Requirement Specifications for Electronic Voting Systems

Patricia M. Pesado
III-LIDI. Instituto de Investigación en Informática LIDI,
Facultad de Informática, Universidad Nacional de La Plata.
La Plata, Buenos Aires, Argentina.
ppesado@lidi.info.unlp.edu.ar

Guillermo E. Feierherd
GITIA. Grupo de Investigación en Tecnologías Informáticas Aplicadas,
Facultad de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco.
Ushuaia, Tierra del Fuego, Argentina
feierherdge@ciudad.com.ar

Ariel C. Pasini
III-LIDI. Instituto de Investigación en Informática LIDI,
Facultad de Informática, Universidad Nacional de La Plata.
La Plata, Buenos Aires, Argentina.
apasini@lidi.info.unlp.edu.ar

ABSTRACT

This paper presents an analysis of the requirements specification in electronic voting systems.
In particular, it poses a specification that assumes a physical distributed architecture model with two networked intelligent units (Voting Terminal and Authorities Terminal). State Transition Diagrams and Use Cases are used in the modeling of the requirements.
Finally, the model adaptation to two classes of different elections is analyzed: a national election of closed daily cycle and a university election with a cycle of several days, both with multiple objectives.

Keywords: Requirement Specification, Electronic Voting, Use Cases, State diagrams.

1. INTRODUCTION

The analysis of the requirements is the basis of Software Engineering: a successful software development is closely linked to the requirements analysis carried out, since in this analysis the macro and micro development objectives are defined: during the analysis stage we must think about the problem to solve, its precise definition, and establish the necessary steps for its solution. [PRE02].
If this specification is not carried out with precision, the expected results may not be achieved. Without doubts, these initial considerations, plus the need of finding parametrizable solutions of great flexibility, turn the modeling of a system into a key tool of a development process.

Leite defines Requirement Engineering as the process through which different points of view are shared in order to compile and model what the system will carry out. This process makes use of a combination of methods, tools and actors, whose product is a model from which a requirement document is generated.
Requirements for a software system determine what the system will carry out and define the operation and implementation restrictions. The importance of properly grasping the requirements not only aims at those functional characteristics of the system, but also at the non-functional aspects such as security and reliability, essential in certain systems to be developed.
A very important point is to choice the most adequate techniques for the specification in the analysis stage and their proper combination so as to reflect the "real world" as precisely as possible.
Among the different modeling techniques we may quote State Machines [SOM02] and Use Cases [JAC99].

- State Machines allows representing the behavior of a system in response of internal or external events. The most used notation for the modeling with this technique is the "State diagram" (SD) defined in the standard UML [SOM02]. The State diagram shows the possible states that an object may take, the events that trigger the transition from one stage to the next, and the actions resulting from each change; reason why these states are really useful for representing objects with dynamic behavior [FOW97]

- Use Cases (UC) are a convenient way of representing the functional requirements of a system, since each of them may be assessed without knowing in detail the subsystem containing it. In this way, we can break the system up into a collection of use cases with low interrelation among them, which allows the requirements traceability and realistic estimation of the analysis and coding times. [PFL02]. In addition, it is a convenient tool for the users when they must validate the system, allowing each actor to verify the UCs in which they take part, without the need of knowing more details about the system.

2. ELECTRONIC VOTING AS SOFTWARE ENGINEERING PROBLEM

General Aspects

An electoral system is an information system entailing from the voters' registry lists to the scrutiny, and the addition of individual decisions. The voting instance (the exact moment in which the elector expresses his/her decision and to which the idea of *electronic voting* is specifically referred) constitutes just one of its subsystems. This is why, like in any information system, it is unavoidable to begin with an analysis and determination of the requirements to be fulfilled.
This analysis will also show that voting systems can be considered as *critical systems* [HUM89] because votes are generally translated into political power. This is the main reason why the precision and quality of their quantification must be carefully considered.
Thus, independently of political matters, in order to establish the questions that should be posed (together with the voting method, human interface, time requirement, control actions), a fundamental requirement of the system is ensuring that the counting of votes is carried out with exactness and in such a way that there exist no doubts about its reliability and, if there is any, it allows eliminating them, eventually recurring to alternative mechanisms.
In the case of political authorities elections, the National Constitution and the enforcing laws (electoral acts or public consults or popular referendums) establish four fundamental requirements or characteristics of the vote [FEI04]:

- Universal (all the citizens fulfilling certain conditions are enabled to vote, and only them).

