

Towards an Integrated Platform for Adaptive Socio-Technical Systems for Smart Spaces

Giacomo Cabri*, Massimo Cossentino[†], Enrico Denti[‡], Paolo Giorgini[¶], Ambra Molesini[‡],
Monica Mordonini[§], Michele Tomaiuolo[§], Luca Sabatucci[†]

*Università di Modena e Reggio Emilia, Italy,

giacomo.cabri@unimore.it

[†]ICAR-CNR, Palermo, Italy,

{cossentino, sabatucci}@pa.icar.cnr.it

[‡]Alma Mater Studiorum - Università di Bologna, Italy,

{enrico.denti, ambra.molesini}@unibo.it

[¶]Università di Trento, Italy,

paolo.giorgini@unitn.it

[§]Università di Parma, Italy,

{michele.tomaiuolo, monica.mordonini}@unipr.it

Abstract—The widespread availability of mobile devices equipped with sensors on board is strongly pushing towards the development of *smart spaces*—offices, streets, hospital, airports, homes and shops equipped with sensing systems to help people find relevant information quickly and use services comfortably, often in a collaborative way.

A common aspect in these scenarios is that the technology complexity is amplified by the *organisational and procedural complexity* of the application domain: this is in fact typical of *socio-technical systems*—kinds of systems that inherently need to be conceived, designed and developed taking into account both the technological and the human/organisational aspects from the earliest stages.

In this paper, we select and discuss some challenges in the definition and development of adaptive socio-technical systems for smart spaces; the selected challenges aim at focusing in particular on (i) a socio-technical model and process, (ii) a multi-level integration framework, and (iii) methods and system analysis techniques for runtime adaptation.

In fact, designing such systems requires the suitable combination of enabling technologies into an operational framework, integrating and coordinating a multiplicity of processes managed by complex organisations, each made up of independent and autonomous units.

Keywords—adaptation, socio-technical systems, smart spaces.

I. INTRODUCTION

The increasing availability of broadband, powerful mobile devices equipped with sensors on board, of home appliances and other everyday-use objects with increasing processing power and interconnection capabilities is setting the base for the widespread diffusion of the so-called *smart spaces* [1], [2]. Smart spaces like offices, streets, hospital, airports, homes and shops equipped with sensing systems help people find relevant information quickly and use services comfortably, enabling also different kinds of collaboration between both devices and users. A common aspect in these scenarios is that the technology complexity is amplified by the organisational and procedural complexity of the application domain. This

is typical of the (so-called) *socio-technical systems*—kinds of systems that inherently need to be conceived, designed and developed taking into account both the technological and the human/organisational aspects from the earliest stages. It is generally acknowledged [3], [4] that designing such systems means to combine suitable enabling technologies into an operational framework, integrating and coordinating a multiplicity of processes; to make things ever more complex, such processes are usually managed by complex organisations, each made up of independent and autonomous units.

By their very nature, the above scenarios assume the presence of a wide set of heterogeneous entities, possibly differing in their execution platform, development language, support technology, enabling infrastructure, but also inner model, role and objectives. These entities interact in a coordinated, secure, responsive and adaptive way, perceiving and manipulating not only the environment in which they are immersed, but also the evolution of the environment itself.

In this paper, we highlight and discuss some challenges in the definition and development of an engineering framework for modelling and programming adaptive socio-technical systems. In particular, we have selected challenges that must be addressed in the definition of:

- a suitable *socio-technical model and process* which is able to support the software engineer throughout all the development phases, providing a global socio-technical view of the whole system, covering both the functional and non-functional aspects (with special regards to security and information privacy), as well as the social and organizational aspects;
- a *multi-level integration framework* aimed at supporting the seamless integration of existing platforms based on different development languages and paradigms;
- *methods and system analysis techniques for runtime adaptation*, needed to deal with the intrinsic dynamism of the execution environment: in fact, since requirements

in socio-technical systems evolve during execution, the reliability of adaptation is – generally speaking – a big research issue. In this paper we specifically address one portion of this problem—namely, to ensure the partial accomplishment of systems goals in case they are not all satisfiable.

