

PROCESS AND AGILE ENGINEERING METHODS FOR THE INTEGRAL SYSTEMS AUTOMATION

Arturo Carlos Servetto
aserve@mara.fi.uba.ar

Ramón García Martínez*
rgm@mara.fi.uba.ar

Gregorio Perichinsky
gperi@mara.fi.uba.ar

Operating Systems and Data Bases Laboratory
*Artificial Intelligence Systems Laboratory
Computer Science Department - School of Engineering
University of Buenos Aires.

Paseo Colón N° 850 - 4th floor - South wing - (1063) Buenos Aires - Argentina

Phone: (54 11) 343-1177 (int. 140 / 145)

FAX.: (54 1) 331-0129

*Buenos Aires Institute of Technology
rgm@itba.edu.ar

Abstract

This paper presents some original proposals and the advance state of an investigation project that has as objective the development of a tool ICASE for systems of median management complexity.

We intend in this paper to generate software products as of specifications of top-level abstraction graphics, defining for this a development process of evolutionary prototyping based solely on a synthetic behavior and functionality system model and another based in their data structure model. The behavior and functionality model comprises the requirements specification as well as the system analysis and design, that go evolving cyclically according to the process.

For data modeling are used conceptual entity-relationship graphs, and for the functions specification is defined an entity relational algebra, that is extended for hyphens confection, and a QBE (Query by Example) style specification mechanism for simple queries. As of the integration of the two models referred with the functional specifications in their algebra or in the visual mechanism is obtained the complete system code.

In this presentation are described the framework project and the graphic models.

Key Words

ICASE Tools, Agil Systems Engineering, Modeling the System Behavior, Modeling the System Functions, Software Engineering, Processes of Software Development.

Introduction

The methodologies of software development emerged on account of need to control and document the increasingly complex projects, actuated mainly by economically important institutions and with safety and reliability requirements in their extremely strict systems, as the Defense Department of the United States.

But the evolution of these methodologies, those that suggest or impose often several parallel activities for each phase of the life cycle of the systems and that also they can require more that a model for each

activity, determined that the cost and the effort required in producing and maintaining the relative documents to the development process grew considerably, to such point that today only very large institutions or that develop very complex systems adopt them and fulfil formally [1].

The traditional development methodologies model independently three aspects or dimensions on the systems informáticos: structure of the information, functions and behavior. Each aspect require a model and its resulting documents can be multiple in the latest two aspects, and as a rule they are quite costly

of maintaining during the evolution of the systems [2] [4] [7] [8] [12] [14].

In the development of systems small or complexity median systems, the adoption of a formal methodology is perceived as an obstacle more than as an advantage, since that those advantages solely they are doing notorious to long-range, and the primary objectives of any institution are to produce software short-range it minor possible and low-cost to result competitive.

Parallely, the employment of automation tools of the systems development yet results quite limited due to the complexity, diversification and lack of standardization of the methodologies, that prevents that an only tool could cover thoroughly processes of development or to sustain at the same time different methodologies; such is that, for a development attended in all his extension it should take into account several tools that often are not mutually compatibles.

Furthermore, only for important companies or those that develop systems of high complexity results economically viable to employ this type of tools to cover íntegramente the process of development[1].

Currently the greater effort of the systems industry informáticos is addressed to the search of a standard methodology as well as to the standardization of notations and representations of the information to the effects of achieving the integration of development tools[15][16][17].

The consolidation of new technologies as those object oriented, with their rich baggage of models of interfaces and its facilities for the reusability, makes auspicious the development of automation tools integral of the systems development through a process of agile engineering; that is to say, minimizing the quantity of models and integrating it in an only tool. This tool could be maintained without greater costs[6].

Graphic Models

For system analysis is proposed to model their operative stages and use cases in two levels.

The first level model would represent the company functional units whose information system would have to be automated (First Level Requirement Graph).

To each stage is assigned to him the name of the corresponding functional unit and, in implicit form

(by request), the user or actor that has permit to accede. To this level of models the functional structure of a company or institution, the graphs have hierarchic form. In the Unified Software Development Process with UML[5][11], the first level could be associated to grouping packages of use cases.

The second level models, accessible as of each first level stage, represent the automatic functions of the respective unit (Second Level Requirement Graph).

Here the links or transitions between interfaces, that in the terminology of the UML would be boundary classes, represent use cases or operations that generally give content to the destination interface.

The second level graphs synthesize functionality and behaviour and, in concept, through amalgamation of use cases and interaction UML models.