- Equal (all the citizens composing the election universe must be enabled to vote only once, and all the votes have the same value: one citizen, one vote)

- Secret (it must be ensured that the identity of the citizens cannot be related, in any way, to the cast vote).

- Mandatory (the citizen must compulsory vote).

Some Provincial Constitutions of Argentine add other requirements, which are detailed in [FEI03], and they all generally express the need of a public and immediate scrutiny at each polling station when the election has finished.
Other requirements, likely to be qualified as non-functional, correspond to the category of expected or implicit: the system must be ***flexible*** (capable of adapting to different types of elections), ***auditable*** (from the perspective of different software levels or *white box auditing*, and of the results of each polling station or *black box auditing*), ***friendly*** (the system should ease its use even to those who are not accustomed to using computer tools), and ***reliable*** (available, trusted, secure, and protected).
Although the electing act has its predominant point the day/s of the vote, there exists a large quantity of tasks which should be carried out in order to ensure its efficiency, transparency, security, and auditability.
It is thus convenient to break the electoral process up into three well defined stages: pre and post election processes and the election in itself. These three processes *are present in any electing model.*
The pre-election processes should take into account the definition of the election type, its posts, the candidates to the posts, the definition of the computing centers, the geographical distribution of computing and voting centers, constitution of voters' registry lists, consulting services and previous surveys , authorities designation, etc.
In the post-election processes, the collection of partial results and the determination of the winning candidates, among many other activities, should be carried out.
On the other hand, the election stage in itself may be subdivided into three sub-stages:

- Election initialization, during which the authorities of the polling stations are to check whether the ballot box is empty, verify the validity of the voters' registry list and of the candidates to the posts, seal the ballot box and issue the Start Act.

- Voting stage, during which the authorities must check the identity of the voters, their correspondence with the voters' registry lists, and make sure, once is able to cast the vote, he/she has completed the process.

- Counting of ballots, to be carried out once the cast of votes has finished, process during which the authorities of the balloting station

must proceed with the opening of the ballot box, the scrutiny of the votes, the systemization of the results, and the issuing of a closing act, which is generally informed to the corresponding computing center.

Types of Elections

It arises from the analysis of different electing processes that they can be classified as follows:

- From the operative point of view, there exist elections of "*closed daily cycle*", which begin and end with no interruptions, generally in one day, including the initialization, voting, closing of the balloting station, opening of the balloting box and scrutiny. Another model is that of "*several days cycle*", which is developed with partial closings of the voting periods, without scrutiny, and a final closing in which the total scrutiny is carried out.

- From the functional point of view, we may find elections with *a single objective* (for example, an election exclusively of a presidential formula or a popular consultation for YES or NO - plebiscite), or with *multiple objective* (for example, election of national legislators, provincial legislators, and school counselors) which may have conditional enablement for the electors (for instance, foreigners).

- Finally, from the point of view of the selection of the candidates, there exist variants to the classical listing systems. Among them, we can mention those of *preferences* or *strike-through lists* (tachas), which add the complexity to the voting operation and, above all, to the voting counting stage.

The idea of performing a parametrizable software for the different types of election is a task more complex than that of a precise solution for a type of model, though it presents the advantage of carrying out just once the *white box auditing,* i.e. the auditing of all the software levels.

3. PHYSICAL SYSTEM ARCHITECTURE OF ELECTRONIC VOTING

This paper assumes a model of physical architecture (in fact, that used in the experiences later detailed) with two networked intelligent systems:

- Voting Terminal (VT), in which the elector finds the options for casting his/her vote. It must include a set of protections allowing the replacement of aspects of the classical "voting booth" and of "ballot box".

- Authorities Terminal (AT) of the polling stations, which must be an intelligent system allowing controlling the voters' conditions by means of an electronic voters' registry list and tracking the effective cast of the vote, as well as any operative problem in the Vote Terminal.

