UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

EVALUATING PROCEDURAL ALTERNATIVES IN AN E-VOTING DOMAIN: LESSON LEARNED.

Volha Bryl, Roberta Ferrario,
Andrea Mattioli and Adolfo Villafiorita

March 2007

Technical Report # DIT-07-005

Evaluating Procedural Alternatives in an e-Voting Domain: Lesson Learned

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

[1] University of Trento, DIT,
via Sommarive 14, Povo (TN) 38050, Italy,
bryl@dit.unitn.it
[2] LOA, ISTC-CNR,
via Solteri 38, Trento 38100, Italy,
ferrario@loa-cnr.it
[3] ITC-irst,
via Sommarive, 18, Povo (TN) 38050, Italy,
{amattioli,adolfo}@itc.it

Abstract. The ProVotE project aims at actuating art. 84 of law 2 – 5/3/2003, which promotes the introduction of e-voting in Trentino for the next provincial elections (2008). To provide a smooth transition to the new way of voting ProVotE takes a multi-disciplinary approach that develops along different lines, among which, sociological, normative, organizational, and technological. Carrying out an election is a critical activity that involves several people of different organizations over a period of time that spans months. This paper describes part of the work carried out within the organizational/logistical line of the ProVotE project and describes the approach we are taking in order to provide precise models of the electoral processes of an electronic voting, while, at the same time, providing mechanisms for documenting, reasoning on the possible alternative implementations of the procedures to support the elections of 2008. In particular, the approach is based on defining an alternating sequence of models, written using UML and Tropos, that allow to document the existing electoral processes and, at the same time, that are used to reason, evaluate, and choose possible alternative implementations of the electronic voting processes.

Keywords: e-voting, Tropos, UML, non-functional requirements

1 Introduction

Art. 84 of PAT (Autonomous Province of Trento) Law 2/2003 mandates the introduction 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 the process/logistical line, that 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 part of the work carried out within the process/logistical line of the ProVotE project and 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, 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 usage of Tropos, in order to reason about process alternatives and therefore provide means to trace choices in devising the electoral "to be" processes.

Broadly speaking, the idea of integrating UML and Tropos is not new (see e.g. [15]). In the literature, however, most of the efforts have been directed toward 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 the UML and Tropos are used independently to achieve different and complementing goals. In particular we stick to the UML as the notation to formalize procedures and processes (both "as is" and "to be"), whereas Tropos is used at the *meta* level to provide a formal link between the "as is" and the "to be" processes. The goal-orientation features of Tropos allow to reason about possible alternatives and thus provide a visual way of formalizing why, among all the possible alternatives, we choose specific ways of implementing the "to be" procedures.

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 the examples. 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 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 (ITC-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 ITC-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. [13].

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 laws. See [19, 3] for more details and [16] 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 1000 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 the 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.

2.3 Modelling Electoral Procedures in the UML

Electoral laws and procedures have strict security and traceability requirements aimed at ensuring that frauds are extremely difficult to be undetected. To fulfill such goal they define chains of responsibility and delegation, mechanisms for monitoring procedures and mutual controls. Thus, for instance, in Italy, even though the results of an election are available the next day, the confirmation and officialisation of the results requires the Public Administration to perform a strict series of checks, that can take up to a month after the election day.[4]

The introduction of new technologies in the polling stations not only changes the way in which we cast votes, 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 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, we decided therefore to provide extensive modelling of processes. The model of the existing procedures provides a baseline for the definition of the new procedures, that have to be devised so that, to the best of our possibilities, the following requirements are met:

- changes are as minimal as possible: we do not want to alter in "significant" ways how elections are conducted;

[4] Usually most of the issues raised involve identifying and dealing with material errors of polling officers during the election day.

– the same security level of a paper election is maintained: that is, in an electronic election we can at least identify and mitigate the same "attacks" of a paper election;
– the new threats introduced by electronic systems are dealt with by specifically devised procedures and checks.

To do so, we decided to provide detailed models of the current electoral processes and devised a specific methodology, based on the UML, to support functional analyst in modelling the applicative domain, while keeping uniform quality standards. 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.

