Tracing Requirements for Adaptive Systems using Claims

Nelly Bencomo

Tracing Requirements for Adaptive Systems using Claims *

Kristopher Welsh
School of Comp. & Comm.
Lancaster University, UK
k.welsh@lancaster.ac.uk

Nelly Bencomo
INRIA Paris - Rocquencourt
France
nelly@acm.org

Pete Sawyer
School of Comp. & Comm.
Lancaster University, UK
sawyer@comp.lancs.ac.uk

ABSTRACT

The complexity of environments faced by dynamically adaptive systems (DAS) means that the RE process will often be iterative with analysts revisiting the system specifications based on new environmental understanding product of experiences with experimental deployments, or even after final deployments. An ability to trace backwards to an identified environmental assumption, and to trace forwards to find the areas of a DAS's specification that are affected by changes in environmental understanding aids in supporting this necessarily iterative RE process. This paper demonstrates how claims can be used as markers for areas of uncertainty in a DAS specification. The paper demonstrates backward tracing using claims to identify faulty environmental understanding, and forward tracing to allow generation of new behaviour in the form of policy adaptations and models for transitioning the running system.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications—*design, software architectures*

General Terms

Design, Documentation

Keywords

Claims, requirements, traceability, adaptive systems

1. INTRODUCTION

A Dynamically Adaptive System (DAS) monitors its environment at runtime, and adjusts its behaviour according to monitored changes. Specifying a system that is to operate in a complex, changeable environment is a challenging task, involving analysis of multiple sets of potentially disparate environmental conditions. The likelihood of Requirements Engineers fully understanding the operating environment and specifying a DAS that will successfully identify, and have the flexibility to operate in, all encountered environmental conditions at the first attempt is therefore remote.

The ability of reusable adaptive infrastructures such as GridKit [4] to utilise adaptive behaviour specifications in the form of adaptation policies means that the task of adjusting a DAS's adaptive behaviour can be less complex than the task of reprogramming a system, where such adaptive behaviour is hard-coded. However, any RE process that takes advantage of this flexibility to allow multiple iterations of proposed adaptive behaviour to be explored (e.g. through prototyping), needs to allow practitioners to "go back to the beginning" and refine their domain understanding as knowledge improves, before reconsidering the DAS's adaptive behaviour. In other words, some requirements tracing activity is promoted as essential standard practice when performing RE for a DAS. We promote the use of recorded assumptions made during the requirements modelling process to provide a means of documenting and tracing the decisions made during the modelling process. By recording assumptions on DAS models, it becomes possible to revisit them in light of changed information or new understanding as necessary. This revisiting of previously made decisions can be done by humans later in the RE process, or potentially by the DAS itself at runtime. In this paper we focus on the work done by humans (off-line).

This paper demonstrates that the ability to trace backwards to previously captured areas of uncertainty, along with knowledge of a DAS's proposed adaptive behaviour, allows for a relatively straightforward identification of the configurations (defined in the adaptive behaviour specification) affected by a change in high-level domain understanding. This paper also demonstrates that the same captured uncertainty information, along with knowledge of the DAS's proposed adaptive behaviour, can be used to trace forwards after a change, allowing a modified specification for a DAS's adaptive behaviour to be generated from modified models to achieve requirements model-driven development.

The paper is organised as follows: Section 2 discusses related work. Section 3 shows how claims can be used to mark areas of uncertainty for future tracing. Section 4 and 5 demonstrate the backwards forward tracing. Section 6 concludes the paper.

*Partially supported by Marie-Curie Requirements@run.time, EU DiVA, and EU CONNECT projects.

TEFSE '11, May 23, 2011, Waikiki, Honolulu, HI, USA

2. RELATED WORK

The study of requirements for DASs shares their initial roots with the research field Requirements Monitoring, which proposed the run-time collection of data to assess a system's conformance to its requirements [9]. The possibility of taking action when monitoring data indicates sub-optimal performance is a natural extension of this work. Several authors [10, 13] report on the use of goals for modelling requirements for DASs, using i* [13] and KAOS [8]. The strength of goal modelling when dealing with DAS requirements lies in the inherent ability to reason about alternate goal attainment strategies, and partial goal fulfilment. The RE community is beginning to consider how systems can utilise these models at runtime to guide their adaptive behaviour [5].

