EVOLUTION OF ENTERPRISE INFORMATION SYSTEMS. AN INDUSTRIAL APPROACH

Pietro D'Ambrosio
NET FIRST srl, Salerno, Italy

Keywords: Enterprise Information Systems, Legacy Systems, Evolutionary Maintenance, EIS Management, Software Development Methodology, Enterprise Architectures.

Abstract: The problem of the evolutionary maintenance of large information systems is very critical. The time and costs required to adapt the system to the market's evolutive dynamics are no longer compatible with the objectives of the business. The experimentation, on small-scale, in the laboratory of new technologies and new methods is ineffective for this type of systems and it is rare that the customer accepts the risk of innovations in an industrial project. In this paper we propose an intervention strategy and a reference architecture to transfer, using an approach based on "small steps", innovations obtained from research into new industrial applications.

1 INTRODUCTION

The increasing complexity of today's competitive scenario is frequently associated with social, organizational and cultural phenomena that introduce discontinuities and critical issues in the evolution process of business organizations and their related information systems.

This results in a significant reduction in the life cycle of the services provided by the company and in a growing need for information to be collected and analyzed in order to take effective decisions.

All this leads to a greater workload for the information system, to an increased need of evolutionary maintenance and a progressive decrease in the quality of the system.

Thus, the time and costs of software development are no longer compatible with the real benefits that they bring to the system.

In this paper we introduce a new strategy of intervention and a new architectural model tested in an industrial environment to achieve new levels of effectiveness in the evolutionary development of complex information systems.

The techniques and the methodology adopted, in part derived from the study of the Ultra Large and autonomic systems (IBM, 2006 and Müller *et al.*, 2009), use an approach based on a logic wrapping of the existing information system and on the introduction of new components that enable new features and add new "perceptual capacities" to the system as a whole.

More generally, this approach tends to move the human intervention to a higher logical level in order to reduce drastically the management and development costs.

2 PREVIOUS RESEARCH AND INTERVENTION STRATEGY

The intervention strategy described in this paper derives from a series of research activities on tools and methodologies for software development carried out by some IT companies and university departments (D'Ambrosio *et al.*, 2010).

This work also benefits of a series of experiences gained over the years on industrial projects of various sizes carried out in situations highly critical in terms of continuity of service, level of functionality and security such as banking information systems.

The heterogeneity of application environments and technologies that we have met and the constant need to minimize costs and timing of the project, suggested a policy of innovation based on "small steps", with a continuous monitoring of project risks

D'Ambrosio P..
EVOLUTION OF ENTERPRISE INFORMATION SYSTEMS. AN INDUSTRIAL APPROACH.
DOI: 10.5220/0003414003570362
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 357-362
ISBN: 978-989-8425-55-3

related to the activities or processes involved with the innovations introduced.

This constant attention to identify other small areas of innovation and apply, in real project tasks, what has been experienced in research activities, created in our company an innovation "culture" and a new open attitude that in a mid-long-term period has undoubtedly produced satisfactory results.

In this sense we believe that our experience may provide a useful contribution to define a more general methodology for industrial research activities on complex information systems. In this context, in fact, the laboratory simulations and the demonstration on a small scale performed via prototypes almost always are ineffective and do not provide the necessary certainty requested for the immediate industrial application of technology.

The criticality and complexity of such systems consists mainly in their large size, in the service's continuity requested, in the time and cost needed for any evolutionary intervention and in the choice of the best intervention strategy. All these issues are not really reproducible in simulated environments.

On the other hand, it is unlikely that users decide to introduce significant innovations (unless they are required to do so) since it may affect the continuity of service and increase the cost and time of the project.

In addition, we have to consider the resistances raised by the actors of development or business processes anchored on their consolidated available skills, by the corporate organizational cultures and by the natural needs of the business.

This, unfortunately, has created a growing gap between the availability of new technology and its industrial transfer and has also significantly slowed the evolution of this type of information systems.

