Using AI to Model Quality Attribute Tradeoffs

Neil A. Ernst, Ian Gorton
Software Engineering Institute–Carnegie Mellon University
Pittsburgh, USA
nernst,igorton@sei.cmu.edu

***Abstract*—Many AI techniques have been applied to goal-oriented requirements engineering. However, such techniques have focused mostly on the intellectual challenge and ignored the engineering challenge of RE at scale. We discuss some of these existing approaches. We then introduce some early work that aims to add contextual quality attribute information to leverage the power of AI techniques and tools with real-world engineering. We believe this will address some of the acquisition and context problems that have plagued AI in RE.**

***Index Terms*—requirements, analysis, agile**

I. Using AI To Solve The Requirements Problem

This paper describes our use of AI in modeling requirements, and introduces an approach to understanding system context using AI techniques. We have immediate experience with applying AI to modeling and analyzing requirements. This seems natural, since both AI and RE are concerned with modeling the world. In particular, in our work (e.g., [1]) we have applied satisfiability solving (SAT), abductive inference for diagnosis [2] and local search [3], among others, to the problem of representing and reasoning over the requirements problem. We formulate the requirements problem as satisfying the condition $W, S \vdash R$, where W is domain assumptions, S is a specification, and $W \cup S$ is consistent and $\vdash$ is some logical deduction relationship. In Techne [4] we represent requirements R as consisting of goals G and softgoals/quality attributes Q. Quality attributes are desired values of non-binary measurable properties of a system-to-be.

This representation positions the populated requirements knowledge base Δ as a container for well-formed formulas and assertions that can be used to answer questions for both the problem and solution domain. This is very similar to knowledge modeling for AI proposed by Levesque [5], among others. It also builds on RML [6] and Telos [7]. AI work in theories of abductive reasoning for diagnosis [2] and planning [8] is incredibly useful for finding satisfiable models for system design. In Menzies [9] these concepts were applied to solving requirements optimization challenges. The common concern here is the treatment of a requirements problem as a knowledge base, much like any other expert system might, with suitable operators for extracting knowledge using ASK style questions. For example, in Ernst et al. [1] we introduced an operator called MINIMAL_GOAL_ACHIEVEMENT to find the minimal set of goals to satisfy high-level requirements.

We are currently applying these AI-derived goal-modeling approaches to understanding how quality attributes (such as security, performance, maintainability) guide possible software architectures and designs. In particular, we are interested in properly sizing and allocating quality attribute specific work to incremental iterations. However, there are three challenges we see in using AI to enhance RE acquisition, modeling and analysis:

1) Overcoming scalability problems of standard AI approaches.
2) Adapting AI knowledge representation techniques to RE problems.
3) Reducing acquisition effort. This is the KA bottleneck and context problem.

Scalability. In a world where Google routinely applies AI-derived algorithms to millions of instances, these techniques—most of which are NP-complete or worse—cannot scale. This is acceptable if the goal model is sufficiently abstract (i.e., there are few elements). Moreover, just as expert systems ran into the knowledge acquisition bottleneck, what is missing (and difficult to acquire) is contextual information (not the meta-level context, but the specifics).

RE Knowledge Traditional AI techniques were applied to knowledge about the world as-is, while requirements engineering is concerned with a system to-be. The distinction is important—see the seminal work of Zowghi and Offen [10]. Consider belief revision: one axiom is typically that new information should be preferred to older information. This is reasonable if we are dealing with self-driving cars; the problem with these principles is that in RE our objective is to solve the (evolving) requirements problem. Thus new information (such as new COTS capabilities) may in fact be rejected, if it does not improve the solution.

Acquisition we still lack guidance on how to get RE models populated to begin with. In other words, while we have focused a lot of energy on modeling and reasoning, we have paid little effort to eliciting requirements for those models. For Ernst's thesis he spent 30+ hours entering the requirements and constraints from the Payment Card Industry Data Security Standard (PCI-DSS) into a goal model. There are some potential approaches to overcoming this; for example, Brill and Knauss [11] have suggested eliciting requirements using crowd-sourcing. There are many repositories of requirements in industrial settings, for example as DOORS databases, but turning a textual description into something useful for AI techniques is difficult. Cleland-Huang's traceability work [12] is a nice example here.

978-1-4799-6355-3/14/$31.00 © 2014 IEEE

AIRE 2014, Karlskrona, Sweden

II. Populating QA Templates

To address the acquisition challenge and the RE knowledge challenge, we have been working on using semantic wikis[1] to leverage quality attributes in design. To date these have scaled due to size limits (for example, there are around 50 tactics in our wiki). Much like the NFR framework [13], the wiki we have created is a set of common tactics [14], [15] for design solutions to important quality attributes. These tactics start with a quality attribute requirement, and are generic and not domain-specific, so they are broadly applicable. For example, Performance is decomposed into Control Resource Demand, which in turn is decomposed into manage sampling rate, prioritize events, etc..

