40

FUNCTIONAL REQUIREMENTS FOR A SECURE ELECTRONIC VOTING SYSTEM

Spyros IKONOMOPOULOS[1], Costas LAMBRINOUDAKIS[1], Dimitris GRITZALIS[2], Spyros KOKOLAKIS[1], Kostas VASSILIOU[1]

[1] *Dept. of Information and Communication Systems, University of the Aegean*
Karlovassi, Samos GR-83200, Greece
e-mail: {ikono,clam, sak, kvas}@aegean.gr

[2] *Dept. of Informatics, Athens University of Economics and Business*
76 Patission Ave., Athens GR-10434, Greece
e-mail: dgrit@aueb.gr

Abstract: Electronic voting has been attracting the attention of governments and research groups with most work on the subject referring to the user requirements such a system should satisfy. For several cases, though, requirement identification seldom goes further than a simple narrative description of a basic set of non-functional characteristics related to security. On the other hand, governmental reports usually refer to requirements as the set of applicable laws pertaining a certain voting procedure. Both sides seem to underestimate the fact that an electronic voting system is an information system with functional, as well as non-functional, requirements. In this paper we apply the Rational Software Development Process for identifying and presenting the requirements an electronic voting system should meet. The requirements are based on a generic voting model that has been developed having in mind the European Union member states legislation, the organisational details of currently applicable voting procedures and the opportunities offered and the constraints imposed by the state-of-the-art technology.

Keywords: Electronic Voting, Security, Privacy, Voting Model, User Requirements, Unified Modelling Language, Rational Unified Process.

Acknowledgements: This work has been supported in part by the European Commission, IST Programme, Project *e-vote* (*An Internet-based electronic voting system*). We thank our colleagues for all their comments and suggestions.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35586-3_46

M. A. Ghonaimy et al. (eds.), *Security in the Information Society*

1. INTRODUCTION

The advent of information society has enabled people to perform most of their activities in a direct, electronically automated, and efficient way. To keep up with the need to provide citizens with the ability to participate and benefit from services over the Internet, as well as to reduce the cost and bureaucracy of public administration, contemporary states are striving to transfer an ever-increasing number of their activities to the new medium.

Electronic voting has been considered to be an efficient and cost effective alternative/complement of the classic voting procedure, as well as a way to attract specific groups of people, like young electors [1], to participate. However, in parallel to their initial interest, state authorities are concerned, and need justification, on the compliance of electronic voting systems with the current legal framework. Along these lines, it is rather usual to identify the "requirements" of an electronic voting system merely as guidelines to conform to the legislation governing general elections [2].

On the other hand, information system developers approach electronic voting with an eye towards identifying the fundamental problems associated with the *adequate level of security* (anonymity, authentication, data security, tractability, etc). It seems though that the severity of these problems has attracted most of the attention, since the majority of the literature concentrates on the ability of an electronic voting system to handle them [3, 4][1]. In [6], for example, this distinction is apparent since requirements are identified as legal, technical, and user oriented - the latter in the form of conditions the system should meet (e.g. "The system *shall* allow online-voting from home"). Other authors select a specific election procedure, e.g. the paper absentee ballot process [5], deriving requirements for electronic voting systems based solely to this procedure.

Although such approaches may produce secure and/or legitimate electronic voting systems, they have not led to the specification of a *complete* system. Thus, the focus of our work is to express the: (a) legal, (b) functional and (c) security requirements in a common *User Requirements Specification*, which will be suitable for providing information system designers with the essential information for designing a valid and complete system. A fundamental milestone of our work has been the development of a generic *voting model* depicting the principles, practices, and processes followed during elections.

The structure of this paper has as follows; in section 2 we describe the methodology adopted. Section 3 presents the voting model developed as a basis for the requirements elicitation process, emphasising the properties

[1] The amount of work on the subject is considerable. We just refer to two papers here, in order to provide the interested reader with an indicative reference.

ensuring a proper voting procedure. Section 4 includes the resulting requirements specification; we focus on functional requirements, depicted in the form of Use Cases. Finally, in section 5 we draw some conclusions and provide some pointers for future work.