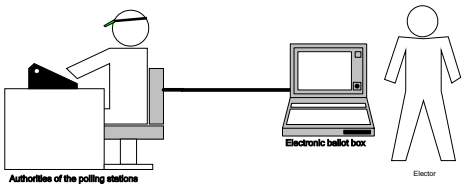

Fig. 1 - System Architecture

This model is common to several electronic voting systems; although in some cases the Authorities Terminal does not exist and the voter receives physical authorization (for example, a card to be entered in the Voting Terminal).
Also, it is possible to find experiences with several networked Voting Terminal, controlled by a single Authorities Terminal. [TUL05]
[BAR04][PES04] details some variants used in various countries of the world and others provided by electronic ballot boxes' manufacturers.

4. STUDIED ELECTING MODELS. ELECTRONIC VOTING USAGE.

We will analyze two cases of elections with multiple objective, one of *closed daily cycle* (a National Election including three political authorities levels), and another of *several days cycle* (University Undergraduate Elections with 2 eligible representatives levels).
Basically, both cases represent quite different elections and will be useful for discussing the flexibility of the used requirement modeling tools.

National Elections

The Electoral Act of Argentine contemplates the general characteristics of the vote expressed in [LeyElec].
The scenario posed by the Electoral Act defines a voting center with a given number of polling stations, for which there exists a group of authorities and the so-called "voting booth". In addition, it introduces a series of steps to follow rigorously for the election process.
In general lines (and without considering the multiple situations of exceptions regulated by the norm), the process is as follows:

- At the beginning of the election, the emptiness of the voting box must be validated, after which a Start Act is drawn up and signed by all the authorities of the polling stations and the poll watchers; then the ballot box is sealed and not opened again until the election has finished.

- The electoral act thus begins. Each elector arrives at the polling stations with his/her Identity Card (IC). The authorities of the polling stations verify the correspondence with the voters' registry list and give the elector an open and empty envelope with which he/she enters the voting booth, selects the ballot of the corresponding candidates, then puts this ballot in the envelope, seals it, puts it in the ballot box, and then gets his IC.

- During the election period, the authorities of the polling stations carry out controls, such as verifying the quantity of cast votes or the quantity of ballots in the voting booth.

- Once the election period has finished, the authority of the polling stations ends the electoral act. Then the ballot box is opened and the counting of ballots is initiated; the Closing Act is drawn up and the results are delivered to the regional centers. Finally, all the votes and the documentation are put in the ballot box, and this is in turn delivered to the authorities in charge of taking it to the regional center.

The software development perfectly reflects all of these steps. In principle, it poses an scenario with the same structure as an "Authorities Polling Stations – Voting Booth", being these elements represented by an Authority Terminal in the authorities polling stations and a Voting Terminal (Electronic ballot box) located in a place which ensures the privacy of the suffrage. When an elector appears at the authorities' polling stations, the president of this precinct enters the number of the elector's identity card in the Authorities Terminal. Once this is finished, the information is verified, and if it is valid (the elector is eligible for voting in that precinct), the Voting Terminal is enabled so that he/she can vote. Once the Voting Terminal is enabled, the elector starts casting the vote. When he/she is casting his/her vote, the Authorities Terminal is disabled. When the elector ends the casting of the vote, and once it is confirmed, the Authorities Terminal is enabled, informing the president of the polling stations about the finalization of the voting process.
In this way, the casting of the vote is ensured. On his part, the president of the precinct holds the elector's identity card until he/she ends the voting process or cancels the attempt to do it.
On his/her part, when the elector is in front of the electronic ballot box, he/she will find the possibility of choosing among the available options for this election. For instance, if it is an election with three types of posts, he/she will opt to vote a complete list (voting the candidates of the same party), cut the ballot (selecting each post in particular, and assigning it a candidate of a particular party), or cast a blank ballot. When the vote is confirmed, the box issues a ticket with its details. This ticket, which can be visualized by the elector, is put automatically in a sealed ballot box. This allows a "black box" auditing of the electoral act.
During the election process, the president of the precinct has access to a series of verification and control operations of the election. For example, he will be able to visualize the total of counted votes until a certain moment, add a voter who does not appear in the voters' registry list, or end the election.
Once the electoral act has finished, the president of the precincts will proceed to the closing of the election. Once the closing is confirmed, a series of results visualization options are enabled. For instance, they will be able to see who the winners of the elections are, as well as the details of the votes for the different posts. Among other options, they will also be able to print the closing acts. It is important to notice that, once the results are visualized, the election cannot be continued.
Once the process has finished, the equipment is turned off and is delivered to the security forces in charge of the transport to regional centers. Only one member of the Electoral Office can reset the equipment with the single purpose of auditing the election.