The remainder of paper is organized as follows. In Section II we present the state of the art in the field, then (Section III) we focus on the main challenges concerning adaptive socio-technical systems in the specific context of smart spaces. The resulting research directions are discussed in Section IV, while future work and conclusions are drawn in Section V.

II. STATE OF THE ART

Socio-technical systems derive from integrating technology into traditional human organizations and can be seen as systems-of-systems [3], i.e. technical systems supporting interaction among social systems, situated in an environment. They typically perform tedious, physically challenging or dangerous tasks, generate and process information [4].

The development of such systems raises unique challenges, particularly as their size grows [5]: also, the borders between the social and technical aspects become more nebulous as the integration of technology into social systems becomes tighter. In this context, the increasing diffusion of powerful mobile devices equipped with on-board sensors and of home appliances with increasing processing power and interconnection capabilities is further opening the way for a wider diffusion of scenarios such as *Internet of Things* (IoT) [6] and *Pervasive Intelligence* [7]. These scenarios are becoming the new reference computational environments, with applications from wearable devices to home computing, up to disaster recovery.

Since socio-technical systems may involve different social and technical systems, the requirements are diverse, can be conflicting, and are usually specified using different forms, modeling and natural languages. For example, in a smart home, networked appliances [8], [9] and devices governed via smartphone apps [10], [11] demand that the technological and social aspects are explored and developed together. In turn, this requires on the one hand a clear vision of the value-added of their integration – i.e. which high-level services can be targeted –, on the other hand a deep understanding of the issues related to security and privacy of the information managed [12]. Designing such systems means to combine suitable enabling technologies into a unique operational framework, integrating and coordinating a multiplicity of processes managed by complex organizations, each made of autonomous units. Socio-technical systems also have to deal with an evolving environment and the participating systems also have to evolve. While failures are unavoidable in such a context, systems need to adapt to changes and keep operating.

Recent works [13], [14] highlight that a *multi-paradigm* and *multi-language* approach [15], [16] could constitute an effective base for the design and development of such systems because of its ability in representing and expressing the

specific needs of each component of the pervasive system [17] enabling an effective “merge” between the logical and declarative aspects that are mostly adopted to support intelligence, on the one side, and the functional and imperative ones that are often adopted for other aspects, on the other.

Generally, a wide set of heterogeneous systems, possibly differing in execution platform, development language, support technology, and enabling infrastructure, will have to coexist and fruitfully inter-operate in a coordinated, responsive and adaptive way, while perceiving and manipulating the environment and its evolution as needed [18].

Socio-technical systems also have to deal with dynamic operative environment. Self-adaptive systems are designed to adjust their behaviour in response to their perception of the environment and the system itself [19]. The development of such a system is complicated by the fact that it is impossible to know at design-time all possible changes in requirements and environment conditions [20]. The designer’s task is to make sure that the system satisfies its requirements as best as feasible. Therefore, operating in conditions of uncertainty, it is necessary to incorporate points of relaxation in the model that allow for automatically adjusting the levels of requirements satisfaction [21].

III. THE CHALLENGES

In complex systems such as smart spaces and critical infrastructures, technological components are *autonomous*, *loosely controllable* and *situated*: large socio-technical systems, in particular, require that components interact both with each other and with humans and organizations. *Adaptation*, *security* and *information privacy* become therefore key factors for both development and runtime execution.

The current approach for the development of socio-technical systems is mainly based on interconnecting a large number of heterogeneous components, each programmed for executing a very specific task, often with limited capabilities of runtime adaptation to the emergent situations. Although technology offers many different solutions to support the communication and the interaction at component level, it still lacks an effective support at *system* level, where the overall functionalities of the system have to be designed and then developed in terms of *coordination* of socio-technical components and global adaptation policies. Similarly, pervasive applications accumulate information from multiple parties and sources, raising critical issues as concerns people’s privacy and personal (possibly sensitive) data, which cannot be addressed by existing models and techniques [22], [23].