2. METHODOLOGY USED

In this paper, requirements elicitation has been based on the Rational Unified Process [7, 8]. The Rational Unified Process is the synthesis of various software development processes [7]; one of its most important characteristics is that it is use-case driven. Use cases as a requirements capturing method were first introduced by Jacobson [9] and despite their somewhat informal nature have become a popular tool [10]. Each use case mainly refers to a functional requirement [11] of the system. Non-functional requirements specific to a use case may become part of its description, whilst system-wide non-functional requirements are usually specified as supplementary specifications [12].

A fundamental activity of the requirements elicitation process is the development of the domain model demonstrating current workers and processes. Initially, a business use case model is developed demonstrating current processes or *what* the business does. Further analysis leads to the business object model revealing *how* business processes are performed. In that way, system designers familiarise themselves with the problem at hand, while at the same time they reach a good level of understanding regarding how users perceive the system to be developed. In addition, a mutual apprehension of objections, suggestions and proposed solutions is achieved, facilitating productive communication. A generic voting model is described in section 2 of this paper. We have merged the business use cases and the business object models, in order to keep its size acceptable, without limiting its value.

Subsequently functional requirements are identified. This is actually equivalent to finding and describing the use cases *the system* will perform. A typical high-level use case description consists of the following:

- *Use Case*: The name of the use case.
- *Description*: A high level narrative description of the use case.
- *Purpose*: The goals actors achieve with that use case.
- *Related Business Use Cases*: The business use case(s) from which each system use case has been derived. It is not always necessary to have related business use cases, since the system may introduce additional functionality to the currently supported one.

- *Actors*: The actors participating in the use case (actor is the coherent role a 'customer' of a use case plays when interacting with a use case) [13].
- *Type:* We categorise use cases as *primary* (major system functions), *secondary* (minor or rarely used system functions) and *optional* (system functions that may not be implemented).
- *Preconditions*: The conditions that must be met in order for the actor to be able to perform the use case.

As use case descriptions tend to become more detailed, the underlying principles and conditions that should be satisfied are clearly stated thus becoming non-functional requirements.

We present the set of requirements for a secure electronic voting system in section 3. It is to be noted that system use cases tend to coincide with the business use cases identified in the domain model. This is natural since current functionality is not altered by the introduction of an electronic system. Professional roles identified in the business object model are categorised to those who will become users of the electronic system and to those whose actions will be substituted by the system.

During the requirement elicitation process, the language of the problem domain – rather than a strict computer science terminology – are those that should be used to describe and validate the results of the requirements capture process, while modelling conventions (i.e. drawings for describing concepts) should be kept to a minimum. These factors have driven both the creation of the voting model and the requirements capture process.

3. VOTING MODEL

Although the process of voting may be generally visualized in the context of an electoral procedure, e.g. general elections, we have identified four different areas where voting plays a central role:

1. *General Elections.*
2. *Internal Election Procedures* (e.g. trade unions' elections, etc.).
3. *Decision-Making* (e.g. Referenda, etc.).
4. *Polls* of indicative or advisory nature.

The above procedures are organised and conducted in a similar way, although - normally - different legal/regulatory frameworks govern them. We can nevertheless argue that general elections, which is the broadest and most complicated procedure, is a superset of the others, even though specific activities may be differentiated.

We will develop a voting model focused on the general elections process. The level of detail of the voting model is generic enough to be applicable to at least the European Union Member States, although in some cases slight variations may exist, mainly due to differences of the applicable legal framework. Such variations do not affect either the completeness or the correctness of the model.

The general election process - at least in the context of the European Union - is almost synonymous to democracy. Despite the variety of electoral systems[2], legislative framework, and infrastructure, the following principles pertain elections in all member states.