We have previously proposed the use of claims to enhance the traceability of DAS requirements models [12], demonstrating that a relatively simple record of an assumption made can be used to support requirements evolution. In this paper, we extend our previous work to demonstrate forward tracing using the same captured assumptions, modelled as claims, to derive modified adaptive behaviour as models evolve.

There is some existing work aimed at traceability in model-driven development. Specifically [2] and [7] address techniques to account for how requirements relate to the various artefacts created during the design phase. However, none of these works investigate traceability of requirements to support changes in requirements specifications. These changes may arise because of actual changes to the requirements of the application, or as a result of improved understanding of the environment in an iterative software development process. Uncertainty plays an important role in any software-based system that needs to evolve and adapt continuously to meet the goals [11]. In [3], authors discuss traceability in the presence of uncertainty. Similar to our work, authors propose to attach supplementary information to traceability links. This additional information describes the confidence and the rationale for its creation. The authors take into account the fact that the rationale that supports design decisions is often based on assumptions and beliefs. However, in contrast to our work, their work focuses on the case of software product lines and their evolution during software life cycle. Our work focuses on evolution due to run-time adaptation. As in [11], this paper, argue that the focus of managing uncertain information should be on the rationale used to come to a decision. The decision may be taken either during design or requirements.

3. CLAIMS IN GOAL-BASED MODELS

In our LoREM (Levels of RE for Modeling), process a DAS is conceptualised as a set of distinct programs (target systems) designed to operate in a particular domain. Each target system and the conditions that trigger the DAS to adapt from one target system to another are explicitly modelled. An i* Strategic Rationale (SR) model is developed for each target system, showing the solution strategies employed by the DAS to achieve its goals and best satisfy its softgoals for a specific domain. This model is known as a level-one SR model, and forms the focus of the tracing work described in this paper. Underpinning LoREM is an assumption that a DAS must satisfice a set of NFRs, but that the balance of satisficement and the consequent trade-offs differs according to domain. Thus, an understanding of a DAS's adaptive behaviour can be reached by analysis of the different trade-offs made, and resultant configuration decisions, made in a set of LoREM level-one SR models

Figure 1 shows a level-one SR model for one of the target systems of an adaptive satellite navigation system to be installed on suitably mobile telephones. The system provides verbal navigation cues to a driver en-route to a destination selected in advance. Two types of navigation cues are supported: *turn navigation* and *lane navigation*. Using turn navigation, the system provides verbal instructions to take left or right turns, to join or leave a motorway as necessary. Using lane navigation, the system also gives the driver advance notice of which lane they need to be in, and instructions to change lanes should the driver be approaching a junction in the wrong lane. Lane navigation offers greater detail, reducing the chance of the driver having to make a dramatic, late lane switch to make a turn. However, these precise navigation instructions require accurate location data. Should the calculated location be out even by a few meters, the system may issue an incorrect instructions, potentially confusing the driver or even causing an accident.

The system supports two different sources of location data, with the phone's operating system switching between the two: a GPS receiver or cell-tower triangulation. The location supplied by a GPS receiver is relatively accurate, but requires line-of-sight to three or more GPS satellites. In built-up or hilly areas this requirement can be difficult to achieve. The location data supplied by the user's mobile network provider varies in accuracy according to the number of cell towers the phone is currently within range of: the more towers in range, the more accurate the triangulated location. The accuracy offered by cell-tower triangulation is inferior to that offered by GPS. Thus, the adaptive satellite navigation system's operating environment can be partitioned into two distinct domains: GPS Signal Available (D1) and GPS Signal Unavailable (D2). Figure 1 shows the level-one SR model for the S1 target system, to operate in domain D1.

