

Run-time Resolution of Uncertainty

Kristopher Welsh, Pete Sawyer
School of Computing & Communications
Lancaster University, Lancaster, UK
k.welsh@lancaster.ac.uk, sawyer@comp.lancs.ac.uk

Nelly Bencomo
INRIA Paris - Rocquencourt
78153 Le Chesnay, France
nelly@acm.org

Abstract—Requirements awareness should help optimize requirements satisfaction when factors that were uncertain at design time are resolved at runtime. We use the notion of claims to model assumptions that cannot be verified with confidence at design time. By monitoring claims at runtime, their veracity can be tested. If falsified, the effect of claim negation can be propagated to the system’s goal model and an alternative means of goal realization selected automatically, allowing the dynamic adaptation of the system to the prevailing environmental context.

Keywords-self adaptive systems; requirements models; goals;

I. INTRODUCTION

In [1] we argued that requirements models for self-adaptive systems should be runtime entities that can be reasoned over in order to understand the extent to which they are being satisfied and to support adaptation decisions. The primary contribution of the work described here is to demonstrate the feasibility of maintaining requirements models at runtime and their utility for guiding principled adaptations to contexts unanticipated at design time.

We present the tool of the REAssuRE method (REcording of Assumptions in Requirements Engineering). With REAssuRE, the representation to model domain assumptions, allows data gathered by monitoring to be used to drive system self-adaptation. The representation used is an extension of i* Strategic Rationale (SR) models [2] using *claims*, borrowed from the NFR framework [3]. Claims record the rationale for selection of one of the goal realization strategies defined at design time.

II. REASSURE

Claims serve as markers for uncertainty, helping the analyst at design time to evaluate the consequences of assumptions proving false. At run-time, REAssuRE maintains representations of goal and claim refinement models to drive run-time adaptation. These run-time models permit dynamic reasoning over how data collected by monitoring affects the truth of the claims.

Claims record the rationale for a choice of goal realization strategy when the selection is based on imperfect information. The rationale claims represent is orthogonal to the rationale that is implicit in the balance of contribution link values for all the possible realization strategies for a

given goal, and may even be used to record why the implicit rationale is not the one used to make the decision.

As an example of our use of claims, consider the fragment of a simple SR model of a robot vacuum cleaner for domestic apartments depicted in Fig 1. The vacuum cleaner has a goal to clean the apartment (clean apartment) and two softgoals; to avoid causing a danger to people within the house (avoid tripping hazard) and to be economical to run (minimize energy costs). The vacuum cleaner goal clean apartment can be satisfied by two different realization strategies. It can clean at night or when the apartment is empty. These two strategies are represented by two alternative tasks related to the goal using means-end relationships.

The choice of best strategy appears unclear because the balance of -ve and +ve effects on satisfaction of the softgoals (the implicit rationale) favours selection of neither the clean when empty nor the clean at night task. However, because the analyst wishes to prioritize the minimize energy costs over the avoid tripping hazard softgoal, the analyst uses a claim, no tripping hazard, to break the hurt contribution link from the clean at night task to the avoid tripping hazard softgoal. The break-ing claim nullifies the contribution link to which it is attached. In this case it nullifies the negative impact that night cleaning was presumed to have on tripping hazard avoidance, so night cleaning effectively has a neutral impact. However, the no tripping hazard claim is an assertion in which the analyst has some doubt; it is based on an assumption that at night, the apartment residents will be asleep and therefore in no danger of tripping over the vacuum cleaner. The analyst hopes that this is so in order to exploit night-time cleaning to promote energy saving. Because the analyst recognizes that the assumption may be false, a claim is used in preference to changing the clean at night- avoid tripping hazard contribution link to neutral. The claim thus acts as a marker for uncertainty.

Nullifying the negative effect of night cleaning on tripping hazard avoidance has the effect of promoting the night cleaning strategy over the empty apartment cleaning strategy. A break-ing a claim always results in a neutral value for the contribution link to which the claim is attached. The inverse of a break-ing claim is a make-ing claim, which, consistently with the NFR framework, boosts the value of the contribution link to which it is attached. Thus a hurt link

is transformed into a contribution that completely prevents satisfaction of the softgoal; a *break* link. A *help* link is transformed into a contribution that completely satisfies the softgoal; a *make* link.