- **Generality:** All citizens, unless otherwise stated by adjudication, above a certain age have the right[3] to vote. This means that:
  - Participation in the voting process can always be confirmed.
- **Freedom:** Everyone is free to vote for the party he/she considers more appropriate. The voting process is thus organised in a way that ensures:
  - Uncoercibility.
  - Ability for - consciously - non-valid vote.
- **Equality:** All votes are considered equal. The voting process is thus organised in a way that secures:
  - *Eligibility*: Only eligible voters can vote.
  - *Un-reusability:* Each eligible voter can vote only once.
  - *Un-changeability/Integrity*: No one can duplicate his or someone else's vote, or change someone else's vote.
  - *Verifiability*: The voter or his representatives should have the possibility to verify that his vote is calculated in the final tally.
  - *Accessibility:* Voters should have indiscriminating access to the voting infrastructure.
- **Secrecy:** None of the actors involved in the voting process should be able to link a ballot to a voter. This means that:
  - *Registration*, *authentication* and voting are evidently separated.
  - Votes are *validated* separately and independently from voter authentication.
- **Directness:** Electors select directly their representatives, meaning that:
  - No intermediaries are involved in the voting process (i.e. no person can be authorised to vote for another person).
  - Each and every ballot is directly recorded and counted.

These attributes pertain the business use cases - and their realizations - comprising the business use case model for the general elections voting process. The model does not cope with the mechanisms employed for determining the candidates or the participating criteria for voters. We will consider that candidates have been appointed and information about the

[2] These can be generally categorised as: a) plurality-majority, b) semi-proportional, and c) proportional.

[3] In some member countries: the obligation.

entire population - whether valid to vote or not - is available. The business use cases included in our voting model are depicted in *Figure1*; they are:

1. Define Election Districts: This process is more or less independent of a specific election. It is performed before the beginning of the election process, in order to define the districts and the corresponding number of candidates that will be represented in the government - according to the number of respective electors. It involves one or more state employees and is generally realized through the following steps:

a) The state employees acquire the official census results.
b) According to the distribution of the population the state employees define the election districts for the current election.

2. Determine Electors: This process is essential for determining the electors for the current voting process. In general, all persons above a certain age have the right/obligation to participate in the election process, as stated above. It is realised by state employees through the following steps:

a) The state employees acquire the official census results.
b) The state employees check each person's age and legal status. Persons over a certain age are included in the elector list, unless convicted to attainder or excluded by judicial judgment.

3. Provide Authentication Means: This process is performed to provide the electors with authentication means, and to allow them to identify themselves during the voting process. The responsibility for the provision of authentication means can be either with the state or the elector. The process ends after voters have acquired the required authentication means in a non-discriminative way. In some countries this process is not performed, since voters can use their identity card or passport to vote. In general, the process involves state employees and electors, and the flow of events is as follows:

a) The state employees create authentication means for every elector.
b) The authentication means are either sent to the voters by the state, or the voters are obliged to receive them from the local state authority.

4. Set-up Election Centres: This process is performed after elections districts have been defined and before the voting event. Its goal is to provide the essential infrastructure - in means of people and equipment - to allow for the execution of the election process. During this process the staff, along with individuals authorised to supervise the process, for each election centre is specified. The process is to be performed by state employees as follows:

a) The state employees appoint certain public places as election centres.
b) They also appoint a person - usually a judicial - as the supervisor of the election centre.
c) A number of other people - usually citizens - are appointed to support the election centre and the tallying process.

