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, DISI, via Sommarive 14, Povo (TN) 38050, Italy, {bryl,dalpiaz}@disi.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}@fbk.eu

Abstract: This paper describes part of the work within the ProVotE project, whose goal is the introduction of e-voting systems for local elections. The approach is aimed at providing both precise models of the electoral processes, and mechanisms for documenting and reasoning on the possible alternative implementations of the procedures. It is based on defining an alternating sequence of models, written using UML and Tropos. The UML is used to represent electoral processes (both existing and future), while Tropos provides a mean to reason and document the decisions taken about how to change the existing procedures to support an electronic election.

Keywords: e-Government, e-Voting, security, modelling, automated analysis

Biographical Notes: Volha Bryl is a PhD student at the Department of Engineering and Information Science (DISI), University of Trento, Italy; her research interests are in goal and agent-oriented requirements engineering and automated software engineering.

Fabiano Dalpiaz is a PhD student at the Department of Engineering and Information Science (DISI), University of Trento, Italy; his research interests are in software engineering and autonomic computing.

Roberta Ferrario obtained a PhD in Philosophy in co-tutorship at the University of Milan and at the University Marc Bloch of Strasbourg in 2003. She is currently a researcher at the Institute for Cognitive Sciences and Technologies of the Italian National Research Council (Laboratory of Applied Ontology in Trento) and she is working particularly in ontology of organisations, ontology of services and e-government.

Andrea Mattioli obtained a degree in Computer Science at the University of Trento in 2006. He is interested in requirements engineering, validation and verification of complex system. Currently he is working at the Scientific Research Centre of Bruno Kessler Foundation on the ProVotE and LTPDA projects.

Adolfo Villafiorita is a researcher at the Scientific Research Centre of Bruno Kessler Foundation. His research interests are in software engineering, formal methods, and critical applications development. He is the project leader of the ProVotE project.

1 Introduction

The organisation of an election in Italy involves various offices of the Public Administration and private contractors, has a time-span of months, and has strict security and traceability requirements. Citizens and politicians are highly sensible to these issues, and litigations over, e.g., implementation of procedures and validity of results are not uncommon.

For historical reasons the Autonomous Province of Trento (PAT, from now on) benefits from special autonomy in the definition of laws, including various aspects of electoral matters. Art. 84 of PAT Law 2/2003 promotes the introduction of forms of e-voting for the 2008 provincial elections. To actuate the law, the Province is sponsoring the ProVotE project, that has the goal of providing a smooth transition to the new technologies and that will lead to large-scale experimentations of e-voting.

Switching from paper-based to electronic elections is a challenging task, in which the technological aspect (e.g., the type of systems used to vote electronically) is just a part of a larger problem that includes sociological, political, normative and organisational aspects. Other works - as for instance (Ghapanchi et al., 2008) - advocate the need to adopt a more general, integrated, "holistic" view on e-government processes as a key factor for success. For such a reason ProVotE is organised along different lines, among which is the process/logistical line, which aims at defining the procedural, organisational, and normative framework that will regulate electronic elections. The process/logistical line, in particular, has the goal of suggesting those changes to the existing procedures that are useful or essential to support electronic elections. Similarly to what happens in other re-organisational projects, the line has the goal of re-shaping the existing processes to efficiently exploit the new technologies. Peculiar to the domain, however, is the fact that such changes to the processes must be compatible with the law, such as, e.g., the national laws over which PAT has no rights and which cover some phases of an election and regulate in a very detailed way the steps and the documentation that has to be produced.

Needless to say, “errors” in the definition of the procedures not only may result in more costly or inefficient elections, but may also compromise the fundamental principles of any democratic election, namely equality, secrecy, and freedom of the vote.

To cope with the complexity of the domain, we have defined a methodology (Mattioli, 2006) based on the Unified Modelling Language (UML) (Booch, Rumbaugh, Jacobson, 2005; OMG, 2007) for modelling the electoral processes. However, this is not enough, as we also need a mechanism for reasoning about the models, in order to choose the most effective modification to a given process. For this reason, we complement the UML process models using the Tropos approach (Bresciani et al., 2004), which allows modelling and analysing organisations and information systems in terms of actors, their goals and social dependencies. We use Tropos to reason about process alternatives and to provide means to trace, reason, and document the choices made in devising the electoral “to be” processes.

In the approach proposed in this paper, we use UML and Tropos independently to achieve different and complementary goals. UML is exploited as a notation to formalise procedures and processes (both "as is" and "to be"), whereas Tropos is used to explain the transition from the "as is" (paper-based voting) procedures to the "to be" (electronic voting) process. In particular, Tropos allows depicting the alternative ways of implementing the "to be" processes, and the non-functional requirements each alternative implementation helps to or prevents from satisfying. Secure Tropos (Giorgini et al., 2006) is an extension of Tropos, which allows for modelling and analysis of security and trust relationships among the stakeholders. Thus, even though the actual modelling of the procedures and elicitation of the alternatives still relies on the analyst's expertise, the proposed joint use of UML and Tropos helps to more precisely document the elicitation process and allows for an explicit reasoning about implementation choices.

Various works investigate security aspects of electronic elections. We mention here (Nevo and Kim, 2006) in which the authors propose an extension to the OCTAVE methodology to compare different kinds of voting technologies and (Magkos et al, 2005), that reviews and assess the cryptographic models that have been proposed for e-voting. In our work, by contrast, the e-voting technology to adopt is a project constraint and we focus on the different possible implementations of the procedures to support it.