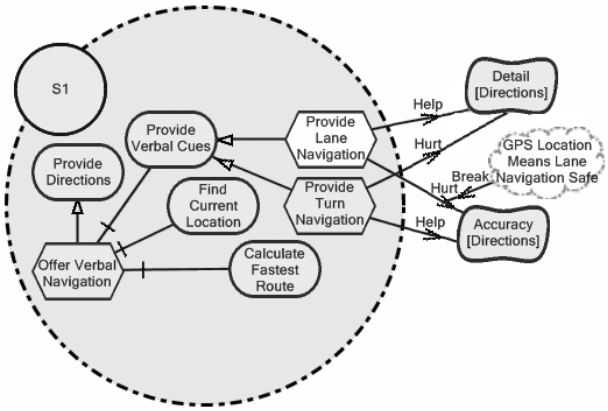

Figure 1: Level-One SR Model for S1 Target System

In Figure 1, the goal: "Provide Directions" is satisfied by the completion of the task: "Offer Verbal Navigation". This task is decomposed into the goals "Calculate Fastest Route", "Find Current Location" and, most importantly, "Provide Verbal Cues". The "Calculate Fastest Route" and "Find Current Location" goals could be modelled as having several alternate means of satisfaction, and are omitted here for simplicity and brevity. Our concern: the "Provide Ver-

bal Cues" goal, may be satisfied by completing either the "Provide Turn Navigation" task, or the "Provide Lane and Turn Navigation" task. The two softgoals, represented by the lozenges on Figure 1, are early manifestations of the system's NFRs: the accuracy and the detail of the provided directions. The system would naturally attempt to maximise both qualities, but an acceptable limit or measurement for either is ill-defined, hence their classification as softgoals.

In the D1 domain, the current location is found using the phone's GPS receiver. Thus, the accuracy of the location is relatively high. A relatively high degree of location accuracy means that, should lane navigation be used, there will be a relatively low incidence of incorrect navigation cues being issued based on faulty location information. To reflect this rationale a claim, represented by a cloud-like symbol, is included in Figure 1. The "GPS Location Means Lane Navigation Safe" claim is attached to the hurt contribution link running between "Provide Lane and Turn Navigation" and "Accuracy [Directions]" with a break contribution. Using this construction, the claim's rationale argues that the previously assumed consequence of providing lane navigation: that there is a greater risk of inaccurate navigation cues, given the greater dependency on accurate location information is insignificant in this domain, and should be overlooked. It is also possible to attach a claim to a contribution link on the model using a make contribution, which would argue that the attached contribution should be lent extra credence in this domain. With "Provide Lane and Turn Navigation" now having a help contribution link, and a hurt contribution link given less credence, the deadlock between the contribution links on the model is broken, and the choice to select "Provide Lane and Turn Navigation" when specifying the S1 target system becomes natural.

For the S2 target system, the level-one SR model is similar, with the exception of the claim discussed above. For the S2, a Cell-Tower Location Means Lane Navigation Unsafe" is attached to the hurt contribution link between "Provide Lane and Turn Navigation" and "Accuracy [Directions]". However, the claim in the S2 target system's model uses a make link, meaning that "Provide Lane and Turn Navigation" negative contribution is lent extra credence.

Claims [6] are used to argue rationale. Claims are simple statements of fact, and when attached to a contribution link speak to the contribution's importance. This differs from the use of fine-grained contribution links in i* models, which speak to a contribution's magnitude. We have argued [12] that the use of single claims in i* SR models for DASs improves the traceability of the models. Here we focus on the use of hierarchies of claims to scope uncertainty in DAS specifications, forward and backward tracing and the associated areas of uncertainty.

To use claims as markers for areas of uncertainty in the analyst's understanding of the operating environment, or the DAS's interaction with it, more detail is often required than that contained in the simple statement of fact contained in a single claim. While claims provide the rationale for selection of one strategy over another, the derivation of a claim may be obscure. A claim derivation is obscure if the logic of the claim is not obvious from its name. The NFR Framework [6] contained a mechanism for examining the basis of a claim hierarchically by ANDing and ORing combinations of supporting facts and assumptions, also modelled as claims in a claim refinement model. The refinement model for the example is depicted in Figure 2.

