerns are crucial for voting scenarios.

Given all the features mentioned above, Tropos is a good candidate to complement the UML modeling; but how does the integration of the two modeling approaches take place concretely? The idea is that of keeping each modeling approach to do just what is best suited for, namely modelling *processes* on the one hand (UML), and doing *goal driven* reasoning on the other (Tropos). Thus the UML models provide an exact snapshot of the procedures (independently from the motivations for which they have been devised in a specific way), while, at the same time, Tropos helps to maintain track of the reasons for any change we had to introduce to support electronic elections. From a technical standpoint, this translates into an approach which produces an alternating sequence of UML and Tropos models. In particular, UML is used to model both "as is" and "to be" processes, while Tropos is used in between to reason about design alternatives with a twofold purpose:

1. to provide a rationale for the solutions adopted for the implementation of the system "to be", by modelling possible alternative ways of accomplishing a goal;

2. to explore trust and security issues related to the e-voting process.

The results of the analysis allow, in turn, to modify the existing UML models and devise the new procedures that meet the requirements stated in the Tropos model. The steps described above are then iterated as needed. In this paper our focus is on modeling and analyzing functional alternatives, with security and trust analysis being among the future work directions.

Ideally speaking, every solution in the system "to be" should be taken after having accurately explored all the alternative possibilities. In this case, the role of the Tropos modelling is that of a visual tool that gives support to the people involved in this decision-making by providing them with a general overview of the choices under consideration, so that they could explore all the available alternatives prior to choosing a solution. In practice decisions often emerge from informal discussions and are constrained by stringent legal requirements. In these cases, given the involvement of different stakeholders, the Tropos modelling is useful as it helps to *model and document the motivation* behind the choices. Finally, even after a solution has already been chosen, once that all the alternatives are represented, it comes out that some alternatives not previously considered suit better the requirements. Thus, Tropos modelling can also be seen as a *validation* tool for the choices made.

The next question is how the elements of a Tropos model are chosen. If the main purpose of the model is that of exploring, evaluating and eventually motivating choices between different alternative ways of accomplishing a goal with respect to a list of non-functional requirements, the methodological questions amount to the following two:

1. how are the different alternatives singled out?
2. how are the requirements that provide the reference for evaluation selected?

The first question can be rephrased as follows: how to transform well established procedures based on physical support, like pencils, sheets of paper, cardboard boxes, etc. in practices based on an electronic support? The possible alternatives are constrained in many ways and these constraints come from several dimensions: *technological, legal* and *social*. The main source for the formulation of the alternatives have been the stakeholders of the project: interviews were conducted with the development team that raised technological issues, other interviews took place with the representatives of the Electoral Service of the Province, who were mainly concerned with the compliance with the provincial legislation regulating elections.

With regard to the second point, namely, choosing the right requirements for reference during evaluation of alternative choices, several sources of requirements were considered. Such requirements as maintainability or cost concerns, have mainly been taken from the Software Engineering literature (for some references see [18] and [4]); these represent properties that are desirable for any information system. Security requirements, such as confidentiality, integrity, availability, etc. (see, e.g., [4], Chap. 7), are particularly relevant in the e-voting scenario, since it is crucial that the system is not vulnerable, e.g. it is not possible to manipulate the results or to associate votes to particular electors.

For a number of more domain specific requirements, such as non-traceability of votes or minimal change to the existing legislation, we took an inspiration from existing work, such as, for instance [19,7]. Finally, a very specific requirement that is peculiar of this very project and that comes from the main project objective is the smooth transition from the old paper system to the e-voting. This objective brings with it a very stringent requirement, which is compliance with the existing PAT voting legislation [1]. This is a requirement that is important for several stakeholders (like, for instance, legislators, but also common citizens), as changing the law is a (politically and bureaucratically) complex and time consuming process. Moreover, the closer the new procedures are to the old ones, the less people involved in such procedures have to be instructed and the lower is the probability of mistakes.

4. – An Example from the Case Study

In this section we will illustrate the Tropos goal modeling activity of the above presented approach with the help of an example, which regards one activity performed after elections are finished, namely counting the votes. The example concerns only one phase of the whole process, the closing procedures, and abstracts away those details that are irrelevant for the purpose of the example. Moreover, we only report the goal modeling phase here because of both the space limitations and the existence of an extensive description of modelling the "as is" voting procedures in [12].

The case study shows how alternative choices are modelled and then evaluated and validated with respect to the non-functional requirements the e-voting system should meet. In the diagram in Figure 1, Tropos modelling notation [21] is used, with goals represented as ovals, and non-functional requirements (softgoals) as clouds. For a softgoal there are no clear-cut criteria of whether it is achieved or not, we can only say that a goal/softgoal contributes positively or negatively to the satisfaction of another softgoal, which is graphically represented as an arrow with "+" or "-" on it, respectively. Goals could be decomposed into or- or and-subgoals; the former type of decomposition is represented in the diagram.