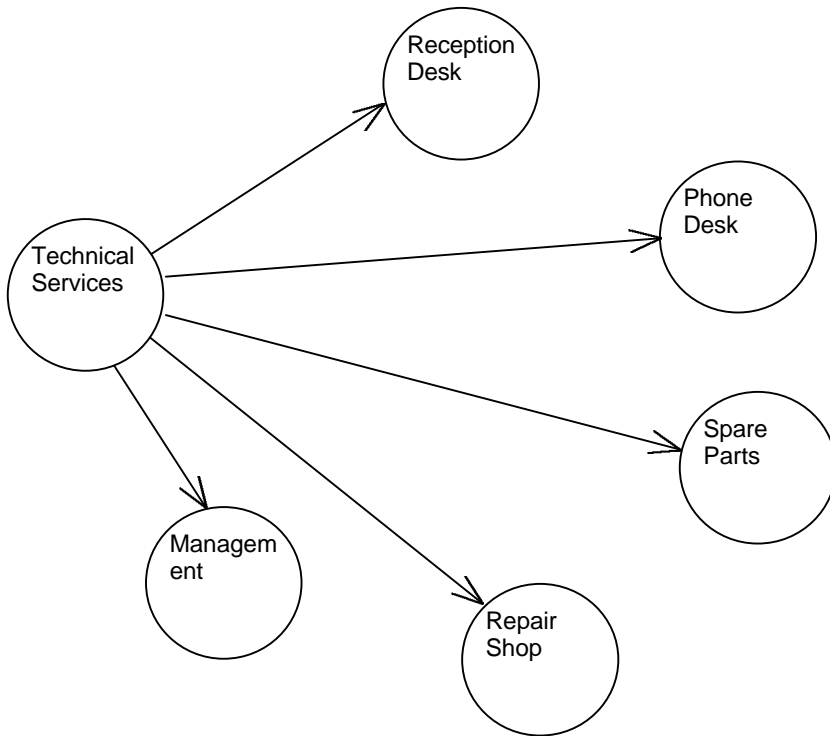
The code refers to the hyphens in relational algebra of entities. Even though the definition of the algebra that is proposed exceeds the framework of this work, fits to emphasize that the operations are defined on entities and relationships maintaining a high abstraction level and providing independence of the code with respect to the implementation of the database.

Study Case

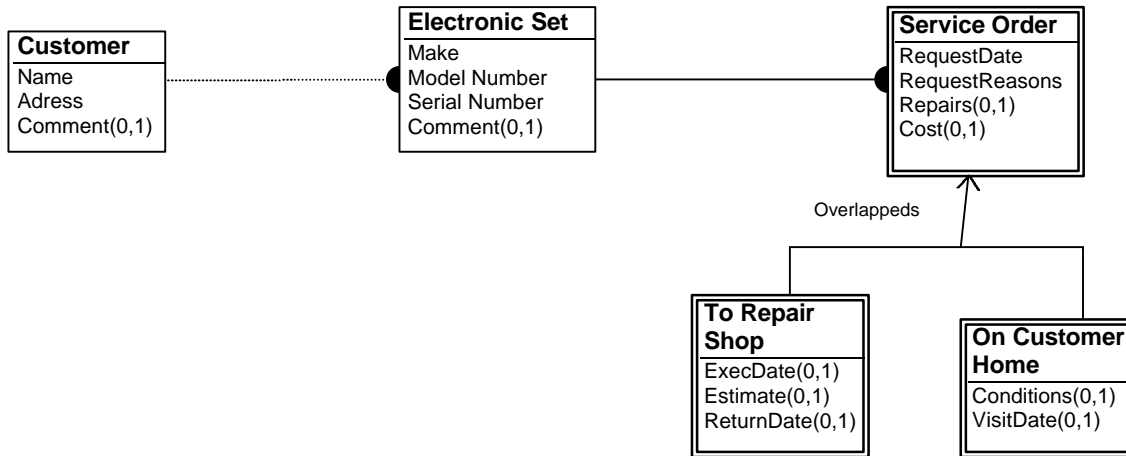
Lets be a company that repairs electronic sets. The technicians make services on customer homes or on the company repair shop.

The company have five functional areas: Reception desk, Phone desk, Spare parts, Repair Shop and Management.

First Level Requirement Graph (general):



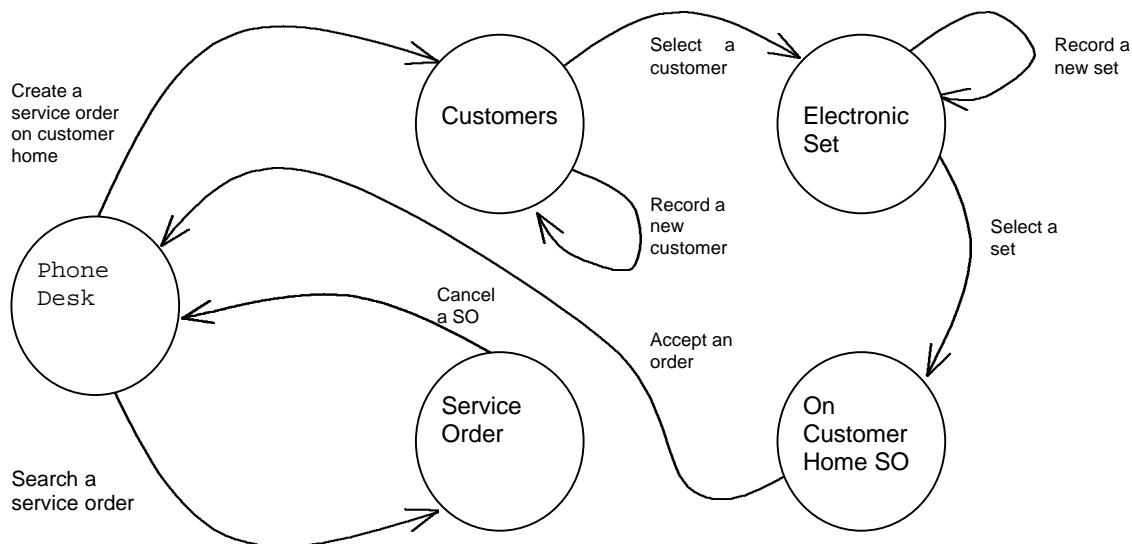
Entity-relationship Graph (from Phone Desk):



Service Order is a weak entity that depends on Electronic Set.