Among the advantages of the definition of a methodology, we can mention the possibility of performing (semi)automated analysis on the models. In particular, we used VisualParadigm as the modelling tool and exploited the API it offers to implement various extensions to perform custom analysis. Among the functions we support, there is the possibility of extracting information on what actors are responsible for what artifacts produced in an election (see [12] for more details).

3 Transition to Electronic Elections

Not surprisingly, there are different ways of modifying the existing procedures ("as is") in order to define procedures that support an electronic election ("to be") and satisfy the constraints mentioned above. Thus, the definition of the exact procedure to be followed, among all the possible alternatives, should take into consideration other requirements (e.g. economicity, efficiency, etc.) and be based on mechanisms to weigh and evaluate the different choices.

However, the UML is weak in providing means of describing alternatives and, by extension, the methodology we devised falls short in providing ways to describe the *why* of the transition from the "as is" to a specific "to be". Hence the 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.

Tropos [2] is an agent-oriented software development methodology, which is requirements driven. The main entities that populate models in Tropos come from the *i** modelling approach [21], and comprise actors, goals and dependencies (among actors and among goals); it reflects a mentalistic approach, thus it 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.

Firstly, Tropos, differently from other similar methodologies, covers the very early phases of requirements analysis, in which the analysis is centered on the organizational environment where the software must operate, thus on the social relations that preexist to the software and on the changes and improvements that should incur to the environment when the software is introduced. On the contrary, the UML model of the system "to be" shows how the voting scenario changes after the introduction of the information system, but it cannot explain why such changes have been introduced. Usually the rationale lies in some features that the designer wants the new system to possess, i.e. in the requirements of the new system. Thus, the application of Tropos to this phase has the purpose to explain the reasons why the system is introduced.

Moreover, one of the analytical tools Tropos suggests is goal analysis [9], that is put forward by modelling goal dependencies – whether the achievement of a goal contributes to the achievement of another or it prevents it – and their decomposition into subgoals, that can be 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, 20], use goal analysis to model the choice between alternatives by combining or-decomposition and contribution between goals and – by representing non-functional requirements as soft goals[5] – help understanding 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 modelling; but how does the integration of the two modelling approaches take place concretely?

The idea is that of keeping each modelling 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, the UML model is used at the "object" level to model processes. After the UML model is given, Tropos is used at the "meta" level, 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.

[5] Soft goals are goals for which it is not straightforward to determine whether they have been achieved or not. In this respect they are similar to non-functional requirements, like security: there is no trivial way to decide if a system is secure or not.

The results of the analysis allow, in turn, to modify the existing UML model and devise the new procedures that meet the goals stated in the Tropos model. The steps described above are then iterated as needed.

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 things are not so well methodologically applicable. Often these decisions emerge from informal discussions and are constrained by stringent legal requirements. However, even in these cases, given the involvement of different stakeholders, the Tropos modelling is useful as it documents and represents the motivation behind the choices. Finally, it can happen that, even in cases in which some 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, the Tropos modelling can also be seen as a validation tool for the choices made.

So far we have seen the potentialities of the Tropos model; but, methodologically, how are the elements of the model 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 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.

The second point, namely, the choice of the right requirements for reference during evaluation of alternative choices, was not an easy task either. Some of these requirements are strictly inherent to the information system itself, its features and functionalities, and are relative to its technical performance. Other requirements come from the interaction of the system with the environment and are more related to the effects of the system on the organizational level.

The more general requirements, as maintainability or cost concerns, have mainly been taken from the Software Engineering literature (for some references see [17] and [4]); these represent properties that are desirable for any information system. Other requirements are more specifically oriented to security issues, such as confidentiality, integrity, availability, etc. (see, e.g., [4], Chap. 7). These are particularly relevant in the e-voting scenario, since it is crucial that the system is not vulnerable, otherwise there could arise major concerns, like results being manipulated, or votes being associated to particular electors.

As mentioned above, security is not the only concern for the implementation of an e-voting system. For other more specific requirements, as non-traceability of votes or minimal change to the existing legislation, we also took inspiration from existing work, such as, for instance (see, e.g., [18, 7]).

Finally, there is a very specific requirement that is peculiar of this very project and that comes from the main objective of the project itself, namely 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 legislation on voting in the PAT (that is contained in [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 the people involved in such procedures have to be instructed and the lower the probability of mistakes is going to be.