Given the above context, several challenges arise at different levels and from different viewpoints—namely:

- 1) the need for novel, *multi-view* modeling languages, specifically designed to cope with the heterogeneity of dimensions that are inherently present in socio-technical systems;
- 2) the definition of appropriate *design processes*, which are both well founded in terms of meta-model and flexible enough to address the wide variety of actual situations;

- 3) the provision for carefully dealing with the *privacy and security issues* raised by the coupling of managing personal data and of the size and completeness (including geo-referencing) of such data;
- 4) the need to support *runtime adaptation* to face the unpredictable changes of the environment, in particular to cope with the human aspects, which strongly condition the technical part by adding ambiguity and uncertainty;
- 5) the design and development of an *integrated platform*, putting together existing platforms typically built around one single computational paradigm, so as to get the best of all worlds;
- 6) the design and development of suitable *prototype application(s)*, both to validate the approach (methodology, language, runtime, infrastructure) and more generally to improve the state of the art of adaptive systems characterized by uncertain knowledge, mutable requirements and goals—with impact on requirements engineering, static and runtime validation, etc.

These issues are briefly discussed in the following, coupling each challenge with the corresponding outcome(s).

A. Multi-view modeling languages for socio-technical systems

Challenge: modeling socio-technical systems call for novel modeling languages, able both to cope with the complexity emerging from heterogeneity of goals, interests and preferences and to support an analysis that combines organizational and social aspects with technological functionalities and software behaviors. In particular, a multi-view approach will be necessary to support the incremental design of large socio-technical models. The views deriving more directly from the socio-technical nature of systems can be expected to be (nearly) always present, while other views might be or not be present based on the selected fragments, on the adopted notations, etc. So, harmonizing the different languages could be seen as a further challenge in the challenge.

Outcomes: a multi-view modeling language enabling the analyst to model the socio-technical system from different perspectives (views)—such as, for instance, an *organizational view*, modeling roles, goals, social and organizational dependencies; a *process view*, supporting the definition of organizational processes; a *functional and non-functional view*, where software elements represented as elements of the system can be further analysed in terms of functional and non-functional requirements (including security); an *environment view*, to model the environment in which systems are deployed and executed; and so on. Of course, the model shall be built incrementally, adding elements from each view using specific notation.

B. Process Definition

Challenge: to define a process meta-model and possibly a set of process fragments aimed at supporting the construction of a family of processes for the development of multi-paradigm, multi-platform, multi-language socio-technical systems. Such a process will use, and influence, the above

modelling language and views. Since an important requirement concerns the integration between the new system with the existing ones, an effective support of legacy devices is a necessary prerequisite. Analogously, legal, ethical and societal issues need to be taken into account throughout the process phases for driving architectural and technological choices.

Outcomes: the expected outcome consists of three elements—the *process meta-model*, a suitable set of *process fragments*, and the corresponding *design process*; all suitable to support the views introduced in the previous challenge. In particular, some fragments will be likely to support the above “fixed” views deriving from the socio-technical nature of these systems. It is worth highlighting that research on multi-paradigm modeling [24] currently does not take into account the smart spaces scenarios, nor does it consider advanced software engineering and programming paradigms such as Agent-Oriented programming. Accordingly, the definition of a suitable process meta-model for the smart spaces should represent one further advancement in the context of multi-paradigm modeling field.

C. Privacy and Security

Challenge: socio-technical system inherently manage big data from a wide variety of heterogeneous sources. Since part of such data is likely to concern personal data, its handling raises notable security and privacy issues, which call for proper methodologies and support technologies. More generally speaking, socio-technical systems by their very nature raise social and ethical implications that need to be carefully analysed and kept into account at all development stages—in particular, to comply with all the national and EU norms and regulations.

Outcomes: the main outcome should be one or more process fragments, based on the set of privacy norms and personal data regulations, to be included in the development process and to drive the platform integration according to the security and privacy requirements, ensuring that the system development is designed and structured in a law-abiding way at all stages.

D. Runtime Adaptation

Challenge: since socio-technical systems are expected to operate in unpredictable environments, in which human aspects strongly condition the technical part by adding ambiguity and uncertainty, the actual challenge is to turn these aspects into strength, by introducing adaptation—in particular, in the form of self-expression, i.e. the ability to exhibit a feasible behavior with respect to design goals and to recover failure situations to ensure the system functioning. The available requirements engineering tools and techniques [25], [26] offer only a partial support to the analysis of adaptive socio-technical systems. Indeed, recent studies [27] have highlighted the need of novel requirements engineering approaches to cope with the evolution of socio-technical systems, where requirements and system’s goals are only partially defined at design time, while they may evolve and change at runtime.