Since the tactics are generic, each new project must specialize/instantiate them. We do this by conducting interviews or surveys with architects and designers. Each tactic group and tactic has an associated set of questions. We walk through each tactic, drilling into the specific sub-tactics. The architect is asked how and where his system will fulfill that tactic. As well, we capture business criticality, implementation difficulty and location (in the code or design). This provides us with a knowledge base Δ, linking the high-level quality attribute down to design/implementation-level decisions. We use semantic annotations to create cross-connections between related tactics (for example, security and performance tradeoffs). How does this overcome the knowledge acquisition bottleneck? Using a pre-populated and non-contextual ontology that reflects best practices (e.g., design patterns, architecture tactics, and quality models) provides a set of common questions that each domain might have. If Performance is a concern (and it always is), then the tactic Control Resource Demand is an option. Now the architect must explain how/where that tactic is relevant. It may not be; this is the 'new' information that is rejected, from our belief revision discussion above. While interviews are burdensome, the process is quick, since we assume that the chief architect or program manager knows the answers reasonably well. The intent is to complete the acquisition exercise within a day. The expert is navigating a pre-existing design space and elaborating it with context, rather than completely documenting the solution.

Since this is captured in a semantic wiki, the data are captured as machine-readable triples in the Resource Description Framework (RDF) and entered into Δ. We are now beginning to explore how these facts can be used to reason about goal satisfaction, quality attribute tradeoffs, and iteration planning, using our existing work in goal evaluation with AI techniques like SAT solving and abductive reasoning. Compared to approaches such as the NFR framework or COTS-selection, while the concept is similar (instantiate a template, and let the template reduce the amount of uncertainty and re-discovery you need), what is innovative is reaching deep into implementation decisions (typically of more interest to industry). Mixing problem and implementation in RE models has historically been seen as a bad idea, but with agile approaches this artificial split hinders an evolutionary development process. Instead, a software project jumps between these peaks as appropriate. In our view, the tactics link implementations (frameworks, design patterns) to quality attributes. The wiki approach has the flexibility to capture the requirements and design knowledge simply, while retaining the underlying formal representation that permits automated reasoning about tradeoffs and scheduling.

[1] https://semantic-mediawiki.org

Acknowledgment

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This material has been approved for public release and unlimited distribution. DM-0001388

References

[1] N. A. Ernst, A. Borgida, and I. J. Jureta, "Finding incremental solutions for evolving requirements," in *International Conference on Requirements Engineering*. Trento, Italy: IEEE, 2011, pp. 15–24.

[2] R. Reiter, "A Theory of Diagnosis from First Principles," *Artificial Intelligence*, vol. 32, pp. 57–95, 1987.

[3] F. Glover, "Tabu Search–Part I," *INFORMS Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[4] I. J. Jureta, A. Borgida, N. Ernst, and J. Mylopoulos, "Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling," in *International Conference on Requirements Engineering*, Sydney, Australia, 2010, pp. 115–124.

[5] H. J. Levesque, "Foundations of a Functional Approach to Knowledge Representation," *Artificial Intelligence*, vol. 23, no. 2, pp. 155–212, 1984.

[6] S. Greenspan, J. Mylopoulos, and A. Borgida, "On formal requirements modeling languages: RML revisited," in *International Conference on Software Engineering*, Sorrento, Italy, 1994, pp. 135–147.

[7] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis, "Telos: Representing Knowledge About Information Systems," *TOIS*, vol. 8, pp. 325–362, 1990.

[8] C. Boutilier, T. Dean, and S. Hanks, "Planning under uncertainty: structural assumptions and computational leverage," in *New Directions in AI Planning*. IOS Press, 1996, pp. 157–171.

[9] T. Menzies, S. M. Easterbrook, B. A. Nuseibeh, and S. Waugh, "An empirical investigation of multiple viewpoint reasoning in requirements engineering," in *International Conference on Requirements Engineering*, Limerick, Ireland, 1999, pp. 100–109.

[10] D. Zowghi and R. Offen, "A Logical Framework for Modeling and Reasoning about the Evolution of Requirements ," in *International Conference on Requirements Engineering*, 1997, pp. 247–257.

[11] O. Brill and E. Knauss, "Structured and unobtrusive observation of anonymous users and their context for requirements elicitation," in *International Conference on Requirements Engineering*, Trento, Italy, 2011, pp. 175–184.

[12] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "The Detection and Classification of Non-Functional Requirements with Application to Early Aspects," in *International Conference on Requirements Engineering*. Minneapolis, Minnesota: IEEE, 2006, pp. 39–48.

[13] L. Chung, J. Mylopoulos, and B. A. Nixon, "Representing and Using Nonfunctional Requirements: A Process-Oriented Approach," *IEEE Transactions on Software Engineering*, vol. 18, pp. 483–497, 1992.

[14] M. Mirakhorli, Y. Shin, J. Cleland-Huang, and M. Cinar, "A tactic-centric approach for automating traceability of quality concerns," in *International Conference on Software Engineering*. Zurich: IEEE, Jun. 2012, pp. 639–649.

[15] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice* , 3rd ed., ser. SEI Series in Software Engineerig. Addison-Wesley Professional, 2012.