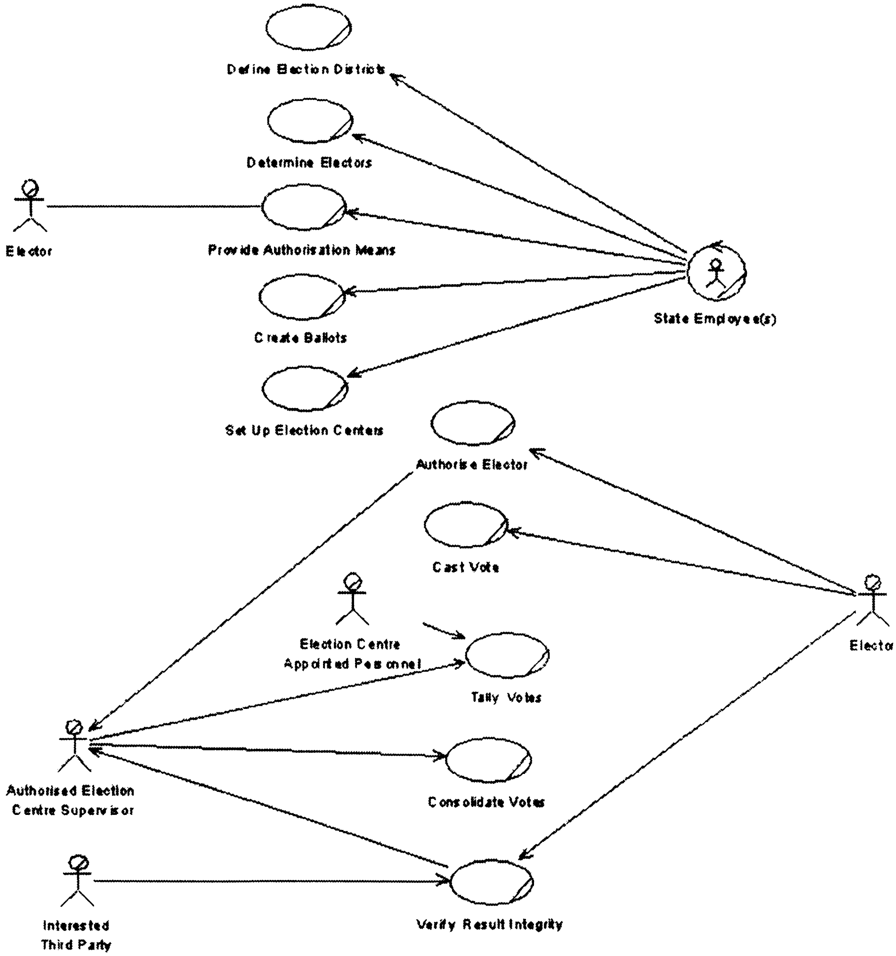

Figure1. System Use Cases for a general elections voting model.

5. Create Ballots: This process starts after elections districts have been defined. Each party provides a discrete ballot format and a list of representatives per election district. The state creates the ballots and supplies them to all election centres. The steps taken for realizing the use case are:

a) Each party representative provides the state with the list of candidates per election district and the ballot format for the party.

b) The state employees create the ballots per eligible district and provide election centres personnel, as well as all authorities responsible for the election process, with them.

6. Authenticate Elector: This process is performed when the elector appears to vote in the election centre she is registered to. It is performed in order to ensure that the elector votes herself, according to the directness principle. It involves the elector and the personnel of the election centre and is performed as in the following way:

a) The elector visits the election centre she is appointed to.
b) She asks the personnel of the election centre to allow her to vote.
c) The personnel authenticate her using the provided authentication means.

7. Cast Vote: This process is performed after the voter authentication. She casts her vote, in a way that protects secrecy, and the authorised individual records are updated. An immediate result of the equality principle is that the voter is not allowed to vote again for that election. The process involves the election centre supervisor and the elector and is performed as follows:

a) The authorised election centre supervisor provides the elector with all ballots for the corresponding election district. He ensures that the sequence of the ballots is random for each elector in order to avoid favouring a specific party. Party representatives supervise this step to verify both expectations.
b) The voter recedes in a private area of the centre and chooses one ballot as his vote. The vote is cast in such a way that its contents are concealed.
c) The election centre personnel update the participation records.
d) A receipt, confirming that the voter has voted, is provided.

8. Tally Votes: This process is performed to validate votes and determine the number of votes each participating party has received, along with not valid votes. The process takes place after the end of the election, in every election centre, and finishes when all votes have been directly validated and tallied by the election centre personnel:

a) The supervisor, with the help of the election centre staff and under the supervision of parties' representatives, opens and validates each vote.
b) Valid votes are counted and added to the results of the election centre.
c) After all votes have been tallied, their number is compared to the number of electors who have cast a vote at the election centre.
d) The result is forwarded to the appointed state authority and added to the election poll.

9. Consolidate Votes: This process aims to consolidate tallied votes (along with the list of persons that have voted in an election centre) from election centres to a central repository. The process starts independently for each election centre, after the tallying has finished, and it involves:

a) The election centre supervisor, with the help of state employees, runs the process. She transfers the votes and the participation record from the election centre to a central state repository. The votes are kept there for as long as the corresponding state law designates.