Moreover, in order to enforce the relaxation of constraints, and therefore to increase the degrees of freedom as concerns adaptation, a paradigm shift from a boolean satisfaction paradigm (a goal is either satisfied or not) towards a partial satisfaction could be introduced, so as to enable the system to reason on many alternative trade-offs for satisfying the goals and to select the most promising one with respect to other assets (such as non-functional requirements).

Outcomes: a set of techniques (process fragments and technological solutions) for the definition of semi-formal goal-oriented requirements, in which multi-level degree of satisfaction may be exploited for engineering self-adaptive system. The runtime model might also consider techniques of constraint relaxation when full satisfaction is not possible or just too costly.

E. Integrated Platform

Challenge: as concerns the development of a unified infrastructure, the main challenge is to integrate existing platforms (e.g., MUSA [28], tuProlog [29]) based on different development languages (e.g. Jason [30]) – each typically based on one single computational paradigm – so as to put together the respective strengths. A promising approach could be to define a suitable multi-level architecture, enabling at the same time the integration of different platforms and the creation of a reasoning layer, which can drive the operation of the various system components – intended as both pieces of software or humans – towards the shared goals.

A key issue will be the development of an active monitoring of the system’s behavior to reconfigure the different components when the overall goals of the system are compromised by local or global failures. Whenever a suitable reconfiguration cannot be found, the system should opt for partial goal satisfaction, where preferences and non-functional requirements will be used to select among different possible solutions.

Outcomes: the expected outcome is twofold: on the one hand, the definition of a taxonomy of existing languages and paradigms for developing adaptive socio-technical system, aimed at outlining the landscape of the existing work in the field; on the other, consequently, a new middleware aimed at providing an integrated development platform. Techniques [27], [31] based on goal-tree decomposition should explore the space of alternative solutions the system can use to guarantee an adequate level of adaptation reliability. In addition, techniques for evaluating malicious behaviors and intrusion detection should be investigated in order to prevent goals failure caused by hacking attacks and malwares [22].

IV. RESEARCH APPROACH AND DIRECTIONS

In order to cope with **the definition and development of an engineering framework for modelling and programming adaptive socio-technical systems the above demanding challenges have to be addressed. In particular, this requires** that many different aspects and technologies from different sources and at different abstraction levels are put together, top-down and bottom-up research methodologies will both be

necessary. We claim that suitably intertwining the top-down approach, where foundational solutions are devised to tackle the problem in a sound way, with the bottom-up approach, based on exploring actual application scenarios in a pragmatic manner, is likely to be the most effective to ensure both that the theoretical intuitions are grounded in practice, and to promote applicability and generality. Accordingly, the starting point could be the identification of the main weaknesses of the current engineering approaches to large scale socio-technical systems, and more generally in the context of multi-paradigm / multi-platform systems. In particular, the relevant case studies in the smart spaces context [1], [2] are to be analyzed.

From the methodological point of view, given the complexity of the development process to be defined and of the implementation infrastructure to be designed and set up, it is reasonable to expect a series of iterations, each aimed at producing one further refinement of the development process and implementation platform. We propose to define each iteration composed of the following key steps:

- 1) Definition of process requirements (multi-paradigm, multi-platform, multi-language socio-technical systems, implementation platform constraints, ...);
- 2) Definition of the process meta-model, outlining the concepts to be instantiated during the process steps and their relationships—in particular, to specify the process fragment interfaces;
- 3) Definition of the process backbone (a tentative set of process fragments with related input/output interfaces).
- 4) Risk analysis and prioritization, to prevent incompatible or partial solutions;
- 5) The study of new theories and best practices for the achievement of new challenge fragment objectives;
- 6) Solutions validation and optimisation: should different solutions to the partial problems addressed by each new challenge fragment be identified, a comparison will be necessary to identify the “best” solution – according to some common set of criteria – and prevent conflicts;
- 7) Process fragments formalization: theories and techniques will be formalized based on the existing state of the art, possibly using the Software Process Engineering Meta-model (SPEM) approach [32];
- 8) Process composition by fragment assembly;
- 9) Further development of the (multi-paradigm) integrated platform;
- 10) Validation of the process and of the platform by refining the software architecture of the application prototype.