University Undergraduate Elections

The scenario posed by the University Charter of the UNLP and ruled by the Senior Council for the undergraduate representants' election, defines a period of three consecutive days for all the Schools. Each School is autonomous, reason why the election process may vary, but in all the cases authorities' polling stations and voting booths are to be constituted in order to preserve the voting principles of "mandatory, secret, and universal" [PES03]. In addition, each School sets the number of stations necessary for the voting process (generally, in function of the number and conditions of undergraduates included in the registry lists), the lists of candidates for Students Center and Academic Council representants, the members of its Electoral Office and the lists of the Authorities of the Polling Stations, among other issues.
For each day of the election, a member of the Electoral Office, together with the poll watchers of each participating list, will enable a "new ballot

box" at each polling station, sign the corresponding Start Acts, and assign the polling station authorities. Alike the national elections, the authorities of the polling stations can vary during the day. However, for each change of authorities, the entering and the outgoing will sign the corresponding acts before a member of the Electoral Office and the poll watchers.
Once the ballot box is enabled and the authorities designed, the voting act starts. A student appears at the precinct with his/her student I.C. The authorities verify the correspondence with the voters' registry list and, if it is correct, his/her condition is analyzed. A student can:

- Meet the requirements, which implies that he/she meet the academic regularity and he/she is thus enabled to vote for the Undergraduate Academic Council representants and for the Students Center.

- Not meet the requirements, which implies that he/she does not meet the academic regularity and, thus, is not enabled to vote for the Undergraduate Academic Council representanrs, but is enabled to vote for the Students Center

- Be First-time (Entering) student and also
    - Meet Condition
    - Not Meet Condition

- Be Double-Registered, i.e. he/she is a student of more than one School of the UNLP. In this case, the student will have to choose in the Rectorate the Academic Unit in which he/she will cast his/her vote for the Undergraduate Academic Council representants (only at one School). However, the student is enabled to vote in all the Schools for the Students Center.

Once each day of elections has finished, the ballot box is sealed and kept safe in a sealed cabinet until the complete finalization of the election period. Once this period is finished, the ballot boxes are opened as obtained and the votes are counted. Finally, the corresponding acts are signed and the results are delivered to the computing center of the Rectorate of the UNLP.
Like in the national elections, the software development will reflect the steps previously mentioned. The scenario is kept, with the Authorities Terminal and the Voting Terminal (Electronic Ballot Box) located in such a way to ensure the privacy of the vote casting. Once the ballot box is initialized, the member of the Electoral Office and the poll watchers ask for the issuing of the Start Act, which is signed by the authorities of the polling stations, sealed, and the voting process is thus initiated.

A student appears with his/her student I.C. before the authorities' polling station. These, in turn, enter the number of student in the Authorities Terminal, which informs about the student's personal data and condition. If he/she is Doubled Registered, they will ask the student for the certificate issued by the Rectorate. Once the student is enabled, he/she places him/herself in front of the ballot box in which he/she will see the options to vote. Like in the case of national elections, the Authorities Terminal will be disabled during the student's voting process. The machine issues the printed ticket which will be put in the ballot box for later auditing. During the day, the authorities of the polling stations will be able to verify the functioning of the ballot box by means of control functions, and the member of the Electoral Office will be able to change the president of the precinct, among other functions.
Once the day has finished, the member of the Office ends the daily task without visualizing any result and the equipment is turned off. When it is turned on again, the equipment is ready to start with a new day of election.
Once the period of elections has finished, the members of the Electoral Office proceed to the definite closing, the closing act is printed, and the partial and total results by election day are visualized. Then the equipment is turned off, which will remain in finalization state for a potential auditing.