Other existing works examine the representation and effective implementation of e-voting procedures using business process notations. In (Xenakis & Macintosh, 2004), for instance, the authors argue the need for procedural security in electronic elections and provide various examples of procedural risks occurred during trials in UK. In (Smith, 2007) the author proposes a cost/benefit analysis to improve security of e-voting. In (Xenakis & Macintosh, 2005) the authors investigate the need for applying business process re-engineering to electoral process and in (Xenakis & Macintosh, 2007) they propose a methodology for modelling electoral processes and highlight the importance of defining roles and responsibilities to come to a better understanding of electoral processes. The framework we propose is intended to reason about similar problems to those pointed out by Xenakis and Macintosh, exploiting a set of mainstream modelling techniques in Software and Requirements Engineering. In (Stojanovic et al., 2006) the authors consider e-Government as a continual improvement process, and they suggest the use of the OntoGov ontology to better support change management.

The UML is an ISO standard and several works investigate its usage for secure system development and business process modelling — see, for instance, (McDermott, J., and Fox, 1999; Sindre, G., and Opdahl, 2005; Jurjens, J., 2004; Penker, M., Eriksson, H-E, 2000). The focus of this work, however, is not on the issues related to software development or business process modelling, but, rather, on tools and methodologies to "reason" about different business process models, that are represented using the UML.

The paper is structured as follows. In Section 2 we briefly present the project under which scope this work has been developed; in Section 3 the main features of our proposal are explained, while in Section 4 these same features are illustrated in more details with the help of an example taken from the work we are doing to model the 2008 electoral processes. Finally, in Section 5 we draw the conclusions and sketch some possible future developments.

2 The Scenario: e-Voting and ProVotE

2.1 The ProVotE Project

ProVotE, a project sponsored by PAT, has the goal of ensuring a smooth transition to e-voting in Trentino. 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 centres and academia (FBK-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 FBK-IRST. Project leadership by the Public Sector helps tackling the issue of potential conflicts of interests of private industries; see e.g. (McGaley & McCarthy, 2004). The technological solution (both software and some hardware components) has been developed in-house, is integrated with some commercial components and covers the phases from voting to the publication of the elected candidates.

The project is multi-phased and multi-disciplinary. Various requirements of the e-voting prototype have been provided with a strict round-trip between the sociological and the technological line, with the normative line ensuring compatibility with the electoral laws. See (Villafiorita & Fasanelli, 2006 and Caporusso et al., 2006) for more details and (Ostveen & Van den Besselaar, 2004) for some considerations related to the sociological aspects of e-voting.

Each project phase defines milestones to check the goals set in each different line of activities. The first phase had the goal of testing technological prototypes, evaluating acceptance by citizens, and ease of use. Verification of the results was conducted through four different trials held during local elections. During the trials polling stations were equipped with one or more e-voting machines and citizens were asked to vote on paper, repeat their vote using the electronic systems, and provide feedback about the system. About 10.000 citizens took part to the trials. Detailed results of all the experimentations and elections conducted within the ProVotE project are available on the Internet[1].

During the second phase of the project we used the electronic systems in two elections with legal value. The first was the election of student representatives in a local high school and it involved 1.298 students. The second election was a poll in the towns of Campolongo al Torre and Tapogliano (in Friuli-Venezia Giulia, a neighbouring region with autonomy similar to that of PAT) to unify the two municipalities; 561 people used the systems. In both cases, logistics, procedures, and laws governing the elections were relatively simple and can be considered a simplified version of the other kinds of elections we intend to use our systems for.

For the third phase of the project, which could lead to a large-scale introduction of the new voting system, aspects related to procedures, logistics, and organisation 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 only at polling stations and proceeds as follows:

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

ii. **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:

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

[1] See http://www.provincia.tn.it/elezioni and http://ed.fbk.eu/

iv. **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 for aggregating all the data.

v. **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 in the law. Data are then made available to the general public.

Apart from the ProVotE project (which supports steps ii to v), some experimentations have been conducted in Italy to introduce new technologies for elections. The largest trial, so far, was sponsored by the central government and concerned a system for automating step iv above. The system, operated by specially appointed technicians, was installed in 47 precincts at the 2004 European elections and repeated at the general elections of 2006. Little, however, is known about the results of the experimentation. See (Governo Italiano, 2004) for some more details.

Proper e-voting experimentations (i.e. including step ii) have been conducted at a local level, usually on a small scale, but they seem to have had little continuity and/or information about them is scarce. The major trials have been conducted in San Benedetto del Tronto (2000), Avellino (2001), Campobasso (2001), Cremona (2002, 2006), Ladispoli (2004), Specchia (2005). References can be found in (Comune San Benedetto del Tronto & MET Informatica) and (E-poll). 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 polling stations changes not only the way in which votes are cast, but also roles and responsibilities of actors and stakeholders, often in subtle ways — see e.g. (McGaley & McCarthy, 2004). 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 voting procedures (e.g., allow the parties and polling officers to test the machines long before an election; provide ways to verify what software is installed on the machines used during an election day).

To mitigate the risk of creeping security "holes" in the electronic procedures, we decided to provide extensive modelling of the electoral 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 electronic means. 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 to the way of voting.

3.1 Integrating UML and Tropos modelling approaches

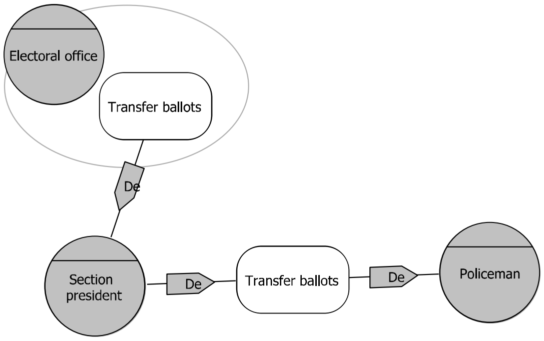
The modelling of the current electoral processes has been performed by devising a specific methodology (Mattioli, 2006), based on the UML, to support analysts in the production of models while guaranteeing, at the same time, uniformity and standardization. The use of UML, in our case, was an essential requirement for various reasons, among which expertise, tool support, and ease of understanding by the domain experts.