The development of a prototype allows to apply the above approach (methodology, language, runtime, infrastructure) to the design and development of suitable *prototype application(s)* in the selected application field. The goal is both to validate the approach itself and to test the infrastructure, and more generally to improve the state of the art of adaptive systems—with impact on requirements engineering, static and runtime validation, etc.

The implementation of a first prototype application will

regard the case study of adaptive monitoring and operation of domotics and assistive technologies. Models, processes and tools will have to be adapted to adaptive monitoring and to the peculiarities of the smart spaces context: the prototype application will constantly monitor at runtime its environment and its own state, implementing one or more feedback loops to adapt its goals, soft-goals and behaviors according to the evolving operating conditions.

V. CONCLUSIONS

In this paper we have discussed the main challenges in the development of adaptive socio-technical systems in the novel context of smart spaces. Smart spaces present interesting issues that can be addressed by adaptive socio-technical systems. Besides the challenges, we have proposed a research approach we are going to carry on to address the proposed challenges.

The main aspects we are going to address in our work towards an integrated platform are: *i)* the multi-paradigm approach, *ii)* adaptation connected to the partial goal satisfaction, *iii)* the higher level of abstraction, *iv)* a more comprehensive approach that provides from model to infrastructures.

The proposed framework is specifically designed to be scalable and fit many different application scenarios in the smart spaces context: for instance, it could be used in smart cities to boost the industrial development by supporting local initiatives and projects with respect to smart mobility, smart building, energy efficiency, and so on. Again, our framework can be exploited to design and build adaptive systems that involves also the presence of humans. In particular, in the smart-home field, we envision a future in which several more or less intelligent components of different kinds will be involved in supporting our home activities. Thus, at a reduced scale, it already exhibits the most interesting challenges typical of socio-technical systems.

Privacy and security issues are another key aspect of today's and especially of tomorrow's socio-technical systems: this is why the development process needs to take these aspects into account from the earliest stages, and embed them not merely as technical issues to be dealt with, but more generally socio-technical issues implying legal, ethical, data privacy aspects—in an international context where different regulations and norms apply.

From the effectiveness and cost viewpoints, the availability of enhanced production tools, supporting multiple views at different abstraction levels, is suitable to decrease the development cost, helping to enlarge the potential market of such systems.

