

8 Conclusion

This conclusion presents a summary of the dissertation and its contributions. Section 8.1 summarizes the main ideas of our work. Contributions and expected benefits are emphasized in Section 8.2. Section 8.3 lists some open issues and outlines directions for future work.

8.1 Summary

Agent-oriented software engineering is a still under development discipline. Nevertheless the success of this new research program comes from the fact that it better meets the increasing complexity and flexibility required to develop software applications built in open-networked environments deeply embedded into human activities. The emergence of a new research program does not nevertheless mean that it will directly be absorbed by the software market. Indeed, it needs standardization, productivity gains, proven efficiency on huge and complex user-intensive software development projects, well-designed development frameworks, etc. to review its standards. The absorption of the object-orientation has also been difficult, but thanks to the benefits this technology offers, it has become the actual market standard.

To impose itself, agent-orientation has firstly to reach unification in concepts and terminology. At the second stage, MAS development methodologies need to adapt and/or adopt other necessary aspects of software engineering and grow to a process able to manage real life huge software projects involving users intensively and not only research case studies. The application of a SE methodology would, indeed, provide very different results when applied for a real life development since the actors involved in a software project have a limited vision of user requirements and the organizational setting inherent to their bounded rationality.

In order to deal with this, SE methodologies have follow modern development life cycles to better capture user requirements in rapidly evolving environmental contexts as the iterative one. Such a development life cycle could however not being conceived without managing the software project. The later needs to be adequately supported when dealing with the increase of complexity induced by the employment of tens of human resources achieving activities in parallel. Development support features are often regrouped under the term *project management*; such a discipline allows supporting requirements elicitation, identifying and evaluating risks, fixing and respecting quality benchmarks, tailor to the software process to a specific project, planning the activities to perform or even manage the process continuous improvement. Defining an iterative MAS development process is however not enough to realize the desired goal; indeed, it has to offer flexible tools such as CASE Tools and easy to use knowledge bases to support software professionals to efficiently deal with the methodology.

8. Conclusion

This dissertation has presented a research aimed to build, apply and validate an original iterative software development methodology for MAS called I-Tropos. Many agent-oriented software development methodologies advice to develop iteratively but they do not provide a clear template or better a project management framework to support this type of development. To this end, on the basis of a multi-criteria framework four MAS development methodologies' life cycles were evaluated. MASSIVE is the only MAS methodology offering a basic framework to support the iterative software development. I-Tropos does however cover project management in a much broader manner than MASSIVE it indeed uses the identified actors goals balanced with risks and quality factors to directly feed an iterative template and provides a global vision of the software development evolving from an iteration to the other rather than a single iteration view.

Applying the methodology on a production management system in a classical industrial environment constituted a challenge since AOSE is generally applied on small scale case studies involving a little number of collaborating agents like a travel planner, a DVD recommendation system or a search engine. The case study has, however, led to interesting results. It showed the applicability of the I-Tropos process into a real life information system development. The organizational modeling capabilities of the i* framework have been of particular interest in this, by essence, agent oriented world. Iterative enhancements allowed to better take into account user requirements, but more important, to refine our vision of the organizational setting and other business processes during the system development.

In a word, unlike other MAS development methodologies this constitutes a strong project management framework which is literally part of the development process. The case study issued from Belgian steel industry has demonstrated that the research methodology is applicable huge real life software projects and that agent-oriented modeling and development can be useful in such a context. Methodology extensions remain however to be done, it could gain a higher level of integration notably for threats and quality factors engineering solutions.

The dissertation has been organized as follows:

- Chapter 1 has motivated the advantages of MAS, the importance of iterative development and formulated the thesis;
- Chapter 2 has presented the state-of-the-art in the field of information systems software engineering;
- Chapter 3 has given an epistemological lecture of knowledge evolution in software engineering at macro and micro levels pointing to the use of iterative development as well as a Lakatosian lecture of agent-orientation;
- Chapter 4 has positioned the I-Tropos process among the agent-oriented software engineering methodologies in the context of our research;

8. Conclusion

- Chapter 5 has proposed a framework to develop MAS in the context of software applications involving users in an intensive manner. The process has been described using a semi-formal description language, the SPEM. The process strongly focuses on software project management as fundamental for adequate iterative development;
- Chapter 6 has illustrated the application of the development methodology onto a case study;
- Chapter 7 has proposed a tool for I-Tropos development support. The tool has been developed with respect to the agent-oriented concepts, it proposes tools for analysis and design diagrams edition, forward engineering capabilities as well as a software project management module.

8.2 Main Contributions

The dissertation has contributed to the field of MAS development as well as to software project management. Our research has advocated an iterative development process and the use of software project management in order to help professionals during the design of MAS. This section firstly overviews the thesis general contributions and then turns on to the contributions of the software process itself with regard to Tropos and MASSIVE.