A number of choices should be made when defining the e-based counting procedures. These choices are validated against three groups of requirements: (i) the ones which come from the e-voting domain, such as the need to provide secrecy of voting, to avoid traceability of votes, and to minimize the changes to the existing legislation; (ii) "standard" system/software engineering requirements, such such as maintainability and cost; (iii) security requirements, such as confidentiality and secure data transfer.

Figure 1 presents the alternative choices for the counting procedure, which are analyzed against the non-functional requirements belonging to the three above mentioned groups. The next paragraphs describe and motivate the details of Figure 1 and draw some conclusions based on goal analysis.

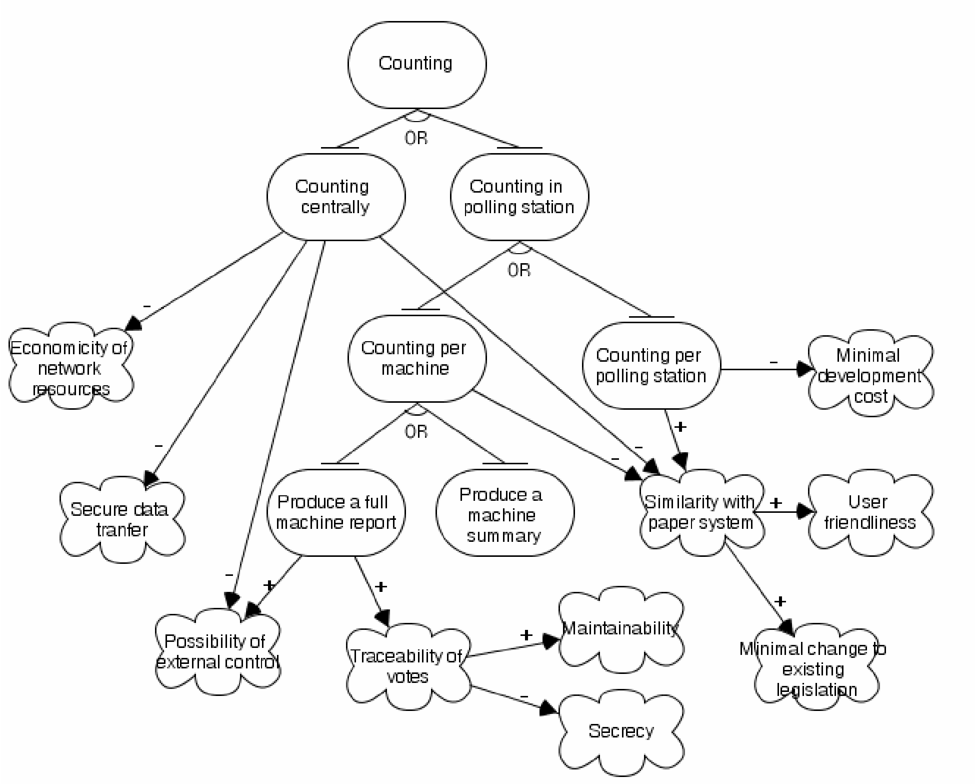
Counting
OR
Counting centrally
Counting in polling station
OR
Economicity of network resources
Counting per machine
Counting per polling station
Minimal development cost
OR
Secure data tranfer
Produce a full machine report
Produce a machine summary
Similarity with paper system
User friendliness
Possibility of external control
Traceability of votes
Maintainability
Secrecy
Minimal change to existing legislation
+
-

Figure 1: Counting: reasoning about alternative choices

At the end of the voting day the counting procedure begins: results are tabulated locally (*counting in polling station*), or the raw data can be sent to the Electoral office (*counting centrally*), where tabulation is performed. The latter choice has many drawbacks, as it is shown on the diagram. For instance, counting the ballots centrally means that unprocessed electronic ballots (e.g. each vote cast) is transferred to the Electoral office; this contributes negatively not only to the *economicity of network resources* (it decreases network availability throughout the day), but also to *secure data transfer* (since, for instance, it increases the time available for possible attacks). Other drawbacks of central counting are the reduction of *possibility of external control*, because political parties representatives are limited in their ability to control the fair conduction of elections. Moreover, it does not comply with the existing counting procedure and legislation (negative contribution to *similarity with paper system*), as in paper-based system ballots are processed in each polling station and only after that they are forwarded to the Electoral office. Thus, goal analysis highlights the drawbacks of this possibility, and ballots are hence counted locally in each polling station.

As far as, for the reasons of system availability, there are several voting machines in each polling station, a number of alternatives should be considered. Namely, *counting in polling station* could be performed either separately for each machine (*counting per machine*), or the data from all machines are to be aggregated and only then processed (*counting per polling station*). The latter alternative complies with the existing paper-based counting procedure ("+" towards *similarity with paper system* softgoal), but requires additional effort to develop the data aggregation mechanism (“-” towards *minimal development cost*). Counting the results separately on each machine is the alternative adopted so far and used during the trials, because of technical concerns such as the easiness of the recovery from errors (an error could be traced back to a specific machine), or development cost (there is no need to introduce a central processing point in each polling station). However, this choice differs from the existing counting procedures meaning that considerable changes to the existing legislation are necessary. This can be even more problematic if we decide to make the results per each machine available (*full machine report* vs. *machine summary* in the diagram). The point here is that, according to the existing electoral law, no one should be able to know partial results, but only the aggregated result per polling station. Having the full machine report available contributes positively to *the traceability of votes*, which introduces the following conflict: on one hand, it makes it easier to trace and recover from errors (“+” towards *maintainability*); on the other hand, the possibility to associate a vote to the machine violates the *secrecy* of voting process making it easier to trace one's vote, which is a confidential piece of data.