Over the years we developed an intervention strategy for gradual introduction of process and product innovations in development activities and maintenance of complex software systems.

This strategy, schematically represented in figure 1, is essentially composed of 5 steps:

- Step 1. Intuition. Discussion around the innovation idea. Research hypothesis formulation. Evaluation of research hypothesis formulated. Structuring of the scientific problem. Verification of the theoretical foundations. Definition of research objectives and verification criteria for all results. Project implementation and development of prototypes.
- Step 2. Evaluation of results and definition of one or more cases of large-scale implementation of the results to understand the theoretical limits and the possible benefits to be obtained. This may result in:
  - Obtaining unsatisfactory results. The process ends here.
  - Obtaining partially satisfactory results. In this case we reaffirm the general principles, redefine some requirement and restart the process again from the step 1.
  - Obtaining satisfactory results. The work continues with the search for a new project that allows us to introduce one or more innovative elements derived from the research.

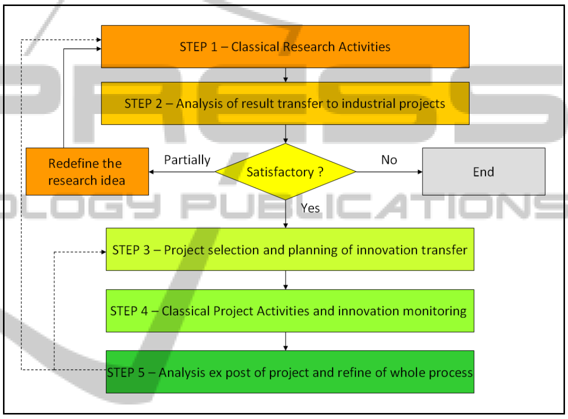

Figure 1: The proposed methodology.

- Step 3. Analysis of innovation in the context of a real project. Identification of intermediate steps with the related risk in terms of: i) delay on the project, ii) implementation strategies (i.e. the need to maintain in parallel the old tools and methods), iii) problems of measuring results. Finding one or more innovation to be applied in the next project and defining the expected objectives.
- Step 4. Development of the Project. Monitoring the activities and achievements.
- Step 5. Post-mortem analysis of the project. Refinement of the results. Depending on the case the process restart from step 3 (on a new project) or from step 1 (on a new research activity). Almost always, the feedback stops at level 3 but it is also inevitable an impact on upcoming activities that start from step 1.

We verified in recent years that on average it takes at least three projects to consolidate any innovation. Moreover, generally the first project to which we apply the innovation has costs and time higher of previous ones. As a result we tend to choose small and not critical project to begin a new experimentation cycle.

The second project has average cost and time comparable with the previous ones. From the third, instead, significant results will be achieved.

3 THE DEVELOPMENT STRATEGY

Almonaies et al. (Almonaies *et al.*, 2010) propose an interesting overview on strategies to modernization of legacy systems towards service-oriented architectures using comparison criteria that take into account different parameters. Among the other, we point out the degree of complexity of the system and the degree of adaptability of the process to the specific type of system.

In terms of conceptual approach, the four strategies identified by Almonaies et al. (Replacement, Wrapping, Redevelopment and Migration) remain valid for the context of the evolution of complex information systems, although in this case, they are not exclusively composed by the legacy applications and are not necessarily obsolete.

In the latter case, the reasons for the evolution of information system can be profoundly different from those that motivate migrating or rewriting a legacy system. Indeed they are mainly oriented to overcome the "unsustainability" of the costs and time for the evolutionary maintenance of the system in a market environment characterized by the acceleration of competitive dynamics and by the drastic reduction of the life cycles of services offered to customers (as in the case of large banking systems).

Among the various modernization strategies, we believe that the less risky, less invasive and more suitable to our case is the "Wrapping". The types of information systems we use as a reference, in fact, consist of systems adopted in large organizations that have a strategic value to the business and generally have a good level of code quality.