5. MODELING WITH STATE DIAGRAM AND USE CASES

Both modelings share the fundamental state of: "initial state", "election state", "voting state", "results obtaining state", among others. The basic difference of this model lies in the system behavior in response to the event of ending an election day.
In the case of the elections of *closed daily cycle*, the subsequent events are related to states inherent to the results' control and obtaining. In the case of *several days cycle* elections, this process may lead to a re-initialization of the voting process for a new session or, in the definite closing, for the obtaining of results.
Another point modeled through state is the equipment power loss. If this happens abruptly, when it is restarted, and after authenticating the users, the ballot box will be in the same state as before losing the connection.

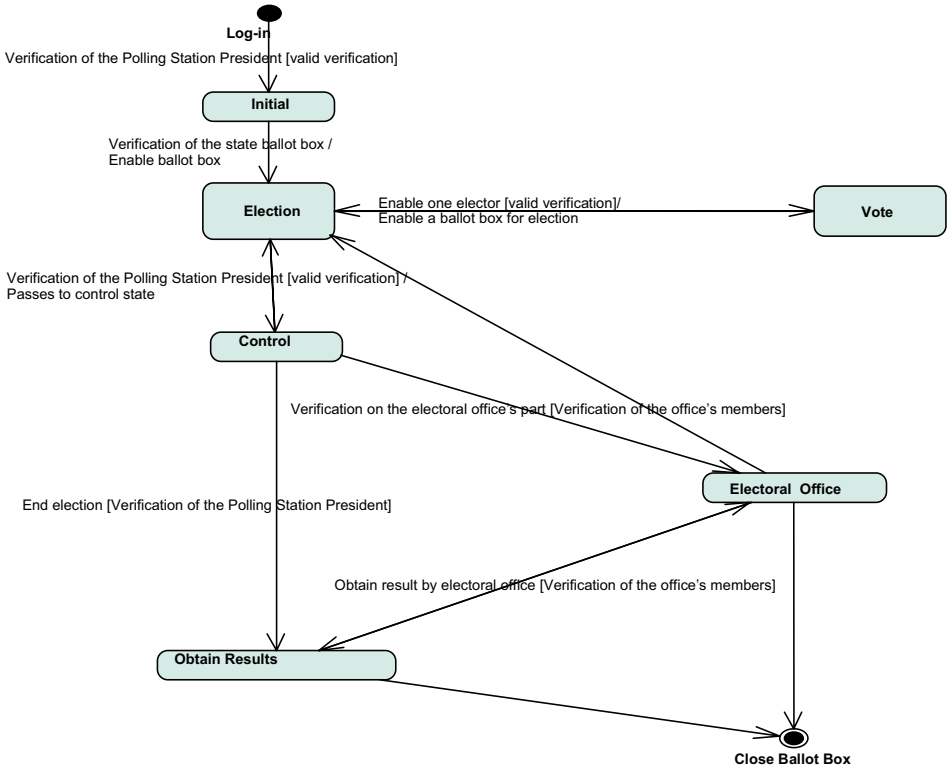

Fig. 2 – National Elections

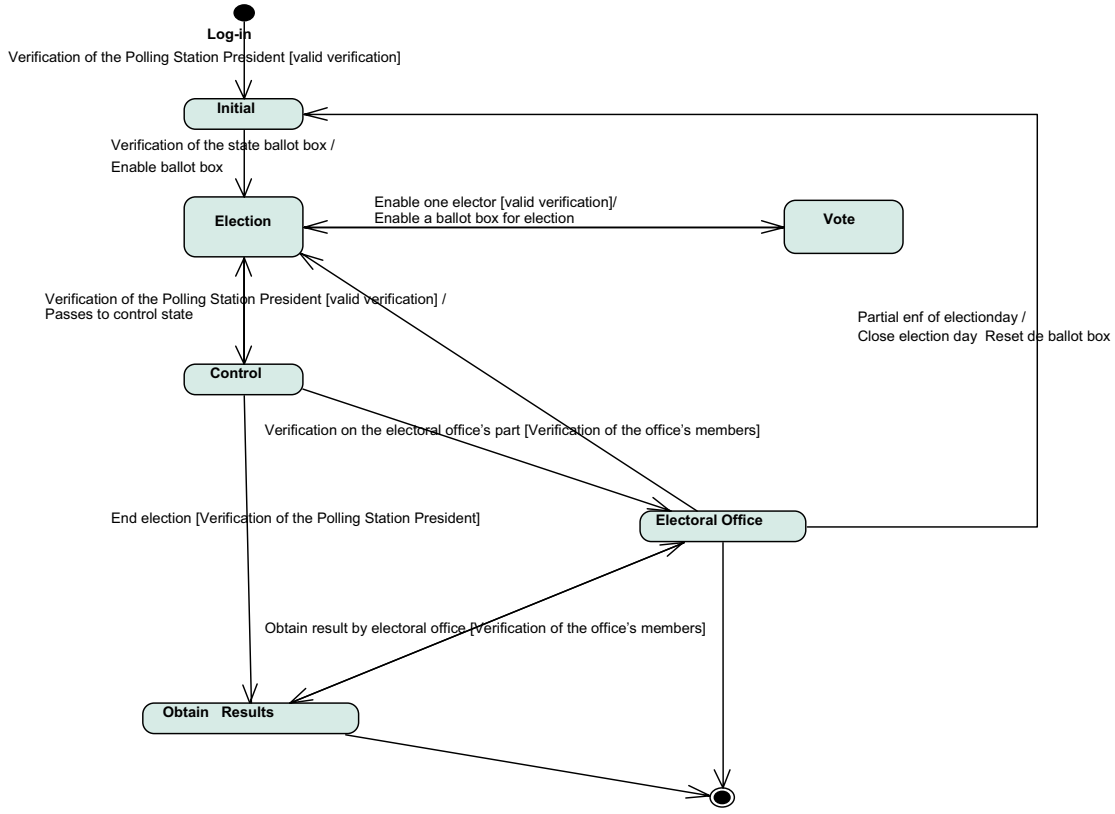

Fig. 3 – Undergraduate University Elections

On the other hand, within each state, its functionalities are modeled using Use Cases (UC). The use of these tools allows for flexibility when the election model is to be modified. On the one hand, the functionality associated to the elector enablement is clearly contained and encapsulated in the "Enable Elector" Use Case, with which the modification is confined to this point. On the other, this type of specification allows modifying the actor