Figure 2: Simple Claim Refinement Model

The simple claim hierarchy depicted in Figure 2 shows that the "GPS Location Means Lane Navigation Safe" is derived from the combination of two supporting claims: "Lane Navigation Accurate if Given Accurate Location" and "GPS Location is Accurate". Should either supporting claim later turn out to be incorrect; the derived claim, as it appears on the level-one SR model (Figure 1), would also be incorrect. In this example, both of the supporting claims are essentially assumptions about the accuracy of GPS, and the required level of location accuracy for verbal lane navigation to be acceptably accurate. Without strong supporting evidence for these assumptions, the claims are representative of an area of uncertainty in the system's specification. The tracing to and from these areas of uncertainty that the next two sections of the paper discuss.

4. BACKWARDS TRACING

The claim depicted in Figure 1 is a rather broad assumption. By capturing this assumption, the implied uncertainty in the assumption's holding is available for tracing. The Claim Refinement Model in Figure 2 provides more detail on the assumption's basis, and thus narrows the scope of the captured uncertainty to that denoted by the two supporting claims. Examining Figure 2, it is clear that the uncertainty in the broader, derived claim stems from uncertainty about whether GPS location data really is accurate enough for lane navigation to be reliable, and whether lane navigation is reliable enough even when given relatively accurate location data. The use of a Claim Refinement Model thus not only serves to capture the basis of an assumption, but identifies the sources of uncertainty in the assumption's validity. Once identified, these sources of uncertainty, represented by claims, can be used as tracing artefacts.

It may be found during prototyping that lane navigation voice cues are too often incorrect, and sometimes multiple contradictory cues are issued, despite lane navigation only being used when the more accurate GPS location data is available. In this instance, it is a simple task to identify the target system to undergo scrutiny: lane navigation is only enabled in S1. Thus, the decision to use lane navigation in S1 is studied. The basis for the decision is clearly indicated by the claim "GPS Location Means Lane Navigation Safe" on the level-one SR model. This claim is, in turn, based on the two underlying claims depicted in Figure 2.

A brief analysis of the erroneous behaviour (the system enabling lane navigation when location data is not accurate) leads to the claim "GPS Location is Sufficiently Accurate" being found faulty. Thus, it is possible to use even a brief record of assumptions and their basis to trace backwards

from a sub-optimal system configuration to the faulty assumption (and associated uncertainty) responsible.

5. FORWARDS TRACING

As understanding of the environment and the system improves, and the scope of the uncertainty in the system's specification reduces, the claim refinement model is updated accordingly. By analysing the changed model, it is possible to trace forwards to (automatically) derive an updated specification for the system's adaptive behaviour.

Two pieces of information need to be traced to derive the part of the adaptive behaviour specification for a given transition between target systems: the differences in configuration between the two target systems, and the condition that triggers adaptation. Comparison between the level-one SR models for each of the target systems, allows the identification of the configuration differences between them, and thus the adaptations necessary to effect a changes.

The other aspect of the system's adaptive behaviour, the environmental properties monitored to trigger adaptation, can be obtained from the level-two model. Level-two models are produced on a one per-adaptation basis, and describe, amongst other things, the adaptation trigger. Figure 3 depicts the level-two model for the adaptive satellite navigation system's transition from the S1 to the S2. The model depicts three agents: the monitoring mechanism, the decision-making mechanism, and the adaptation mechanism.

The adaptation mechanism is responsible for effecting adaptation, the monitoring mechanism for providing and aggregating monitoring data, and the decision-making mechanism is responsible for analysing monitoring data and triggering adaptation. In Figure 3, the key element for our analysis: the environmental condition that should trigger an adaptation from S1 to S2, is "Set GPS_LOCATION = FALSE".