The UML-based methodology, which has been used to model town and provincial elections, is based on three perspectives: processes & actors, processes & managed entities, and processes & law constraints. These perspectives are aimed at keeping the bulk of data under control (e.g., about 80 processes, 30 actors and over 90 entities were identified), sharing processes, which are common to various types of elections, and enabling automated analyses. The methodology uses three different types of UML diagrams: *use case diagrams* are employed to provide a static process view, by organizing processes hierarchically as use cases and showing which actors are entitled to participate in their realisation; *activity diagrams* deal with the dynamic aspects of the system, rendering processes as actions which transform entities and their state (i.e. they can be seen as *transformation functions* that change data and physical resources); *object diagrams* focus on describing the law which influences the process. Detailed standards on the models allow automating information extraction and analysis. Among the supported functions are: automatic computation of what actors are responsible for which processes (RACIV responsibility matrix), the identification of which resources are needed and who is in charge of employing them (CRUD matrix), and the automatic generation of some documentation (e.g., a web site to publish the electoral processes for increasing administrative transparency).

While providing models of the paper-based electoral procedures in the UML is a relatively straightforward step, using it to model the possible alternative implementations of the "to be" procedures and to motivate the choices made, is a lot more complex. This is because UML is weak at representing non-functional requirements (e.g., cost-effectiveness, efficiency, security, etc.) and at representing alternative implementations. Hence, there is a need to complement UML modelling with some other approach more suited to face these aspects and describe "*the why*" of choices.

Our proposal is to fill the gap with Tropos (Bresciani et al., 2004), an agent-oriented software development methodology. The main entities that populate models in Tropos come from the i* modelling approach (Yu, 1995), and are actors, goals, and dependencies (between actors for goals). In Tropos, both organisations and information systems are modelled as networks of interdependent actors endowed with goals.

Figure 1: Organisational environment modelling using Tropos.



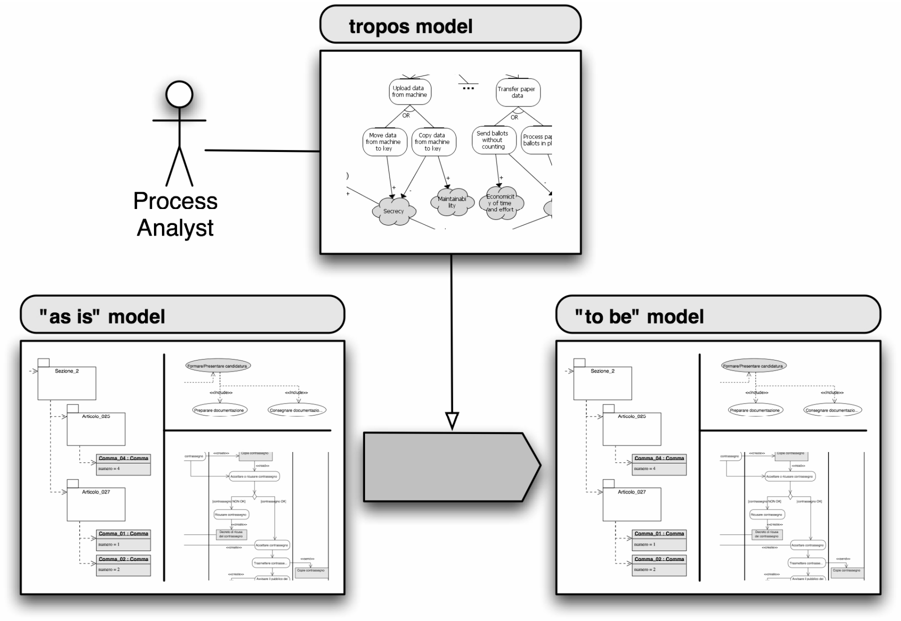
In Tropos, the analysis of system requirements starts with modelling an organisational environment in terms of stakeholders, their strategic goals, and the social relations between them. In this way, Tropos helps to *understand* and *motivate* the changes that should incur to the organisational structure and procedures when, e.g., an information system is introduced. This makes Tropos suitable for solving the problems left open by UML modelling activity, namely, 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. Figure 1 depicts a small scenario showing how to use Tropos for modelling an organisational environment, where three stakeholders (agents) are identified (*Electoral office*, *Section president*, and *Policeman*). The agent *Electoral office* has the strategic goal *Transfer ballots*, and delegates the execution of that goal to the *Section president*. In turn, the *Section president* delegates the execution of the goal to the *Policeman*, which terminates this social relation chain. This means that the stakeholder who actually achieves the goal of transferring ballots is the *Policeman*.

Modelling and analysing design choices is done in Tropos by means of goal analysis (Giorgini et al., 2002), which enables automated reasoning and inference on goal models. Relations among goals in a goal model amount to a positive or negative contribution a goal can have to the achievement of another goal, and a goal decomposition into subgoals, which can be either an and-decomposition (all the respective subgoals must be fulfilled in order to fulfil the parent goal) or an or-decomposition (the subgoals are alternatives: it is enough to fulfil one of them to achieve the parent goal). Some works, e.g. (Giorgini et al., 2002 and Gross & Yu, 2001), use goal analysis to model the choice among "to be" alternatives by representing functional alternatives with or-decomposition, and then analysing 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 better favour the satisfaction of a requirement. In this paper, goal analysis is used in Section 4.2 and graphically presented in Figure 5.

