Practical Applications of *i** in Industry:

The State of the Art

Eric Yu
University of Toronto
Toronto, Canada
eric.yu@utoronto.ca

Daniel Amyot, Gunter Mussbacher
University of Ottawa
Ottawa, Canada {damyot, gunterm}@eecs.uottawa.ca

Xavier Franch
Universitat Politècnica de Catalunya
Barcelona, Spain
franch@essi.upc.edu

Jaelson Castro
Universidade Federal de Pernambuco
Recife, Brazil
jbc@cin.ufpe.br

***Abstract*—i* is a goal-oriented and agent-oriented modeling framework that focuses on the analysis of intentional and strategic relationships among actors. In this mini-tutorial, we highlight a number of recent applications in practical industrial and business settings.**

***Index Terms*—iStar, social modeling, goal-oriented requirements engineering, GRL.**

I. i*, GRL, VARIANTS, AND EXTENSIONS

i* is an agent- and goal-oriented modeling framework that focuses on the analysis of intentional relationships among social actors [1]. Actors depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. Actors are strategic in that they explore and choose among alternative patterns of dependencies to further their interests, seeking out opportunities and avoiding vulnerabilities. By highlighting the motivations, intents, and rationales behind actual or potential sociotechnical configurations, the kind of social modeling and analysis enabled by i* complements conventional requirements models that focus on behaviors and interactions between systems and their environments. As information and software systems become ever more networked and deeply embedded socially, insights on strategic relationships are increasingly pertinent.

i* modeling has been applied in many contexts, including requirements analysis (especially in "early-RE"), business process redesign, business model innovation, information security, privacy and trust, digital asset protection, regulatory policy compliance, enterprise architecture, software development methodologies, software architecture, software processes (e.g., agile methods), software ecosystems, and more. Many of these application directions have spawned extended versions of i* (along with specialized tools and methodologies), such as Tropos for multi-agent system development, and several variants of Secure i*.

A standardization effort has led to the Goal-oriented Requirements Language (GRL), which together with Use Case Map (UCM), constitute the User Requirements Notation (URN), an ITU-T international standard since 2008 (updated 2012) [2,3]. Nevertheless, despite much experimentation within the research community and international standardization, many open research questions remain. In this mini-tutorial, we offer a sampling of recent applications of i* in practical industrial and business settings. More information can be found on the iStarWiki.org website.

II. BUSINESS PROCESS MANAGEMENT & COMPLIANCE

GRL has been applied to business process management and legal compliance in different sectors (healthcare, security, and finance) [4,5]. Building on the core concepts of i*, GRL provides additional features such as quantitative evaluation scales (in addition to qualitative ones), evaluation strategies (initial satisfaction levels for some of the intentional elements) with varied propagation algorithms, indicators (comparing real values against targets, often used as an interface with data sources outside the goal model), profiling mechanisms (to tailor GRL to particular application domains), and integration with UCM scenarios within URN. While UCM offers a scenario-oriented business process modeling notation, it is much enhanced when used in conjunction with GRL, which brings a unique goal-oriented view (answering "why" aspects) missing in many business process modeling environments. This is further augmented by the recent addition of GRL indicators to the URN standard, which enable measuring the performance of processes and organizations against quantifiable objectives. URN has been used for many years for modeling business processes and their objectives, including clinical processes in healthcare. In addition, GRL can be used to model legal aspects with which business processes must comply. Much work has been done in the past seven years towards evolving GRL and the jUCMNav tool to better support compliance management (including assessment, measurement, maintenance, and improvement) in domains as varied as healthcare policies, aviation security regulations, and financial regulations.

III. CONTEXT-AWARE & ADAPTIVE SYSTEMS

In recent work, GRL is being used to define system context in a way that enables measuring this context (via indicators) and using this information to compute goal satisfaction and influence the system behavior (described with UCM) dynamically, and even interactively. The combined use of GRL and UCM in this context enables one to describe adaptive systems, whose behavior can change at run-time by taking into account external contextual information.

IV. Open Source Software Ecosystems

The RISCOSS project (www.riscoss.eu) aims to produce methods and tools to support the evaluation of risks, costs, and opportunities for organizations about to integrate OSS components into their software products [6]. Project partners (corporations, SMEs, communities, observatories, etc.) recognize the importance of a strategic perspective on the OSS project ecosystem for successful adoption. Building upon i*, the project is developing: (a) an ontology for OSS-related concepts; (b) organizational patterns around the relevant identified dimensions (e.g., business strategy: from full OSS collaboration to OSS exploitation); (c) different views of models for different purposes (communication, specification, risk analysis, etc.); (d) specialized modeling concepts for the RISCOSS domain; and (e) adaptation of classical risk management techniques to the i* framework setting.