responsible of the different processes without modifying the process characteristics, as reflected in the UC "Verify Voters' Registry List", "See Totals", "Verify Candidates", in one case the responsible is the President of the Polling Stations (national elections) and in the other, the Electoral Office (university elections).

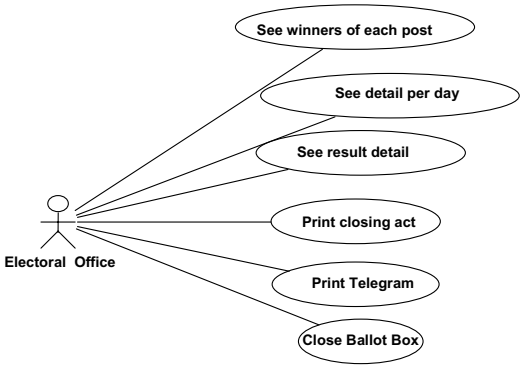

Fig. 4 – State "Obtain Result " National Elections

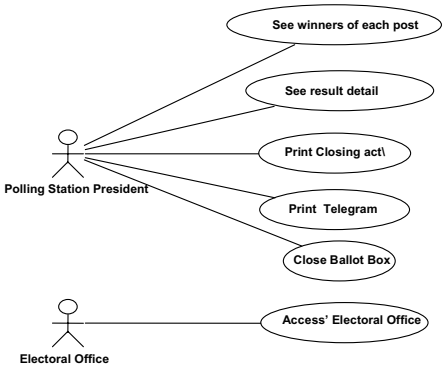

Fig. 5 – State "Obtain Result " Undergraduate University Elections

Specification through use cases allows the natural description of normal and abnormal process flows. For example, the elector's time limit in the voting booth. If the elector spends too much time in voting, he/she will be asked if additional time is needed: if this is the case, he/she is granted with another time period, and if there is no response on the part of the elector, after thirty seconds, the voting attempt is canceled. If the president of the precinct ends the election by mistake and the results are not yet visualized, a representative of the Electoral Office can turn the ballot box again into election state.

Apart from being a significant tool from the point of view of requirement analysis, which then simplifies the software design stage, the use cases can be used as a communication tools towards clients, since in them all the software functionalities will be perfectly reflected in a language accessible to the client.