Finally, an extension of Tropos, called Secure Tropos (Giorgini et al., 2006), specialises the Tropos dependencies into security specific relations, such as trust and delegation of permission. This is also relevant 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. The key idea of the integrated approach is that of keeping each modelling approach to do just what it is best suited for: 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 Tropos models help to keep track of the reasons for any change that has been introduced.

Figure 2: Combining UML and Tropos modelling to devise the "to be" model.



From a technical standpoint, this translates into an approach, which produces an alternating sequence of UML and Tropos models, as shown in Figure 2. 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:

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

The results of the analysis allow, in turn, modifying the existing UML models to devise the new procedures that meet the requirements stated in the Tropos model. The steps described above are then iterated as needed. In the following, we first focus on modelling and analysing functional alternatives, and then explain how security and trust analysis is organised.

3.2 Modelling "to be" alternatives and non-functional system requirements

Ideally, every solution in the system "to be" should be taken after having accurately explored all the alternative possibilities. Tropos is exploited as visual and analytical tool supporting the people involved in the *decision-making* process, so that they can 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, Tropos modelling is useful 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 better suit 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. The main purpose of the model is exploring, evaluating and motivating choices between alternative ways of accomplishing a goal with respect to a list of non-functional requirements, and the methodological questions amount to the following two:

- *How are the different alternatives singled out?* That is, how to transform well established procedures based on physical support, like pencils, sheets of paper, cardboard boxes, etc. in e-based practices? The possible alternatives are constrained by several dimensions: *technological, legal* and *social*. The main source for the formulation of the alternatives has been the stakeholders of the project: interviews conducted with the development team raised technological issues, whereas meetings with the representatives of the Electoral Service of the Province of Trento highlighted the need of compliance with the provincial legislation regulating elections.

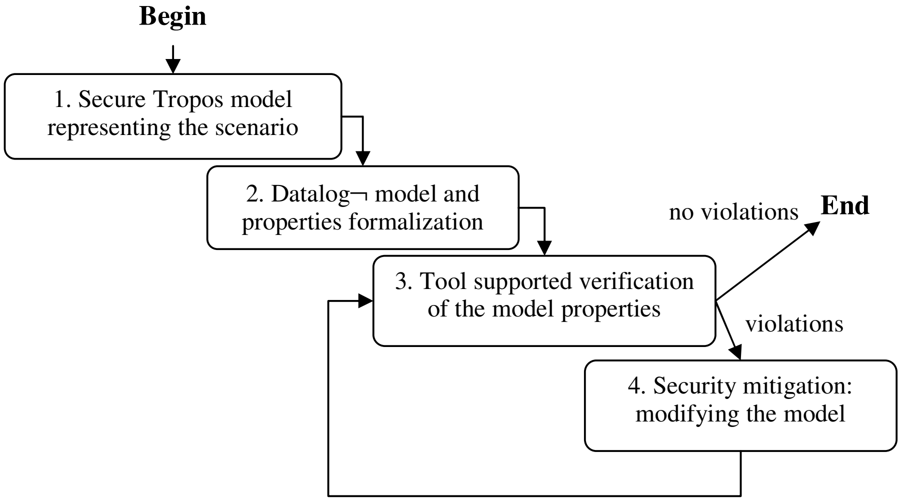
- *How are the requirements that provide the reference for evaluation selected?* Several sources of non-functional requirements were considered. Requirements such as maintainability or cost concerns, which are desirable for any information system, have mainly been taken from the software engineering literature (see (Sommerville, 2004) and (Chung et al., 2000)). Security requirements, such as confidentiality, integrity, availability, etc. (see e.g. Chung et al., 2000, Chap. 7), are particularly relevant in the e-voting scenario, since it is crucial that the system is not vulnerable. Domain-specific requirements, such as non-traceability of votes or minimal change to the existing legislation, were elicited by taking inspiration from existing work, such as, for instance (Venice Commission, 2004) and (EAC, 2002). Finally, a specific requirement of ProVotE project is a smooth transition from the old paper based system to e-voting. This objective introduces a very stringent requirement, which is compliance with the existing PAT voting legislation (Regione TAA, 2005). Law compliance is important for several stakeholders (not only legislators, but also common citizens), for changing laws is a (politically and bureaucratically) complex and time-consuming process.

3.3 Reasoning on security properties of "to be" models

After alternative design options are analysed, and a satisfying one is adopted, security properties of the model should be verified. The approach we propose is based on the use of Secure Tropos (Giorgini et al., 2006), which describes an organisational scenario in terms of socially interacting agents that aim at the achievement of their own goals, either directly or delegating the responsibility to other agents. The main security relations between agents are those of delegation, trust, and ownership. This activity is performed with two objectives:

- Identify the "problematic" trust/delegation relationships, such as cases where an actor delegates the accomplishment of an important goal to another actor whom he/she does not trust;
- Adopt a solution to the detected security properties violations, such as monitoring the accomplishment of goals delegated along untrusted links.

Figure 3: The analysis process using Secure Tropos.


Begin
1. Secure Tropos model representing the scenario
2. Datalog¬ model and properties formalization
3. Tool supported verification of the model properties
no violations
End
violations
4. Security mitigation: modifying the model

These objectives are accomplished through the process presented in Figure 3. Firstly, a Secure Tropos model of all trust and delegation relationships is constructed (step 1). Both the scenario and the security properties to check are then formalised (step 2) using Datalog¬ (Eiber et al., 1997). The formalization activity is tool supported (SiStarTool, 2007): the translation of a model into Datalog¬ is fully automatic, a number of basic security properties are built into the tool, and any additional required property can be manually encoded in an auxiliary property definition file. Reasoning is then performed to verify the security properties of the model (step 3). If no violation is detected, the process ends successfully; otherwise, an analyst modifies the Secure Tropos model proposing a solution to mitigate the detected security problems (step 4). There exist a number of "standard" solutions, or patterns, which may help an analyst during this stage. Then, another verification run is executed to check the correctness of the model. The verification process is automatically executed by the tool running DLV (Leone et al., 2006), a Datalog¬ solver that is used to compute and show the violations.