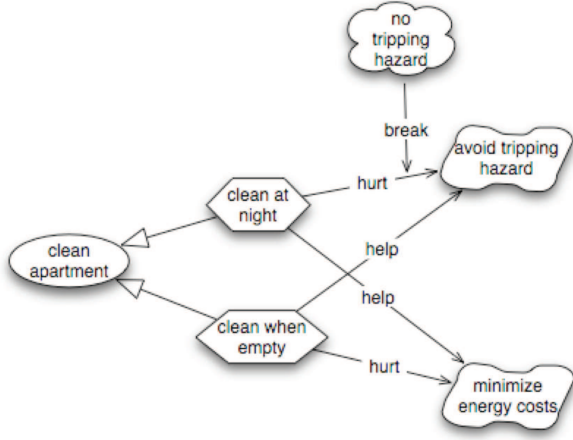


Figure 1. A robot vacuum cleaner

A claim may be derived from another claim. Claim derivation can be represented explicitly in a claim refinement model. Fig 2 is a claim refinement model that shows that the *no tripping hazard* claim is derived from three other claims: *family sleeps at night*, *vacuum is easy to see* and *vacuum has warning light*, arranged in a hierarchy. Falsity of any claim will propagate down the claim refinement model to the bottom-level claim.

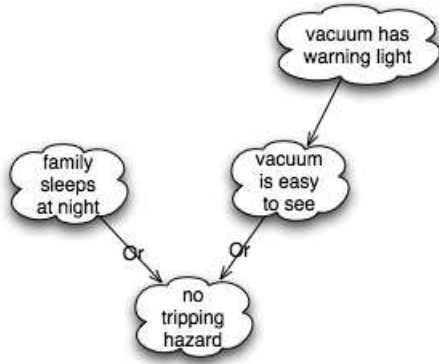


Figure 2. Claim refinement model for no tripping hazard claim

Claim refinement models and the relevant parts of i^* SR models can be represented efficiently in memory. The impact of the goal realization strategies on softgoals can be re-evaluated as evidence for and against the veracity of assumptions is acquired. The decision on whether to adapt follows the following reasoning model:

Let the function *satisfices* represent the contribution value from a task, softgoal pair:

$$\text{satisfices: } T \times SG \rightarrow C$$

where SG is the set softgoals, T is the set of tasks and C is the set of possible contribution values $\{break, hurt, neutral,$

$make, break\}$.

i is an index in the set of tasks that represent alternative realizations of goal g and t_{ig} is thus one of these tasks.

The task selected as the realization strategy for goal g thus follows a simple selection criterion. The strategy with the net greatest value of contribution link values for all of the softgoals it influences is the one selected.

$$\max_i \sum_{sg \in SG} \text{satisfice}(t_{ig}, sg) \quad (1)$$

An automatic reasoning system can be provided to evaluate softgoal satisficement at runtime. Provided the claims can be monitored by real-time data collection and criteria defined that allow the data to be interpreted computationally, and if the system has a self-adaptive capability, then the reasoning engine can trigger adaptation as a direct response to events in the system's environment.

Furthermore, the tool allows us to generate either directly adaptation policies or Genie models [4]. Genie generates adaptation policies and other middleware related artefact from models including components, component configurations and reconfiguration scripts. With these approaches and tools working in tandem we provide a tool-supported model-driven chain approach. We have applied REAssURE to GridStix [4], an experimental flood warning system that was deployed on the River Ribble in England.

III. CONCLUSION

This paper has presented a technique and its implementation for making explicit where uncertainty underpins analyst decisions in goal models of self-adaptive systems. Our aim was to investigate the feasibility of run-time requirements models as a means to achieve requirements-aware systems. REAssURE is able to reason about how design-time assumptions affect goal realization strategies. Evidence for or against design-time assumptions is gathered by claim monitoring. Adaptation between alternative goal realizations is determined by the balance of softgoal trade-offs.

REFERENCES

- [1] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein, "Requirements-aware systems: A research agenda for re for self-adaptive systems," *RE'10*, pp. 95–103, 2010.
- [2] E. S. K. Yu, "Towards modeling and reasoning support for early-phase requirements engineering," in *RE 97, USA*, 1997.
- [3] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Springer, 1999, vol. 5.
- [4] N. Bencomo, P. Grace, C. Flores, D. Hughes, and G. Blair, "Genie: Supporting the model driven development of reflective, component-based adaptive systems," in *ICSE'08 Formal Demos Track*, 2008.