4 Case Studies

In this section we will illustrate the approach described above with the help of two examples. In these examples two activities, performed after elections are finished, are concerned, namely, counting and transferring the processed data to the Electoral office. Both examples concern modelling alternative choices, and their evaluation and validation with respect to the non-functional requirements the e-voting system should meet.

The examples we present here concern only one phase of the whole process, the closing procedures. Moreover they abstract away various details, that, however, are of no relevance for the purpose of the example.

In the diagrams in Figures 1 and 2, i*/Tropos modelling notation is used [2, 21], with goals represented as ovals, and non-functional requirements (softgoals) as clouds. As it was explained in the previous section, 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, which is also represented in both figures.

A number of choices should be made when defining the procedures of e-based ballot counting and data transfer. These choices are validated against the following three groups of requirements.

– the ones which come from the e-voting domain, e.g. the need to provide secrecy of voting, to avoid traceability of votes, to give to external observers (representatives of political parties) certain control over the counting process,

to minimize the changes to the existing legislation imposed by the new e-based procedure, etc.;
- "standard" system/software engineering requirements, such as maintainability and cost concerns;
- security requirements, e.g. the ones related to confidentiality and secure data transfer.

4.1 Counting

In Figure 1 the alternative choices for the counting procedure are analyzed against a number of non-functional requirements.

At the end of the voting day results are accumulated, and either processed locally, or the raw data are sent to the Electoral office and only there the counting is performed. The latter choice has many drawbacks, as it is shown on the diagram. For instance, counting the ballots centrally means first transferring the unprocessed electronic ballots to the Electoral office, which appears to be not only resource-consuming, but also dangerous from the security point of view. The transferred data should be carefully protected against unauthorized access and change by malicious users, as the voting results are not known yet. Also, counting the results centrally limits external observers (representatives of political parties) in their ability to control elections. Moreover, it does not comply with the existing counting procedure and legislation, as in paper-based system ballots are processed in each polling station and only after that they are forwarded to the Electoral office. Thus, this possibility is not considered and ballots are counted in place, i.e. 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 could be performed either separately for each machine, or the data from all machines are to be aggregated and only then processed. The latter alternative, counting per polling station, complies with the existing paper-based counting procedure, but requires additional effort to develop the data aggregation mechanism. Counting the results separately on each machine is the alternative adopted so far and used during the trials. This choice was made due to the technical concerns such as, e.g., the easiness of the recovery from errors (an error can be traced back to the concrete 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 could 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 the votes, which introduces the following conflict. On the one hand, traceability makes it easier to trace and recover from errors, thus improving maintainability and robustness of the whole system. On the other hand, the possibility to associate