In order to better explain the type of security concerns that can be expressed and verified using our framework, we list some examples:

- **untrusted delegation of execution** takes place when the execution of a service S (i.e. a goal or a task) is delegated from an actor A to an actor B, but there is no trust for S between these two actors;
- **trust conflict** occurs when an actor B is in charge of executing a service S, and there are two (or more) actors whose trust relations with B for S have different polarities, for instance, actor A trusts B for S, while actor C distrusts B for S;
- **need-to-know** principle intuitively states that an actor should have only those permissions that are strictly necessary to achieve his/her duties. To not violate this principle, permission for a service should be delegated only to an actor who is either in charge of a service or will further delegate the permission to another actor who actually needs it.

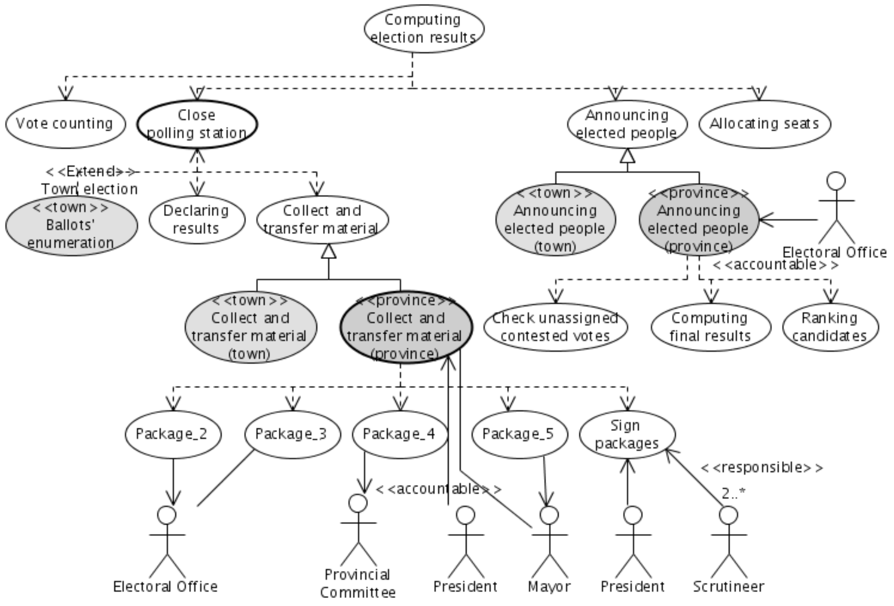
4 Case study: modelling and analysis

In this section we will illustrate the approach presented above with the help of a fragment taken from the e-voting scenario. The fragment regards an activity performed after voting is finished, namely, the process of transferring election results from the polling station to the Electoral Office. We describe the modelling of the as-is procedures, the selection among design alternatives with respect to non-functional requirements (NFR), and the modelling of the to-be procedures. Due to space limitations, we do not show here the Secure Tropos modelling of the scenario; the interested reader can refer to (MOSTRODel8, 2007).

4.1 Description of the as-is scenario

Figure **4** shows a use case diagram representing part of the as-is scenario, that is the main processes performed the day after voting is over.

Figure 4: The UML as-is use case diagram



Computing election results
Vote counting
Close polling station
Announcing elected people
Allocating seats
<<Extend>>
Town election
<<town>> Ballots' enumeration
Declaring results
Collect and transfer material
<<town>> Announcing elected people (town)
<<province>> Announcing elected people (province)
Electoral Office
<<accountable>>
<<town>> Collect and transfer material (town)
<<province>> Collect and transfer material (province)
Check unassigned contested votes
Computing final results
Ranking candidates
Package_2
Package_3
Package_4
Package_5
Sign packages
<<responsible>>
2..*
<<accountable>>
Electoral Office
Provincial Committee
President
Mayor
President
Scrutineer

Each use case represents a process. The process tree is to be read top-down and from left to right: each horizontal layer represents a different level of details (e.g., top layers contain abstract processes), while the process sequence suggests the order in which operations are executed. The exact process order is defined by the activity diagrams, which are also part of the as-is scenario modelling, but are not shown here, as they are not particularly relevant for the purposes of this paper. As shown in the diagram, the *Computing election results* process consists of four sub-processes: vote counting, closing polling stations, announcing elected candidates, and allocating seats.

At the end of vote counting (not further refined here), the president of the section prepares the polling station material, e.g., the votes, for transmission**.** The goal is verifying the consistency of the sums, namely that votes and voters correspond and that no ballot has been stolen from the polling station. The next step is sending the material to the responsible offices. As the number of packages to be prepared and the receivers depend on the election type, the model distinguishes different situations as a use case generalisation (see *Collect and transfer* process). In case of provincial elections, there are five packages to be prepared; number two and three are sent to the Electoral Office, while packages four and five to the provincial committee and to the mayor, respectively. After receiving all the material, the Electoral Office is responsible for judging marked or ambiguous ballots, updating the final result and declaring the number of votes obtained by each candidate, party and coalition. Finally, elected candidates are officially proclaimed.

For the sake of completeness, we remark that the modelling methodology also introduces stereotypes on use cases, actors and connectors; their function is to categorize processes, actors and to define relations among them, respectively. In particular, stereotypes on the "use case – actor" associations allow describing how an actor participates to the process, in order to automatically build a RACIV (Responsible, Accountable, Consulted, Informed, Verified) responsibility matrix. For instance, the *«town»* and *«province»* stereotypes represent the election type a use case belongs to; the use case *Ballots Enumeration* is in place only for town elections.