Moreover, the aim of our present work is not the modernization of existing components, but their use as part of a more advanced system.

In our case, the wrapping is not made with the aim of improving the existing system but to "encapsulate" it and to extend its functionalities and its "perceptual capacities."

In this way, we can add new elaborative processes to the existing ones, intercept and elaborate the input events and implement new system features.

More generally, this strategy favors, over the time, a gradual and natural shift of system gravity center to the new "application engine".

Our line of action then provides indications for: i) implementing a new architecture (which includes the existing system as "black box"), ii) extending the perceptive capabilities of the system (monitoring the events), iii) increasing capacity of self-control (autonomic-like engine) and, finally, iv) simplifying the development of business logic (SOA and workflow based).

The main goal pursued through these strategies is to move to an higher logical level the human intervention, whenever possible, thinking in terms of events, objectives, rules, logic and workflow services rather than in terms of transactions and functions.

4 THE ARCHITECTURE

The hypothesis of architecture that we proposed is derived from a previous experience carried out under the research project EMAF (D'Ambrosio *et al.*, 2010) and contains essentially principles drawn from cooperating agents architecture, autonomic systems and SOA.

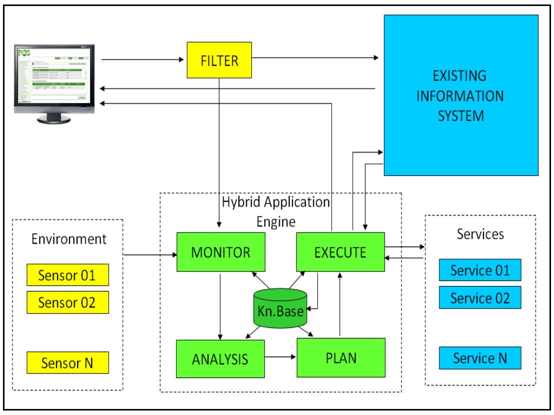

Figure 2: The proposed approach for EIS evolution.

The reference to autonomic solutions is essentially conceptual, since everything has been strongly oriented to the needs of targeting applicative systems.

The architectural scheme, briefly described in figure 2, depicts the following components:

- a network of cooperating agents that monitors specific behaviours and environmental or relational events (Sensors);

- a main component, inspired by autonomic system, that handles the rules and the objectives to satisfy (Hybrid application engine);
- a component of web service orchestration for the implementation of identified actions and for collecting feedback;
- a set of application services for the management of the new features.

In this architecture we will identify at least four conceptual levels at which the logic of the process operates: 1) the agents that monitor events, 2) the rules and objectives of the autonomic engine (dynamically changed by the collected feedback), 3) the workflow system of service orchestrator engine, 4) the specific processing services.

Supporting the development cycle with specific authoring tools, we can "describe" a good part of the process at a logical level higher than classical programming techniques. This results in a significant reduction in time and cost for development, testing, and for the future evolutive maintenance of the system.

4.1 The Filter Component

The filter component in the diagram of figure 2 has been included only to highlight the need to intercept the transaction's input received by the system and translate it into events for the new application engine. In reality this can be done differently depending on how the information system has been created.

The filter can operate in a "transparent" mode with respect of transactions in the existing system or it can intercept new transactions that can be deployed exclusively in the new system.

With respect to existing transactions, if the system includes a "Front Controller" component, the solution coincides with that shown in the diagram.

Alternatively events can be taken from a log file of transactions, captured by a transparent proxy or even through a simple function to invoke before each component of the existing information system.

The latter solution is certainly the most invasive but anyway has a low criticality since it is an intervention that can also be made automatically.

The main objective of this component, however, remains that of "listening" the requests sent to the system and eventually turning them into "events" of interest for the new application engine.

4.2 The Hybrid Application Engine

This is an event-driven "application engine" based on autonomic principles (which follows essentially the structure). In particular, the "Monitor" component collects and controls events sent from the filter or from the sensors network and, if necessary, forwards them to the analysis component.