More specifically, the main contributions of our work has included:

- An evaluation of the MAS methodologies with a strong focus on their development life cycle;
- A motivated epistemological position on software engineering pointing to the use of iterative development;
- A motivated epistemological lecture on the evolution from object to agent-orientation;
- A software development process for iterative MAS development semi-formally described with the Software Process Engineering Meta-model;
- the illustration of concepts introduced in the generic process description onto a real life case study: the production management system of a coking plant.
- A CASE-Tool, DesCARTES-Architect, for developing MAS. It assists (i) analysts when dealing with i* diagrams; (ii) designers when dealing with NFR goal diagrams, UML/AUML design diagrams, ease the use of architectural style and design patterns, (iii) generate code for automating the pro-

8. Conclusion

grammer task and (iv) project managers to track threats and quality factors, to evaluate the project effort and to manage resources allocation.

I-Tropos represents an evolution of the Tropos process. It constitutes an operationalization of the Tropos methodology in order to be used in large software developments. I-Tropos mainly fills up the gap of project management which is, for now, seldom approached in MAS literature. The main contributions are:

- a meta-level process documentation. Tropos disciplines had been described in terms of models produced without going in further details;
- a full SE life cycle coverage:
 - Early and late requirements disciplines have been rebuilt into, respectively, organizational modelling and requirements engineering;
 - An implementation, test and deployment discipline have been introduced in the process to cover the identified gaps in life cycle coverage;
 - Risk, quality, time and process management disciplines have been introduced to cover the project management aspects of the software process.
- A project management framework for the inclusion of a MAS development methodology in an iterative process template. To better evaluate this contribution, we suggest comparing it with the MASSIVE system view, the only known support framework in MAS development methodologies. I-Tropos PM framework is characterized as follows:
 - Software project development and management are model-driven since they use the the goals issued from i* SD and SR models as fundamental scope elements. This can hardly be compared with MASSIVE system view since it only reports an iteration experience to the next one and does not introduce a model-driven management approach;
 - Threats and quality factors are used for prioritizing goals' realization so that we dispose of risk and quality management at process level. The MASSIVE system view only focuses on managing its analysis and design stages in an ad hoc manner;
 - Time management provides a multi-iterations vision refined at milestones (when iteration ends) while MASSIVE only offers a single iterations management perspective.

8. Conclusion

8.3 Future Work

This section introduces the future developments that can be made on the basis of this dissertation. Future work is separated on two levels, the first level introduces the work that can be done to improve the dissertation, the second level introduces the possible evolutions of I-Tropos and of its PM framework.

I-Tropos lays the foundations of a mature software development process methodology; it covers in depth the whole traditional engineering life cycle and introduces project and process management disciplines. The process could however still be extended at different levels, among those we find:

- A higher level of integration: risk and quality management frameworks evaluate respectively threats and quality factors at process level for iterative development planning. Work has also been done for risk and quality management within the software process i.e. to deal with threats and quality issues at operational/technical level. A systematic approach would cover both aspects on the basis of a unique model or set of models. We suggest this could be achieved by integrating the framework proposed in [WAK08] into the process;
- Dealing with organizational change management. Nowadays organizations are constantly evolving: reorganisations, merges, acquisitions, etc. lead to a need to constantly manage change. I-Tropos in its actual version is better able to deal with such issues than traditional waterfall methodologies because of its iterative nature and organizational modelling capabilities. Nevertheless we suggest that to adequately manage change, one could include a dedicated discipline for a systemic approach of this topic. The interest of such a discipline is of particular interest in our MAS modelling and development method since it uses the i* framework which incorporates organizational elements as fundamentals. Such an approach would be particularly interesting in the context of ERP systems which are software systems continuously evolving with the organizational changes of the environment they are deployed in;
- As explained in this dissertation, the iterative character of a software process allows to make the organizational modelling and requirements engineering stages continuous along the whole software process to better understand the organizational setting and take into account users needs. The iterative nature of the process could, however, also be exploited to deal with the non-deterministic character of software agents. Since agents do not behave deterministically they induce a contextual change during an iteration. Since this aspect could hardly be forecasted, this could be taken into account thanks to the use of an iterative process allowing to determine a contextual change during an iteration and taking it into account during the next one;

8. Conclusion

- The generic process description needs to gain experience with its use. It should be tested on more case studies, and eventually compared to other methodologies including object-oriented ones. This would allow:
 - the refinement of the generic software process description and gain experience on the process of implementing the generic structure;
 - the calibration of the frameworks for risk, quality and time management on the basis of statistical tools applied to collected empirical data.
- The generic process description is not exhaustive. It would be interesting to extend the description by including new *WorkDefinitions*, *Activities*, *Work-Products* to:
 - Provide an explicit and systematic approach for selecting and applying patterns, Commercial-Off-the-Shelf components and other open source code reuse rather than ad hoc ones;
 - Provide an explicit and systematic approach for tailoring the software process rather than an ad hoc one;
 - Include other complementary software engineering aspects/views/models that have not been envisaged;

The PM framework could be extended independently of the I-Tropos process. Indeed, since the PM framework offers a certain degree of genericity, it could be adapted for model-driven PM in other MAS development methodologies. This could be easily done in methodologies using use-case modelling as MaSE or ADELPHI but also for traditional object-oriented UML software projects.

Finally, work remains to be done on case tool supporting the I-Tropos MAS development methodology. Some directions for future development include:

- Currently, the tool only allows generating code in SQL, Jade, JACK. Extending it with the generation of code for other agent oriented languages would add value to the automation process of MAS development.
- DesCARTES Architect still needs to be extended to fully include the proposed project management frameworks in an integrated manner.