6. RESULTS OBTAINED

The software for the Electronic Voting Ballot Box prototype has been developed from the specification. In the development, C programming language was used in a Linux operating system modified for the architecture prototype mentioned in point 3. In addition, graphical multiplatform libraries were used, which allows the program itself to be compiled over the prototype or over a PC with Windows operating systems in simulator mode.

In the development of the prototype, free software was used in order to allow the transparency and auditability of the source program at all levels. Vote printing was also implemented as a security measure for a potential post-election auditing and in order to fulfill the requirements of the Electoral Act of our country and the University Charter, and the regulations of the different Faculties.

After ten months of work, the development is operating in a simulation and running version over an electronic ballot box prototype.

7. LINES OF FUTURE WORK

We are currently working on the evolution of a prototype, considering more "models" of election (with preferences, strike-through lists –*tachas*-, plebiscites).

We are also attempting to cover part of the pre-electoral phase, with a definition of a series of steps allowing the necessary configuration according to the different types of election and the different initialization and distribution methodologies of ballot boxes.

On another line, we are analyzing the possibility of using the same Authorities Terminal for the enablement of several Voting Terminals, thus reducing the number of authorities of the polling stations necessary to carry out an election.

Another future line of work is related to the post-electoral stage, in which each ballot box can be connected by a safe means with a regional computing center, easing the voting center data delivery for the votes computation.

8. CONCLUSIONS

The combined use of the techniques of "State Machines" and "Use Cases" in the requirement analysis of this problem was of utmost importance in the definition of a concrete objective and strengthened the re-utilization for other election variants.
The implementation, through a simulation scenario or with different prototypes, is transparent to the requirement specification carried out. The prototype software verification, carried out by the testing data built upon use cases, allowed a complete analysis of reliability and response to the system requirements.

9. BIBLIOGRAPHICAL REFERENCES

[BAR04] "Análisis de Urnas Electrónicas"– Barbieri A., Pasini A., Estrebou C. – Reporte Técnico III-LIDI Facultad de Informática UNLP - Febrero 2004

[FEI03] "Análisis del voto informatizado en Tierra del Fuego" –Feierherd G. – Reporte Técnico Facultad de Ingeniería UNPSJB – Sede Ushuaia – Agosto 2003 –

[FEI04] "Una aproximación a los requerimientos del software de voto electrónico de Argentina" – Feierherd G., De Giusti A., Pesado P., Depetris B. - I Workshop de Ingeniería de Software y Bases de Datos – Octubre 2004

[FOW97] "UML gota a gota" – Fowler M., Scout K. - Editorial Pearson, 1º Edición. 1997

[HUM89] "Managing the software process" - Humphrey W. 1989.

[JAC99] "El proceso unificado de desarrollo de software" - Jacobson I., Booch G., Runbaugh J. 1999

[LeyElec] Código Nacional Electoral Decreto 2135/83 – Ley 19.945 / 20.175 / 22.838 / 22.864 y sus modificatorias.

[PES03] "Voto Informatizado en la Facultad de Informática UNLP" – Pesado P., De Giusti A., Pasini A., Estrebou C. – Reporte Técnico III-LIDI Facultad de Informática UNLP - Diciembre 2003

[PES04] "Análisis de Requerimientos de LVM Urnas Electrónicas"– Pesado P, Pasini A.– Reporte Técnico III-LIDI Facultad de Informática UNLP - Febrero 2004

[PRE02] "Ingeniería del Software" – Pressman R., Editorial McGraw Hill, 5º Edición.

[PFL02] "Ingeniería de software. Teoría y práctica" Pfleeger S. L. - Prentice Hall, 2002

[SOM02] "Ingeniería de Software" Sommerville I. - Addison Wesley, 2002

[TUL05] Tula, Maria Inés. "Voto electrónico". Ariel. Argentina, Bs. As. 2005. Paginas 372 a 374.

www.mininterior.gov.ar Experiencias de Voto Electrónico a nivel nacional en Argentina.

www.gba.gov.ar Experiencias de Voto Electrónico a nivel Provincia de Bs. As.

www.buenosaires.gov.ar Voto Electrónico en la Capital Federal. Proyecto.