We exploited different shades to visually distinguish use cases belonging to town and province elections. Darker shading implies provincial elections, whereas lighter shading means town elections. Our further analysis will mostly focus on use case *Close polling station* and especially on its sub-process *Collect and transfer material* (for Provincial elections).

4.2 Modelling and evaluating alternatives: transfer of votes

In this section, we show how alternative choices are modelled and evaluated with respect to the non-functional requirements the e-voting system should meet. In Figure 5 Tropos modelling notation is used, with goals represented as ovals, and non-functional requirements (or softgoals in Tropos) as clouds. For a softgoal there are no clear-cut criteria for determining 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 can be decomposed into or- or and-subgoals; the former type of decomposition is represented in the diagram, to represent alternative strategies to achieve a goal.

Many choices are needed to define the e-based counting procedures. These choices are validated against the three groups of requirements mentioned above: (i) domain-specific requirements, such as traceability of votes and secrecy of voting; (ii) "standard" system/software engineering requirements, such as maintainability and cost; (iii) security requirements, such as confidentiality and secure data transfer.

At the end of an election day, the election results of each section are sent to the Electoral Office. In the e-voting system developed within ProVotE, three kinds of artefacts represent these results: electronic ballots copied from a voting machine to a USB key, the USB keys, and the paper ballots produced by the voting machine printer. Electronic data are transferred through a Virtual Private Network (VPN) to the central server, while USB keys and paper ballots can be either physically sent to the Electoral Office or kept somewhere locally, and used only if some problems with the electronic data occur.

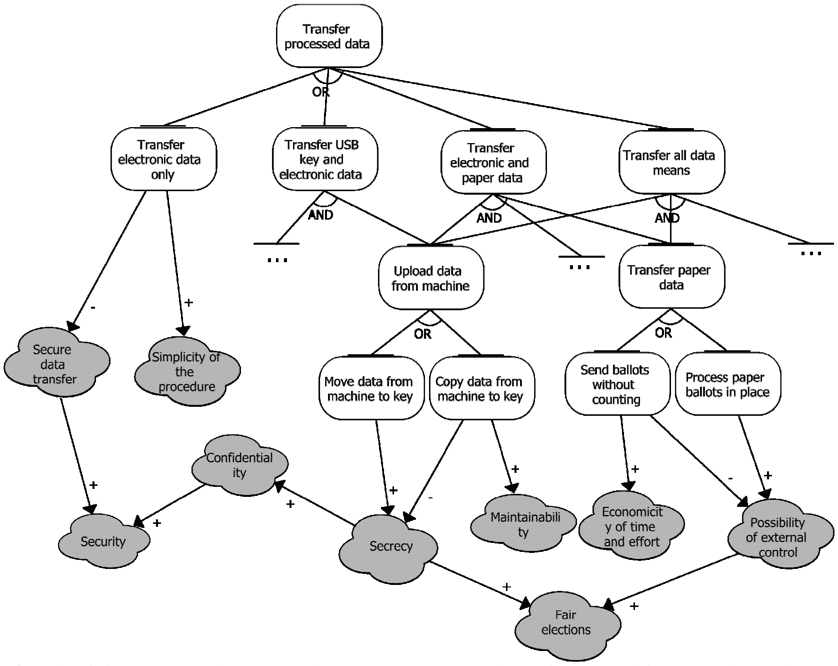
There are various ways for carrying out the data transfer process. The actor responsible for this duty is the Section President. We list four basic possible alternatives:

- **Transfer electronic data**: this option removes the need of any physical data transfer, but it requires all the actors to trust the reliability of both electronic data means and transfer channels;
- **Transfer USB keys and electronic data**: this option adds to the previous one redundancy in the transferred data, and stills requires strong trust in the electronic data means and channels;
- **Transfer electronic and paper data**: paper data is still considered part of the process, which is important in cases in which certain actors consider paper based data means much more reliable than the electronic ones;
- **Transfer all data means**: all the data means are transferred, in order to provide the highest level of security requirements such as, e.g., data integrity and the reliability of the transfer procedure.

Figure 5 represents the (partial) contribution analysis of alternative choices and non-functional requirements of the data transfer process.

Transferring only electronic data makes the procedure easy to organise and control, but makes the security issues crucial. Namely, the connection and the procedure of uploading the data on the server should be secure enough to avoid malicious user intervention. This observation is the key motivation for preventing the adoption of this design alternative.

Figure 5: Transfer of votes: reasoning about alternative choices.


Transfer processed data
OR
Transfer electronic data only
Transfer USB key and electronic data
Transfer electronic and paper data
Transfer all data means
AND
AND
AND
...
...
...
Upload data from machine
Transfer paper data
OR
OR
Move data from machine to key
Copy data from machine to key
Send ballots without counting
Process paper ballots in place
Secure data transfer
Simplicity of the procedure
Confidentiality
Security
Secrecy
Maintainability
Economicity of time and effort
Possibility of external control
Fair elections
-
+
+
+
+
+
-
+
+
-
+
+
+

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 data be copied from a voting machine to the key, leaving a copy on the machine hard disk, or should it be moved, erasing the machine hard disk content? Moving the data reduces the number of media which should be carefully protected against unauthorised access, whereas copying the data doubles the number of vulnerable points. However, leaving the data copy on a voting machine contributes positively to system maintainability as it becomes easier to detect and recover from errors. The latter reason motivates the choice of copying the data to USB keys rather than moving and erasing it.