The sensors network, useful to extend the perceptual capabilities of the system in the "out of home" environments (if is present) is implemented through a system of cooperating agents that process the events themselves and send them to application engine.

The Analysis component will check which business rules apply to the treatment of the event.

The choice of rules can be taken in many ways, especially taking into account a number of state information related to the environment (environment variables) and the feedback received from the execution manager.

In particular, the rules used to decide which processes to run, are of two types: 1) rule "still valid" that are evaluated in a predefined order and, when met, are performed (and therefore more of a rule can be enforced), 2) rules "alternatives" that are evaluated and weighed, with a fuzzy logic approach, calculating a truth value for each of them and choosing the one with the greatest weight (of course only if it exceeds a certain truth value).

The rules can also use environment variables to take into account the state of the system and to contextualize the choices. These rules can be conditioned (in the calculation of the value of truth) by the feedback returned by any processing components.

The "Plan" component is used to schedule the execution of actions, in the predefined sequence defined by the selected rule and controls the outcomes of any action.

The actions are performed using the "Execute" component that interfaces various "effector" to manage the internal services execution, the new applicative features, the feedback and any replies to be sent to user.

5 INDUSTRIAL APPLICATIONS

In addition to the use cases developed as part of the research projects mentioned above, the intervention strategy described in this work was being applied (at different levels) in various industrial projects and in particular for two projects addressed respectively to a banking organization and to a local Public Administration.

5.1 Project 1 – Integration of Banking Applications

The first of these projects was oriented to make an integration between two products for the banking market: an application for managing transactions on POS terminals and a multimedia system for the presentation of geographically referenced data on a choropleth map (using colorimetric scales for observe macro phenomena of interest).

We started the industrial application of these principles with this project because we had the ability to install both products in the development environment and did not have critical deadlines.

The goal of this project was to support the user while interacting with the main application (pos management) with a series of additional information automatically displayed on a second monitor.

Figure 3: Functional integration schema.

In particular, this mechanism can be applied to the analysis and to the statistic functionalities of the system.

When the user selects a set of data or makes an inquiry, the multimedia application managed by the system automatically proposes on the second monitor a visual representations of data using choropleth maps (see figure 3).

The idea is to extend this mechanism to the other applications of the information system in order to support the operator with additional information related with transactions running on the main system.

For example, while the operator performs front-end operations on a specific account, the secondary system can display the photo of the customer, the authorized signatures, alerts or other information to be notified.

Figure 4 shows a scheme of the integration model implemented. Let us point out that the proposed approach produces a substantial improvement to the existing information system without making any invasive change.

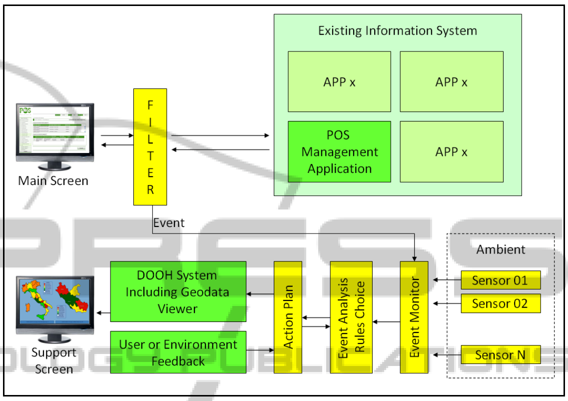

Figure 4: The integration model implemented.

5.2 Project 2 – Multimedia System Enhancement

The second project aims to enhance the functionalities of a web application for the management of interactive digital communication in an urban area.

In this case access to the system takes place predominantly by mobile terminals and local systems for the "digital out of home" (video walls, multimedia totems, LCD panels).

Here the functionality to achieve with the project is the listening of normal requests that are made to the system A (figure 5) and the analysis of user related information (the position in the city, the expressed interests, the history of performed activities).