5. – Conclusions

In this paper we have presented a modelling approach based on the integration of UML and Tropos. The integration exploits complementary features of the two modelling approaches and allows to maintain both an operational view of the voting procedures and a visual approach to evaluate choices in designing the electronic processes “to be“.

The approach, whose definition has been motivated and driven by a specific need of the ProVotE project, is not restricted to the application domain and we believe it should be easily applicable to other business process re-engineering contexts.

Future works develop along different lines. From the UML point of view, extensions of the tools to support automated analysis are a top-priority. From the Tropos point of view, as already mentioned, we plan to build a trust and delegation Secure Tropos model, which will be aimed at performing a security check over the chosen solutions. Moreover, other improvements, which desirability this experience has highlighted, can be obtained just by improving the Tropos notation. For instance, in the present Tropos model goals are conceived of as independent, while in real world they are very often constrained (they must be achieved in a certain sequence, the achievement of one can cause or prevent the achievement of another, etc.); the possibility of expressing these constraints will significantly improve the power of the approach.

Acknowledgement

The work carried out in this paper is being developed within ProVotE and MOSTRO projects, both sponsored by Provincia Autonoma di Trento. The work of the first author has also been partially funded by the EU Commission through the SERENITY project.

References

[1] Testo unico delle leggi regionali sulla composizione ed elezione degli organi delle amministrazioni comunali. DPReg. n.1/L, 1 febbraio 2005.

[2] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An agent-oriented software development methodology. *JAAMAS*, 8(3):203-236, 2004.

[3] L. Caporusso, C. Buzzi, F. Giolo, P. Peri, and F. Sartori. Transition to electronic voting and citizen participation. In R. Krimmer, editor, *Electronic Voting 2006*, pages 191-200, 2006.

[4] L. K. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Publishing, 2000.

[5] Comune di San Benedetto del Tronto, MET Informatica. Preliminar report on the electronic voting experimentation. In Italian. Available at http://www.comune.san-benedetto-del-tronto.ap.it/ePoll/r100.html.

[6] E-Poll. Electronic polling system for remote voting operations. Available at http://www.e-poll-project.net/.

[7] EAC. Voting systems performance and test standards, 2002. Available at http://www.eac.gov/election_resources/vss.html.

[8] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modeling Security Requirements Through Ownership, Permission and Delegation. In *Proc. of RE'05*, pages 167-176. IEEE Press, 2005.

[9] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani. Reasoning with Goal Models. In *Proc. of ER'02*, pages 167-181, 2002.

[10] Governo Italiano. *European Elections 2004, Automated Counting of the Votes.* In Italian. Available at http://www.governo.it/GovernoInforma/Dossier/voto_conteggio.

[11] D. Gross and E. S. K. Yu. From non-functional requirements to design through patterns. *Requirements Engineering*, 6(1):18-36, 2001.

[12] A. Mattioli. Analisi dei Processi Elettorali in ambito di voto elettronico per le Elezioni in Provincia di Trento, 2006.

[13] G. Grau, X. Franch and N.A.M. Maiden. A Goal Based Round-Trip Method for System Development. In *Proc. of* REFSQ'05. June 13-14, 2005. Porto, Portugal.Pages:71-86.

[14] M. McGaley and J. McCarthy. Transparency and e-voting: Democratic vs. commercial interests. In *the International Workshop on Electronic Voting in Europe*, 2004.

[15] R. T. Mercuri and L. J. Camp. The code of elections. *Communications of the ACM*, 47(10):53-57, 2004.

[16] J. Mylopoulos, M. Kolp, and J. Castro. UML for agent-oriented software development: The Tropos proposal. In *Proc. of UML'01*, pages 422-441, London, UK, 2001. Springer-Verlag.

[17] A. Ostveen and P. van den Besselaar. Security as belief - user's perceptions on the security of electronic voting systems. In *the International Workshop on Electronic Voting in Europe*, 2004.

[18] I. Sommerville. *Software engineering (7th ed.)*. Addison-Wesley, 2004.

[19] Venice Commission - European Commission for Democracy Through Law (2004). Report on the compatibility of remote voting and electronic voting with the standards of the Council of Europe adopted by the Venice Commission. Available at http://venice.coe.int.

[20] A. Villafiorita and G. Fasanelli. Transitioning to eVoting: the ProVotE project and Trentino's experience. 2006. *In Proc. of* EGOV-06.

[21] E. S.-K. Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1995.