V. Architecting hybrid systems

Most current software systems are hybrid, integrating heterogeneous software components from diverse sources. In architecting such systems, many challenges emerge, including: identification of the strategic needs for the system; identification of the specific services to be offered; and the grouping of services into atomic domains. The DHARMA method, based on i*, was designed to drive the architecting process of hybrid systems [7]. DHARMA was applied in two large-scale projects: the renovation of the IS inside the company ETAPATELECOM, and the elaboration of an IT strategic plan for the Cuenca Airport. These two projects provided useful insights on the practical use of i* when non-technical stakeholders are involved in the production of large-scale models. Key factors that contributed to project success included simplified models, adequate training, and use of familiar tools (by tailoring MS Visio instead of adopting an i*-specific tool).

VI. From i* to architectural models

Goal orientation is an increasingly recognized paradigm for eliciting, modeling, specifying, and analyzing software requirements. However, methods for relating goal models to architectural models are still lacking. The STREAM approach (Strategy for Transition between REquirements models and Architectural Models) uses model transformations to derive architectural structural specifications from system goals [8]. The source and target languages are respectively the i* modeling language and the Acme architectural description language. STREAM-A is an extension that supports the design and evolution of systems that require adaptability [9].

VII. Software Product Lines

Goal Oriented Requirements Engineering (GORE) approaches can effectively capture stakeholder objectives as well as system requirements. In the context of Software Product Lines (SPL), they offer a natural way to capture commonalities and variabilities in a product family. Goals to Software Product Lines (G2SPL) [10] is an approach that systematically guides the creation of an SPL feature model from i* models with cardinality. Further, in order to capture system behavior (not possible using a strictly GORE approach), G2SPL is integrated with a technique to specify use case scenarios with variability. This new approach, named GS2SPL (Goals and Scenarios to Software Product Lines) includes a sub-process for configuring specific applications of a SPL based on the priority given to non-functional requirements [11].

Acknowledgment

This work presents some results from the RISCOSS project, funded by the EC 7th Framework Programme FP7/2007-2013 under the agreement number 318249, and from projects supported by CNPq and CAPES.

References

[1] E. Yu, P. Giorgini, N. Maiden, J. Mylopoulos (eds.) Social Modeling for Requirements Engineering. Cambridge, MA: MIT Press. 2011.

[2] International Telecommunication Union, Recommendation Z.151 (10/12), User Requirements Notation (URN) – Language definition, Oct. 2012. http://www.itu.int/rec/T-REC-Z.151/en

[3] D. Amyot and G. Mussbacher, "User Requirements Notation: The First Ten Years, The Next Ten Years", Journal of Software (JSW), 6(5), pp. 747–768, 2011.

[4] A. Pourshahid, P. Chen, D. Amyot, A.J. Forster, S. Ghanavati, L. Peyton, and M. Weiss, "Business Process Management with the User Requirements Notation", Electronic Commerce Research, 9(4), 2009, pp. 269–316.

[5] R. Tawhid, E. Braun, N. Cartwright, M. Alhaj, G. Mussbacher, A. Shamsaei, D. Amyot, S.A. Behnam, and G. Richards, "Towards Outcome-Based Regulatory Compliance in Aviation Security", IEEE Int. Reqmts. Eng. Conf., 2012, pp. 267–272.

[6] X. Franch et al., "Managing Risk in Open Source Software Adoption", Proc. 8th Int. Conf. on Software Engineering and Applications (ICSOFT-EA 2013). SciTePress, 2013.

[7] J.P. Carvallo and X. Franch, "On the use of i* for Architecting Hybrid Systems: A Method and an Evaluation Report", Proc. 2nd IFIP Working Conf. on the Practice of Enterprise Modeling (PoEM), LNBIP, vol. 39, Springer, 2009.

[8] J. Castro, M. Lucena, C. Silva, F. Alencar, E. Santos, and J. Pimentel, "Changing attitudes towards the generation of architectural models", Journal of Systems and Software, vol. 85(3), March 2012, pp. 463–479, 2012.

[9] J. Pimentel, M. Lucena, J. Castro, C. Silva, E. Santos, and F. Alencar, "Deriving software architectural models from requirements models for adaptive systems: the STREAM-A approach", Requirements Engineering, 17(4), pp. 259–281, Springer, 2012.

[10] C. Silva, C. Borba, and J. Castro, "A goal-oriented approach to identify and configure feature models for software product lines", Proc. of the 14th Workshop on Requirements Engineering (WER'11), pp. 395–406, 2011.

[11] G. Guedes, C. Silva, J. Castro, M. Soares, D. Demerval, and C. Souza, "GS2SPL: Goals and Scenarios to Software Product Lines", 24th Int. Conf. on Software Engineering & Knowledge Engineering (SEKE), pp. 651–656, July 2012.