The trigger condition, combined with the adaptations required to transition between the two target systems derived from the level-one SR models, is sufficient to specify the relevant piece of system's adaptive behaviour. In this case, lane navigation is to be disabled when the "GPS_LOCATION" flag is set to "FALSE". For the inverse transition and its level-two model, the same analysis is performed.

The process of obtaining the information from the models is amenable to automation, and we developed a tool capable of analysing LoREM models created in the OME3 modelling tool [1]. The OME3-based tool allow us to generate either directly adaptation policies or Genie models [4]. Genie allows the systematic generation of adaptation policies and other middleware related artefact from models including components, component configurations and reconfiguration scripts. With these tandem approaches and tools we provide a tool-supported model-driven chain approach.

6. CONCLUSIONS

This has demonstrated that claims can be used to enhance the traceability of DAS models for use in an iterative RE process, preserving the decision rationale underpinning the behaviour of the DAS. The availability of this rationale in later iterations can ease in reviewing the decision in light of new and updated information. The paper has also presented a tool allowing claim-enriched DAS models to be accessed automatically to derive an adaptive behaviour specification expressed as Genie models or adaptation policies. This ability to derive an adaptive behaviour specification automatically

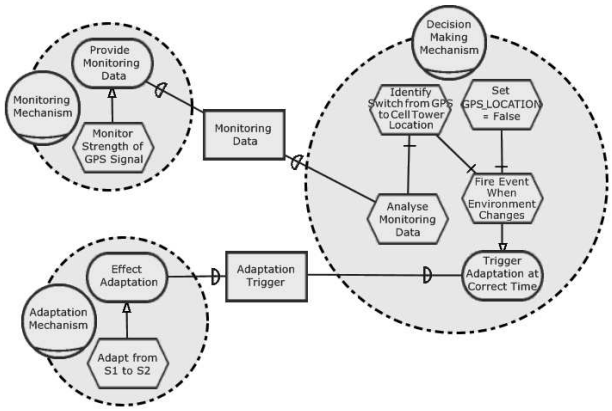

Figure 3: Level-Two Model for S1-S2 Transition

further reduces the difficulty in trialling candidate specifications during prototyping, and in making adjustments during maintenance. We are investigating the potential consultation of claims at runtime to allow backwards tracing performed by the systems itself (allowing self-explanation).

7. REFERENCES

[1] http://www.cs.toronto.edu/km/ome/.

[2] J. P. A. Almeida, M.-E. Iacob, and P. van Eck. Requirements traceability in model-driven development: Applying model and transformation conformance. *Information Systems Frontiers*, 9, 2007.

[3] N. Anquetil, B. Grammel, I. Galvao , J. Noppen, S. Shakil , H. Arboleda, A. Rashid, and A. Garcia. Traceability for model driven, software product line engineering. In *ECMDA*, 2008.

[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,*, volume 2, 2008.

[5] N. Bencomo, J. Whittle, P. Sawyer, A. Finkelstein, and E. Letier. Requirements reflection: requirements as runtime entities. In *ICSE*, volume 2, 2010.

[6] L. Chung, B. A. Nixon, and E. Yu. *Non-functional requirements in software engineering*. 1999.

[7] G. Cysneiros, A. Zisman, and G. Spanoudakis. Traceability approach for i* and uml models. In *SELMAS'03*, 2003.

[8] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Sci. Comput. Program.*, 20:3 – 50, 1993.

[9] S. Fickas and M. Feather. Requirements monitoring in dynamic environments. In *RE*, 1995.

[10] A. Lapouchnian, S. Liaskos, J. Mylopoulos, and Y. Yu. Towards requirements-driven autonomic systems design. In *Workshop DEAS*, 2005.

[11] M. M. Lehman and M. M. x Ramil. Software evolutionŮbackground, theory, practice. *Information Processing Letters*, 88:33 – 44, 2003.

[12] K. Welsh and P. Sawyer. Requirements tracing to support change in dynamically adaptive systems. In *REFSQ*, 2009.

[13] E. S. K. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *RE*, 1997.