A number of other choices concern paper ballots, which are sent to the Electoral Office either on a regular basis, or only in exceptional cases, for instance if there is a problem in transferring the other data media, or if the results appear to be inconsistent. The choice we consider here concerns the processing of paper ballots: should they be counted in a polling station, or only later, at the Electoral Office, or even in both places? Unlike the above discussed choice (moving or coping data from a USB key), these alternatives concern the organisation 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 latter 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.

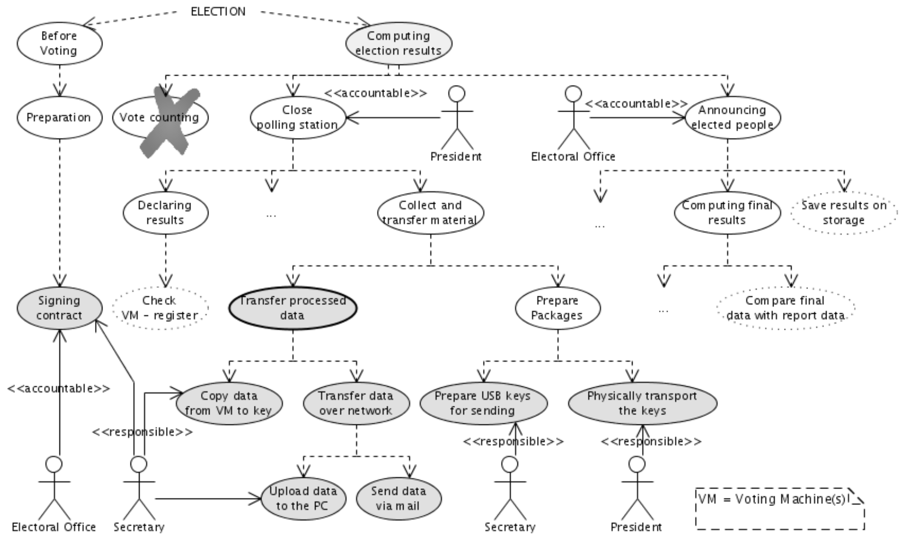
In (Bryl et al., 2007) the analogous contribution analysis of another other e-voting subprocess (vote counting) is presented.

4.3 Modelling the to-be processes

The last phase in our approach is the *to-be* processes modelling. This activity, which derives from the *as-is* UML use cases and activity diagrams, aims at representing the hypothetical future scenario, integrating expert comments with the Tropos modelling and analysis suggestions. The resulting model can then support procedure validation and establishment or act as a launching pad for the next analysis iteration.

Looking at the portion of the *to-be* model in Figure 6, it is apparent how many processes need changes with the introduction of new technologies. Many of them can be immediately included or redefined in the domain, without detailed explanation or law transformation, while others require choosing which strategy is the most reasonable. As an example, some processes disappear, like the *Vote counting* (because manual counting is no more needed); others are moved or combined together, as *Allocating seats*, which is merged with the polling results computation and the candidates ranking. Some other processes slightly change, like the result declaration (*Declaring results*), in which the number of voters stored in the voting machine (VM) should be compared with the number manually registered by the polling officers during the voter identification (the use case *Check VM-register* is added).

Figure 6: UML use cases diagram modelling the "to-be" processes.



More stimulating are those processes evolved and identified as a result of the evaluation of alternatives, done using Tropos goal reasoning, and improved by the Secure Tropos automated analysis.

In particular, the *Collect and transfer material* process established in the *as-is* phase becomes more complex in the electronic election, requiring, as stated by the Tropos analysis, both to move data over the network and to physically transport the USB keys in order to transfer the processed data with an adequate degree of confidence. Furthermore, the Secure Tropos analysis revealed that to increase trust in this process we should add a new one, *Signing contract*, where the secretary signs a contract with the Electoral Office, in which he/she states his/her responsibility in copying data from the machine to the USB key. It can be noticed that this process does not belong to the same abstract process related to counting activities, but resides in the electoral preliminary operations, before voting. Finally, the introduction of this new process justifies, through earlier Tropos analysis, the link between the Secretary and the preparation of the USB keys, which otherwise would be there without any explanation.

To summarise, the options chosen during Tropos modelling and analysis (both in terms of non-functional requirements contribution and trust relations) help in the definition of the *to-be* electoral process, selecting the most appropriate option among the possible alternative processes.

5 Conclusions

We believe that understanding the implications of possible alternatives in the re-definition of complex organisational processes is a challenging and important task. This is particularly true in domains, such as that of voting, in which the fundamental rights of citizenship and democracy are at stake.

In this paper we have presented an approach, based on the integration of UML and Tropos, and shown how we have been using it for migrating PAT electoral system from paper to electronic without losses in security and reliability of the voting process. The integration exploits complementary features of the two modelling approaches and allows maintaining both an operational view of the voting procedures and a visual approach to evaluate choices in designing the “to be” processes. UML models are used to express the “what” and the “how” of the voting procedures, while Tropos goal models specify the “why”.

Among the advantages are the possibility of representing the processes in a way that is well understood by the stakeholders and the functional analysts, the possibility of clearly reasoning about possible alternatives and documenting the choices taken. In addition, Secure Tropos analysis of trust and security concerns emerging from the relations of the involved actors allows proposing models of the future procedures in which these concerns are taken into account and solved.

The approach, even though its definition has been motivated and driven by a specific project need, is not restricted to the e-voting domain. We believe that our approach can be re-used in other business process re-engineering contexts, especially in those situations in which a clear understanding and documentation of the possible alternatives provides a high value. This is the case, e.g, in several e-government scenarios, where the attribution of responsibilities is central and, thus, the analysis in terms of trust relations reveals all its usefulness.