From Technical Services is possible to see the general entity-relationship graph.

Second Level Requirements Graph (from Phone desk):



To create a Service Order on Customer Home, the system opens a dialog window asking data that lead a customer search (the dialog windows dont appear on the models). On user request, the system shows a client list, ordered according to the serch criterion and with the first customer that matches with it on top; then the phone clerk (the user) can choose a customer record or add a new one to select it later. On customer selection, the system opens another dialog window to search a customer's electronic set.

Once selected an electronic set, or added one and then selected, the system shows a service order form.

To search a service order, the system ask the service number and then shows the corresponding form. Then the order can be canceled.

Conclusions

It is hope that this tool represents a great practical importance and a meaningful contribution for the informatics community that it works and researchs in

the discipline into the systems development of small and complexity median, since that implies the adoption of a simple methodology and specifies that favors the participation of the final users and maintains to all the development permanently documented.

The participation and the commitment of the final users in developments based on this tool it is presume guarantee due to the fact that the used models for the specifications are of a high abstraction level and understandable for persons that are not specialized; furthermore, the combination of the model of cases of use, permits verify the completetude and the track of the requirements fulfillment, with the possibility of prototyping for

References

- [1] Perichinsky, G., García Martínez, R., Servetto, A.C., Jimenez Rey, E. and Grossi, M.D. 2002. Integrated Environment of Systems Automated Engineering 3. Product Oriented. (ACIS) Conference on Computer Science. Software Engineering, Information Technology, e-Business, and Applications Institute (SEITI), Central Michigan University. Foz do Iguazú. Brazil.
- Bass, L. , Clements, P. and Kazman, R. "Software Architecture in Practice". Addison - Wesley. 1998.
- Batini, C., Ceri, S., Navathe, S. "Conceptual Database Design: An Entity-Relationship Approach". Addison-Wesley. Spanish, 1991; ISBN 0-201-60120-6.
- Bell, D. and Morrey, I. "Software Engineering, Second Edition".Prentice Hall.1992.
- Booch, Grady; Rumbaugh, James; Jacobson, Ivar. "The Unified Modeling Language - User Guide". Addison-Wesley, 1999.
- Brown, A. W. & McDermid, J. A., 1991, On Integration and Reuse in a Software Development Environment, Software Engineering Environments '91, F. Long & M. Tedd (Editors), Ellis Horwood, Mar 7.
- Constantine, L. and Lockwood, L. "Software for use". A Practical Guide to the Models and Methods of Usage - Centred Design. Addison - Wesley. 1999.
- Ghezzi, C. , Jazayeri, M. , Mandrioli, D. "Fundamentals of Software Engineering". Prentice Hall. 1991.
- Grimaldi, Ralph P. "Discrete Mathematics and combinatorial. Application and introduction". Addison-Wesley. Spanish. 1989.
- IEEE STD 1074-1991 Customer Request IEEE Standard fo developing Software Life Cycles Process, Identify Ideas or Needs, Preliminary Statement of Need, Feasibility Studies, Statement of need.

early solutions, accomplish with the systems generation from the specification of the design of interfaces, optimizes the contractual relationships facilitating the approval of the phases.

Future Projects

As the project recntly enters the development stage, the future work will be the design of the experimental investigation: build measurement instruments, gather and process experimental data, and analyze and evaluate results. In a subsequent stage it is anticipated the amplify of the tool through the aggregate of the versions and repositories administration.

- Jacobson, Ivar; Booch, Grady; Rumbaugh, James. "The Unified Software Development Process". Addison-Wesley ISE. 1999.
- Pfleeger, S. "Software Engineering: Theory and Practice". Prentice Hall. 1998.
- Servetto, A. "Mechanisms of Abstraction of Conceptual Modelling of Data and its Application to the Model of Entity-Relationship." 3rd. Sessions of Informatics and Operative Investigation. University of the Uruguay Republic (12-14 Dec. - 1996).
- Sommerville, I. "Software Engineering". Addison - Wesley. 1996.
- Thomas, I. & Nejme, B. A., 1992, Definitions of Tool Integration for Environments, IEEE Software, 9, 2 (Mar), 29-35.
- Wasserman, A., 1990, Tool Integration in Software Engineering Environments, Software Engineering Environments: Proceedings of the international Workshop on Environments, F. Long (Editor), Springer-Verlag , 137-149.
- Wasserman. A. I., 1988, "Integration and Standarization Drive CASE Advancements, Computer Design", 27, 22 (Dec), 86.