Based on this information, the system B provides additional information to standard output provided by system A in response to user requests (information, advertisements, notices).

The model of integration of new components with existing components (briefly described in figure 5) remains essentially unchanged from the previous project although they are profoundly different in the application environment, objective pursued and type of improvements made.

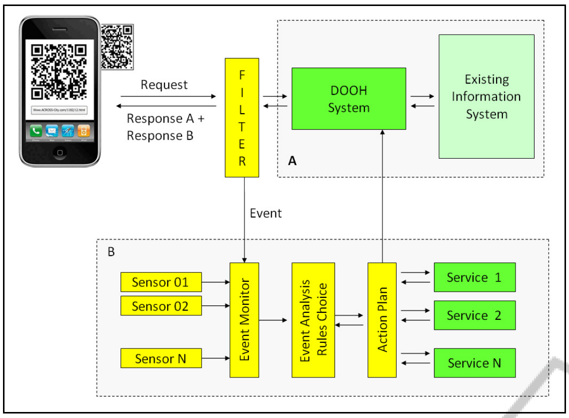

Figure 5: The integration model implemented in the second project.

6 FUTURE DEVELOPMENTS

With the increasing complexity of the scenario and the increasing competitiveness of the market in the near future we expect a significant "discontinuity" in technology and methodology adopted for the information systems development and evolutionary maintenance.

To make information systems more consistent with the emerging functional requirements and to reduce time and cost of their maintenance it is necessary to adopt evolutionary approaches, methodologies and tools that are more suited to new levels of complexity.

More generally, we believe that it is time to seriously consider creating a new level of IT infrastructure, shared among multiple organizations, which can significantly improve the effectiveness of evolutionary interventions on the large enterprise information systems.

This new IT infrastructure can ensure new "levels of transparency" to the system with respect to processes and events, and go further in the direction already taken by the service orchestrator engines and by the event-driven EAI systems.

The overall goal of our approach remains, however, to reduce human intervention in the management of the system and in the evolutionary development, moving this operation on a higher logical level. In this way we can achieve significant benefits without introducing new risks and without compromising the continuity of existing services.

ACKNOWLEDGEMENTS

The research has also been carried out thanks to:

- The "tomato-WAD" project, designed by NET FIRST srl and cofunded by the Italian Ministry of Research. This project aimed to create a software framework for rapid development of complex web applications.
- The “EMAF” project (Measure 3.17-POR Campania 2000/2006), an ICT Cluster Project funded by the Following partners: DMI Unisa, Gruppo Net srl, Net First srl, Acube Lab srl. The project had a number of goals including the introduction of new levels of functionality into existing information systems and the drastic reduction in development time and maintenance of the new components. In addition, this project had defined an architectural pattern, a component of run time and some authoring tools specifically oriented to the introduction of "Digital out of home" features in the traditional information systems.

REFERENCES

D’Ambrosio P., Ferrucci F., Sarro F., Tucci M., “An Approach for the development of Dooh-oriented information systems” - *12th International Conference on Enterprise Information Systems – ICEIS 2010, June 8-10, 2010, Funchal Madeira Portugal.*

Almonaies A. A., Cordy J. R., Dean T. R.. “Legacy System Evolution Towards Service-Oriented Architecture”. *Proc. International Workshop on SOA Migration and Evolution (SOAME 2010), Madrid, Spain, pp. 53-62*

IBM Corporation 2006. “An Architectural Blueprint for Autonomic Computing”, *4th ed, http://www-03.ibm.com/autonomic/pdfs/ACBlueprintWhitePaper4th.pdf*

Müller H. A., Kienle H. M., Stege U., 2009. Autonomic Computing Now You See It, Now You Don't. *In A. De Lucia, F. Ferrucci (Eds.) Software Engineering, International Summer Schools, ISSSE 2006-2008. LNCS 5413Springer, pp. 32-54.*