10. Verify Result Integrity: This process takes place in case a voter - or any other party interested - requests to verify that any of the aforementioned election procedures has been conducted properly. In all cases, the state using the records kept during the corresponding procedure should demonstrate that fact. Thus, the process involves the persons issuing the request, the state authorities, and persons having participated in the procedure in doubt:

a) A voter, or any other interested party, makes a request to verify that election procedures have been conducted properly.

b) The state employee retrieves the record of the procedure in question and demonstrates the steps followed during the procedure and their outcome.

4. E-VOTING FUNCTIONAL REQUIREMENTS

As we have stated before, the traditional model provided the basis for the e-vote system requirements specification. In accordance with the business use cases of the model we have identified the following system use cases:

Authorize Actor: This use case is the starting point for any interaction with the information system. It is a general use case specialized for organizers, i.e. "Authorise organiser" and users i.e. "Authorise user".

Purpose	Provide access to the system functions that the actor is authorised to perform.
Related Business Use Case(s)	6
Actors	All
Type	Primary
Preconditions	None

Manage Election Districts: This is expected to be a rarely used use case, since election districts usually remain unchanged.

Purpose	Create, view and modify different sets of election districts for one or more election procedures.
Related Business Use Case(s)	1
Actors	Election organiser
Type	Secondary
Preconditions	The actor has successfully completed the authorisation procedure (use case "Authorise Actor"). No election procedure is currently in progress.

Manage Election Units: Election unit is the system counterpart of the "election centre". In the conceptual model the election centre is central to the election procedure, as it is the fundamental tallying point. Besides, in case electronic voting is performed in a controlled environment, with machines provided by the state, all votes cast from a certain "electronic" election point should be able to be traced back to that point.

Purpose	Create, view and modify election units for one or more election procedures.
Related Business Use Case(s)	1, 4
Actors	Election organiser
Type	Primary
Preconditions	The election district where the election units will belong has been defined in the system. The actor has selected the selection district where the selection units belong. No election procedure is currently in progress.

Manage Electors: The fundamental assumption of this system use case is that almost all citizens above a certain age should be able to participate in the election procedure. It is practically infeasible for election organisers to manually enter all electors into the system. The system should thus be able to import an electronic list of electors.

Purpose	Import, insert, view and modify electors for one or more election procedures.
Related Business Use Case(s)	2
Actors	Election organiser
Type	Primary
Preconditions	The election district where electors belong has been defined in the system.

Provide Authentication Means: A fundamental requirement of this system use case is to be able to anticipate for a range of contemporary, along with future, authentication mechanisms.

Purpose	Create and distribute authentication means to electors.
Related Business Use Case(s)	3, 6
Actors	Election organiser, elector
Type	Primary
Preconditions	The elector(s) exist(s) in the system

Manage Parties: This system use case is purely operational and is not directly linked to any business use case of our voting model.

Purpose	To notify the system about candidate parties for an election
Related Business Use Case(s)	-
Actors	Election Organisers
Type	Primary

Purpose	To notify the system about candidate parties for an election
Preconditions	No election procedure is currently in progress

Manage Candidates: This system use case extends the previous one.

Purpose	To insert, modify and delete a party's candidates for a specific election district
Related Business Use Case(s)	-
Actors	Election Organisers, Party Representatives
Type	Primary
Preconditions	The candidate's party exists in the system. No election procedure is currently in progress.

Preview Ballots: This use case provides the ability to anyone to preview the electronic ballots for any election district.

Purpose	Create sample ballots for the election
Related Business Use Case(s)	5
Actors	Election Organisers, Party representatives, others
Type	Primary
Preconditions	The ballot icon for the parties and all candidates have been inserted in the system

Provide Party Info: This is an optional use case that can be available either before the voting procedure or during the actual voting.

Purpose	Provide Information about candidate parties
Related Business Use Case(s)	-
Actors	All
Type	Optional
Preconditions	Candidate parties exist in the system.