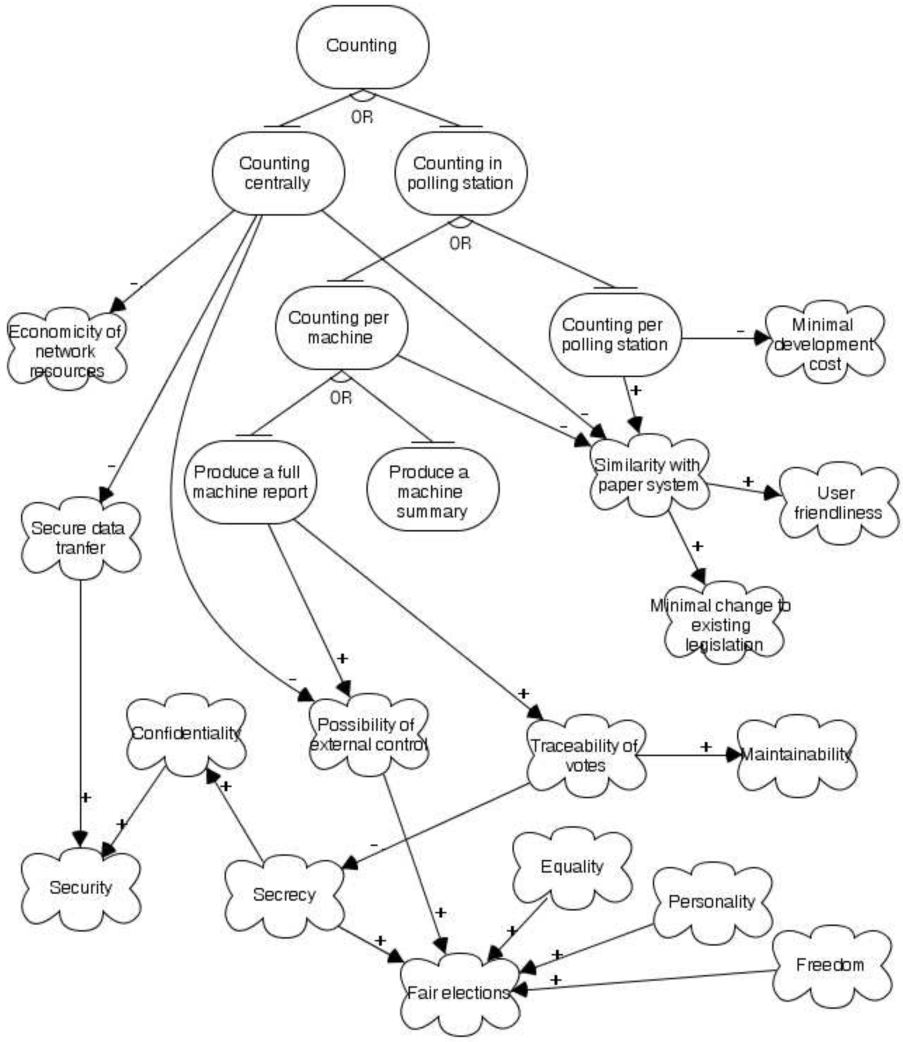

Fig. 1. Counting: reasoning about alternative choices

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.

4.2 Transferring Data

In Figure 2 the (partial) analysis of alternative choices and non-functional requirements of the data transfer process is represented.

After the ballots are processed, the results should be sent to the central point, i.e. to the Electoral office. In the e-voting system developed within ProVotE, three kinds of resulting artifacts represent the voting results: electronic ballots