Future work develops along different lines. From the process point of view, extensions of the tools to support automated analysis are a top-priority. Another improvement consists of refining the connections between the methodology activities, in order to support and simplify analysts' activities.

From the ProVotE point of view, concluding the definition of the "to be" processes is the step we are currently working on. It will be the basis for proposing the changes to the current legislation. When that happens, to our knowledge, it will be the first case in Italy in which the definition of a law will be aided by an explicit modelling of the procedures the law intends to represent.

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

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

Booch, G., Rumbaugh, J., Jacobson, J., The Unified Modeling Language User Guide. Second Edition, Addison Wesley, 2005

Bryl, V., Dalpiaz, F., Ferrario, R., Mattioli, A., and Villafiorita, A. (2007). Evaluating Procedural Alternatives. A Case Study in E-Voting. In Proc. MeTTeG'07.

Caporusso, L., Buzzi, C., Giolo, F., Peri, P., and Sartori, F. (2006). Transition to electronic voting and citizen participation. In Krimmer, R. (Ed.) Electronic Voting 2006 (191-200).

Chung, L. K., Nixon, B. A., Yu, E., and Mylopoulos, J. (2000). Non-Functional Requirements in Software Engineering. Boston: Kluwer Publishing.

Comune di San Benedetto del Tronto, and MET Informatica. Preliminary report on the electronic voting experimentation. Available at http://www.comune.san-benedetto-del-tronto.ap.it/ePoll/rl00.html.

Eiter, T., Gottlob, G., and Mannila, H. (1997). Disjunctive Datalog. ACM Transactions on Database Systems (TODS), 22(3), 364-418.

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

EAC (2002). Voting systems performance and test standards. Available at http://www.eac.gov/election_resources/vss.html.

Ghapanchi, A., Albadvi, A., and Zarei, B. (2008). A framework for e-government planning and implementation. Electronic Government, an Int. J, 5(1), 71–90.

Giorgini, P., Massacci, F., Mylopoulos, J., and Zannone, N. (2006). Trust Management: Model, Methodology, and Reasoning. The International Journal of Information Security, 5(4), 257-274.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E., and Sebastiani, R. (2002). Reasoning with Goal Models. In Proc. of ER'02 (167-181).

Governo Italiano (2004). European Elections 2004, Automated Counting of the Votes. Available at http://www.governo.it/GovernoInforma/Dossier/voto_conteggio.

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

Jurjens, J. Secure Systems Development with UML. Springer-Verlag, 2004.

Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., and Scarcello, F. (2006). The DLV system for knowledge representation and reasoning. ACM Transactions on Computational Logic (TOCL), 7(3), 499-562.

Magkos, E., Kotzanikolaou, P., Douligeris, C. (2007). Towards secure online elections: models, primitives and open issues. Electronic Government, an Int. J., 4(3), 249 – 268.

Mattioli, A. (2006). Process analysis in the electronic voting domain about the elections in the Province of Trento, Università degli Studi di Trento.

McDermott, J., and Fox, C. Using Abuse Case Models for Security Requirements Analysis. In Proceedings of 15th Annual Computer Security Applications Conference (1999), IEEE Computer Society Press, pp. 55–66.

McGaley, M., and McCarthy, J. (2004). Transparency and e-voting: Democratic vs. commercial interests. In The International Workshop on Electronic Voting in Europe.

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

MOSTRODel8 (2007). MOSTRO Deliverable 8, available at http://www.loa-cnr.it/mostro/files/MostroDel8.pdf.

Nevo, S. and Kim, H. (2006). How to compare and analyse risks of internet voting versus other modes of voting. Electronic Government, an Int. J, 3(1), 105–112.

OMG, Unified Modeling Language (UML), Version 2.1.2, Nov 2007. Available for download from http:// www.omg.org/technology/documents/modeling_spec_catalog.htm#UML

Ostveen, A., and Van den Besselaar, P. (2004). Security as belief – user's perceptions on the security of electronic voting systems. In The International Workshop on Electronic Voting in Europe.

Penker, M., Eriksson, H.-E. Business Modeling With UML: Business Patterns at Work, Wiley, first edition, 2000

Regione TAA (2005). Regional regulations about the composition and the election of the local administrations, DPReg n.1/L, 1st February 2005.

Sindre, G., and Opdahl, A. L. Eliciting security requirements with misuse cases. Requirements Engineering Journal 10, 1 (2005), 34–44.

SiStarTool (2007). The tool is available at http://sesa.dit.unitn.it/sttool/.

Smith, A. D. (2007). Securing e-voting as a legitimate option for e-governance. Electronic Government, an Int. J., 4(3), 269–289.

Sommerville, I. (2004). Software engineering (7th ed.). Addison-Wesley.

Stojanovic, L., Stojanovic, N., and Apostolou, D. (2006). Change management in e-government: OntoGov case study. Electronic Government, an Int. J., 3(1), 74–92.

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.

Villafiorita, A., and Fasanelli, G. (2006). Transitioning to eVoting: the ProVotE project and Trentino's experience. In Proc. of EGOV-06.

Xenakis, A., and Macintosh, A. (2004). Procedural Security Analysis of Electronic Voting. In ICEC '04: Proceedings of the 6th international conference on Electronic commerce (541–546).

Xenakis, A., and Macintosh, A. (2005). Using Business Process Re-engineering (BPR) for the Effective Administration of Electronic Voting. The Electronic Journal of e-Government, 3(2), 91-98.

Xenakis, A., and Macintosh, A. (2007). A methodology for the redesign of the electoral process to an e-electoral process. Int. J. of Electronic Governance, 1(1), 4–16.

Yu, E. S. K. (1995). Modelling strategic relationships for process reengineering. PhD thesis, University of Toronto.