REFERENCES

- [1] X. Wang, J. S. Dong, C. Chin, S. R. Hettiarachchi, and D. Zhang, "Semantic space: An infrastructure for smart spaces," *Computing*, vol. 1, no. 2, pp. 67–74, 2002.
- [2] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty, "Intelligent agents meet the semantic web in smart spaces," *Internet Computing, IEEE*, vol. 8, no. 6, pp. 69–79, 2004.
- [3] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with computers*, vol. 23, no. 1, pp. 4–17, 2011.
- [4] F. Dalpiaz, P. Giorgini, and J. Mylopoulos, "Adaptive socio-technical systems: a requirements-based approach," *Requirements engineering*, vol. 18, no. 1, pp. 1–24, 2013.
- [5] B. Pollak, "Ultra-large-scale systems—the software challenge of the future," *Software Engineering Institute—Carnegie Mellon*, 2006.
- [6] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the internet of things," *Internet Computing, IEEE*, vol. 14, no. 1, pp. 44–51, 2010.
- [7] I. Khalil, *Atlantis Ambient and Pervasive Intelligence*. Springer, 2015.
- [8] LG. (2015) French 3-door refrigerator. [Online]. Available: <http://www.lg.com/us/refrigerators/lg-LFX31995ST-french-3-door-refrigerator>
- [9] Samsung, "Samsung: Intelligent refrigerator." [Online]. Available: <http://tinyurl.com/q6oaysv>
- [10] Google. (2014) Works with nest. [Online]. Available: <http://techcrunch.com/2014/06/23/google-makes-its-nest-at-the-center-of-the-smart-home/>
- [11] Apple. (2014) Home kit. [Online]. Available: <https://developer.apple.com/homekit/>
- [12] C. Bodei, P. Degano, L. Galletta, G. Mezzetti, and G. Ferrari, "Security in pervasive applications: a survey," *European Journal of Law and Technology*, vol. 4, no. 2, 2013.
- [13] E. A. Belloni, "A multi-paradigm approach for mobile agents development," *Journal of Computer Science & Technology*, vol. 1, 2001.
- [14] P. Van Roy, P. Brand, D. Duchier, S. Haridi, C. Schulte, and M. Henz, "Logic programming in the context of multiparadigm programming: the oz experience," *Theory and practice of logic programming*, vol. 3, no. 06, pp. 717–763, 2003.
- [15] J. Ponge, F. Le Mouél, and N. Stouls, "Golo, a dynamic, light and efficient language for post-invokedynamic jvm," in *Proceedings of the 2013 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools*. ACM, 2013, pp. 153–158.
- [16] M. Odersky and T. Rompf, "Unifying functional and object-oriented programming with scala," *Communications of the ACM*, vol. 57, no. 4, pp. 76–86, 2014.
- [17] D. Wampler and T. Clark, "Guest editors' introduction: multiparadigm programming," *IEEE Software*, vol. 27, no. 5, p. 20, 2010.
- [18] A. Omicini and S. Mariani, "Agents & multiagent systems: En route towards complex intelligent systems," *Intelligenza Artificiale*, vol. 7, no. 2, pp. 153–164, 2013.
- [19] B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, G. Di Marzo Seruendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. M. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. A. Müller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns, and J. Whittle, "Software Engineering for Self-Adaptive Systems: A Research Roadmap," in *Reasoning Web. Semantic Technologies for Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–26.
- [20] I. J. Jureta, A. Borgida, N. A. Ernst, and J. Mylopoulos, "The Requirements Problem for Adaptive Systems," *ACM Transactions on Management Information Systems (TMIS)*, vol. 5, no. 3, pp. 17–33, Jan. 2015.
- [21] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J.-M. Bruel, "RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems," *2009 17th IEEE International Requirements Engineering Conference (RE)*, pp. 79–88, 2009.
- [22] P. Samarati, M. Tunstall, J. Posegga, K. Markantonakis, and D. Sauveron, *Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices: 4th IFIP WG 11.2 International Workshop, WISTP 2010, Passau, Germany, April 12-14, 2010, Proceedings*. Springer, 2010, vol. 6033.
- [23] S. Geetha, G. Zayaraz, J. Madhusudanan, and V. P. Venkatesan, "Novel security model for pervasive systems," *International Journal of Computer Applications*, vol. 63, no. 2, 2013.
- [24] P. J. Mosterman and H. Vangheluwe, "Guest editorial: Special issue on computer automated multi-paradigm modeling," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 12, no. 4, pp. 249–255, 2002.
- [25] V. E. S. Souza, A. Lapouchnian, K. Angelopoulos, and J. Mylopoulos, "Requirements-driven software evolution," *Computer Science-Research and Development*, vol. 28, no. 4, pp. 311–329, 2013.

- [26] W. N. Robinson, "A requirements monitoring framework for enterprise systems," *Requirements engineering*, vol. 11, no. 1, pp. 17–41, 2006.
- [27] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Formal reasoning techniques for goal models," in *Journal on Data Semantics I*. Springer, 2003, pp. 1–20.
- [28] MUSA. (2015) Middleware for user-driven service adaptation. [Online]. Available: <http://aose.pa.icar.cnr.it/MUSA>
- [29] (2015) tuprolog. [Online]. Available: <http://apice.unibo.it/xwiki/bin/view/Tuprolog/>
- [30] (2015) Jason language home. [Online]. Available: <http://jason.sourceforge.net/wp/>
- [31] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Reasoning with goal models." Springer, 2002, pp. 167–181.
- [32] OMG, "Software process engineering metamodel. version 2.0." <http://www.omg.org/spec/SPEM/2.0/PDF>, April 2008.