Cast Vote: This use case has drawn particular attention due to the problems it poses with regards to security.

Purpose	Obvious
Related Business Use Case(s)	6, 7
Actors	Elector
Type	Primary
Preconditions	The elector has been authorised to cast his vote (system use case "Authorise Actor").

Tally Votes: This system use case deals with the final tally calculation.

Purpose	To augment the election result
Related Business Use Case(s)	8,9
Actors	Election Organiser
Type	Primary

Purpose	To augment the election result
Preconditions	The election procedure has ended.

Verify Result Integrity: This use case serves the requests for the verification of the procedure integrity.

Purpose	Verify that system use cases have been properly performed
Related Business Use Case(s)	10
Actors	Election Organisers
Type	Primary
Preconditions	-

In addition to the above functional requirements, two more system-wide requirements should be also met:

Requirement	Details and Constraints
Logging	All internal system operations must be logged without sacrificing voter confidentiality.
Communications	Information regarding *any* of the above functions should be private, even if transmitted over public networks.

The set of user requirements described above cover the functional requirements for an electronic voting system, while revealing - at the same time - its functionality.

5. CONCLUSIONS AND FURTHER WORK

In this paper we have identified the need for systematically producing a complete set of requirements specification for electronic voting systems that unifies the requirements imposed by the existing European legal framework, the functionality reflected by the conventional voting procedures, and the required security attributes that the system should exhibit.

We have applied a software engineering methodology for elicitating user requirements specification in a widely accepted format. This has been accomplished through a set of use cases, along with supplementary specifications. We have, thus, conceptualised an e-voting system in its entity, in a way that confines the number of possible subsequent designs, yet does not dictate a particular one.

This requirements specification is the outcome of the first “iteration” of the requirements elicitation process. We are currently validating and enhancing these requirements focusing, also, on non-functional ones and expect to incorporate the outcome of these activities in the system design and development phases.

REFERENCES

1. Internet Policy Institute, *Report of the National Workshop on Internet Voting: Issues and Research Agenda*, March 2001, available at http://www.internetpolicy.org/research/e_voting_report.pdf
2. The Swedish Government, *Internet Voting – Final Report from the Election Technique Commission*, 2000, available at http://www.justitie.regeringen.se/propositionermm/sou/pdf/sou2000_125.pdf
3. R. Cramer, M. Franklin, B. Schoenmakers, M. Yung. "Multi-authority secret ballot elections with linear work". In *Advances in Cryptology*-EUROCRYPT '96, Vol. 1070 of Lecture Notes in Computer Science, pp. 72-83, Berlin, 1996. Springer-Verlag.
4. B. Schoenmakers. "A simple publicly verifiable secret sharing scheme and its application to electronic voting". In *Advances in Cryptology*-CRYPTO '99, Vol. 1666 of Lecture Notes in Computer Science, pp. 148-164, Berlin, 1999. Springer-Verlag.
5. California Secretary of State B. Jones, *A Report on the Feasibility of Internet Voting*, January, 2000 available at http://www.ss.ca.gov/executive/ivote/
6. CyberVote (IST-1999-20338) project, *Report on electronic democracy projects, legal issues of Internet voting and users requirements analysis*, public deliverable, available at http://www.eucybervote.org
7. Jacobson I., Booch G., Rumbaugh J., *The Unified Software Development Process, 1999, Addison Wesley.*
8. Rational Corporation, *The Rational Unified Process*, http://www.rational.com/products/rup/index.jsp
9. Jacobson I., *Object-oriented software engineering - A use case driven approach*, Addison-Wesley, 1993.
10. Fowler M., *Use and abuse use cases*, available at www.distributedcomputing.com
11. Simons A., Graham I., "37 Things that don't work in object-oriented modelling with UML". ECOOP 98 Workshop on Behavioural Semantics, Technical Report TUM-I9813, Technische Universitat Muchen, 1998.
12. Larman G., *Applying UML and patterns*, Prentice-Hall 1998.
13. Jacobson I., Booch G., Rumbaugh J., *The Unified Modelling Language User Guide*, Addison-Wesley 1999, pp 457.