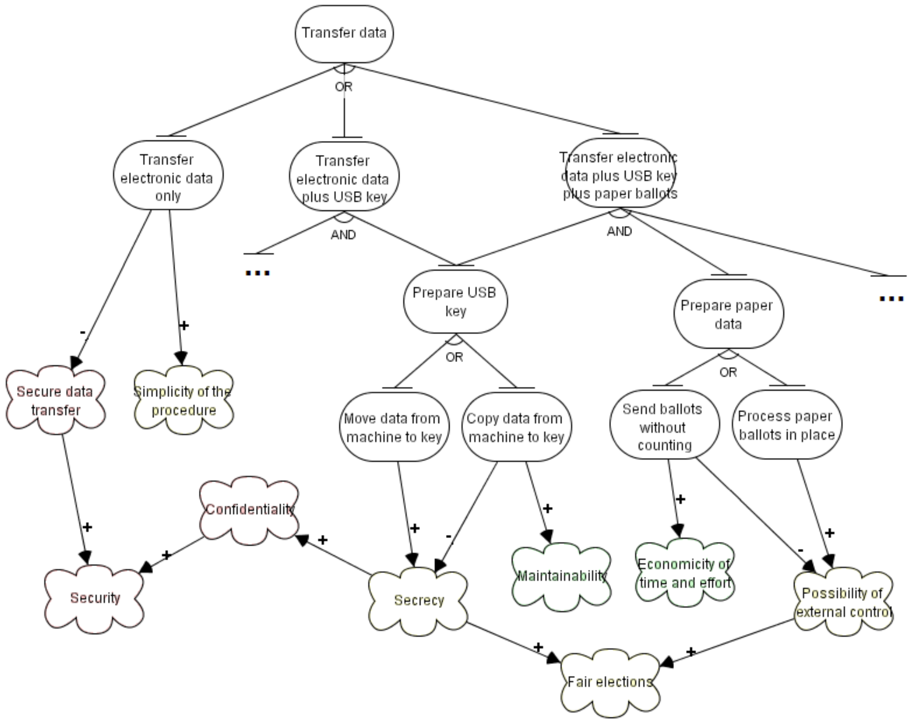

Fig. 2. Transfer data: reasoning about alternative choices

copied from a voting machine on a USB key[6], the USB keys, and the paper ballots produced by the voting machine printer. Electronic data are transferred through a VPN to the central server, while USB keys and paper ballots are either sent to the Electoral office or not (e.g. they might be kept somewhere locally, and used only if some problems with the electronic data occur). Transferring only electronic voting results makes the procedure easy to organize and control, but, at the same time, makes the security issues very crucial. Namely, the connection and the procedure of uploading the data on the server should be secure enough to avoid malicious user intervention. Of course, security should not be ignored when both the electronic data and USB keys and/or paper ballots are transferred, but in this case it is less crucial as deliberate data inconsistencies can be detected and fixed by comparing data on several media.

If a decision to send a USB key (one per each voting machine) to the Electoral office is taken, another choice needs to be made. Should we copy the data from a voting machine to the key, leaving a copy on the machine hard disk, or should we move it, erasing the machine hard disk content? To move data means to reduce

[6] Note that here we assume that one technological choice is already made – at the end of an election day voting data (either processed or raw) are copied from each voting machine on a special USB key, one for each machine.

the number of media which should be carefully protected against unauthorized access, whereas to copy data means to double the number of vulnerable points. On the other side, leaving the data copy on a voting machine contributes positively to system maintainability as it becomes easier to detect and recover from errors.

A number of other choices concern paper ballots, which are sent to the Electoral office either on a regular basis, or only if there is a problem in transferring the other data media, or the results appear to be inconsistent. The choice we consider here is about the processing of paper ballots: should they be counted in a polling station, or only later, in the Electoral office, or even both? Unlike the previous discussed choice, related to moving or coping data from a USB key, these alternatives concern the organization of the new e-based voting process, rather than technological issues. Counting the paper ballots in a polling station requires more time and human resources, however, unlike counting centrally, it allows the external observers to control the process. This last point might be crucial as the interests of the representatives of political parties should also be taken into account while designing the new voting system.

Both examples presented in this section illustrate the kind of analysis that can be done using Tropos. They show how this modelling approach can complement standard UML modelling by covering the aspects which UML is not able to capture. With the help of Tropos models it is possible to represent all the alternative choices considered when building the new electronically based voting procedure. The models allow to analyze the impact that different alternatives have on the system security, maintainability, compliance with the electoral laws, and many other issues. They support validating the taken decisions, documenting the motivation behind them, and detecting conflicts an alternative may introduce.

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 methodology, 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 is 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, it would be useful to be able to express (qualitatively) different level of contribution of goals to softgoals; this is because very often in complex scenarios like this, the same requirement can be favored by more than one solution and the only way to be able to choose among these solutions is to have a sort of "scale" of levels of contribution. Finally, 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.

6 Acknowledgements

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, G. Fele, P. Peri, and F. Sartori. Transitioning to evoting and citizen participation, 2006. In the Proceedings of eVoting-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 expreimentation (sperimentazione sulla votazione elettronica – preliminare). In Italian. Available at `http://www.comune.san-benedetto-del-tronto.ap.it/ePoll/rl00.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 (Elezioni Europee 2004, Conteggio Automatizzato del Voto)*. 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. M. McGaley and J. McCarthy. Transparency and e-voting: Democratic vs. commercial interests. In *the International Workshop on Electronic Voting in Europe*, 2004.
14. R. T. Mercuri and L. J. Camp. The code of elections. *Communications of the ACM*, 47(10):53–57, 2004.
15. J. Mylopoulos, M. Kolp, and J. Castro. Uml for agent-oriented software development: The tropos proposal. In *Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools*, pages 422–441, London, UK, 2001. Springer-Verlag.
16. 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.
17. I. Sommerville. *Software engineering (7th ed.)*. Addison-Wesley, 2004.
18. 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`.
19. A. Villafiorita and G. Fasanelli. Transitioning to eVoting: the ProVotE project and Trentino's experience. 2006. To appear in the Proceedings of EGOV-06.
20. M. Weiss, D. Amyot, and G. Mussbacher. Formalizing architectural patterns with the goal-oriented requirement language. In *Proceedings of VikingPLoP 2006*, September 2006.
21. E. S.-K. Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1996.