

Fluent-Based Web Animation: Exploring Goals for Requirements Validation

Robert Chatley, Sebastian Uchitel, Jeff Kramer and Jeff Magee

Dept of Computing,
Imperial College London,
180 Queensgate,
London SW7 2AZ, UK

{rbc,su2,jk,jnm}@doc.ic.ac.uk

ABSTRACT

We present a tool that provides effective graphical animations as a means of validating both goals and software designs. Goals are objectives that a system is expected to meet. They are decomposed until they can be represented as fluents. Animations are specified in terms of fluents and driven by behaviour models.

Categories and Subject Descriptors

D.2.1 [Software Engineering]: Requirements/Specifications – *elicitation methods, tools.*

General Terms

Design, Experimentation, Human Factors

Keywords

Requirements, Design, Validation, Animation, Modelling

1. INTRODUCTION

Behaviour models are event level descriptions of how systems operate. They can describe how system components, the environment and users interact in order to provide system level functionality. However, being operational descriptions, they leave the required properties of the intended system implicit.

Goals focus on the objectives of the systems to be constructed. They are declarative statements and as such complement the operational nature of behaviour models. Goals can be used to elicit and elaborate operational descriptions of the required system.

Validation is a key design activity. Engineers must not only elicit and document operational models and goals, but also validate that these are indeed what stakeholders want [9]. Animation is an effective requirements validation technique, and so an approach that can provide model-based animations of goals is attractive. In this paper we present tool support for our approach for combining

goals with state models (as described in [10]). This can result in effective animations that not only support validation but can also prompt elaboration of a specification.

We believe that animations of system behaviour should be driven by the model and displayed from the perspective of the stakeholder participating in the animation in a way that relates to the stakeholder's goals. We use fluent linear temporal logic (FLTL) formulas [3] to express goals that can be formulated in terms of states controllable by the software system. These FLTL formulas have as basic elements fluents, which are central to establishing the relation between goals and models.

Fluents are abstractions of system state specified in terms of the occurrence of events. Miller and Shanahan [8] informally define (propositional) fluents as follows: "Fluents (time-varying properties of the world) are true at particular time-points if they have been initiated by an event occurrence at some earlier time-point, and not terminated by another event occurrence in the meantime. Similarly, a fluent is false at a particular time-point if it has been previously terminated and not initiated in the meantime." In the syntax accepted by our tool, fluents are specified as in the following example:

```
fluent LoggedIn =  
    <authenticate, {logout, disable}>
```

The LoggedIn fluent specifies that for a user to be in the logged in state, that user must have previously been authenticated by an authenticate action and that the user must not have logged out or been disabled.

Fluents are central to specifying animations. The same fluents that are used to express system goals are used to construct visualisation rules used by an animator to build graphical views of the system. This results in intuitive, simple visualisation specifications, and also in animations that support validation of goals and models together. By having stakeholders animate system behaviour viewed through abstract states defined in terms of fluents, not only is the operational model validated as it drives the animation, but also confidence in the validity of the system goals is gained.

Our tool supports animations that can be displayed in, and interacted with, using a standard Web browser. System states are captured by fluents and rendered as Web pages.

2. TOOL

Our visualisation tool is implemented as a plugin to the Labelled Transition System Analyser (LTSA) tool [6,7]. LTSA has been adapted to make use of the MagicBeans plugin architecture [1] so that extension modules can easily be written and added to the system. The Web Animator (visualiser) plugin provides an XML editor pane to LTSA where showwhen rules, defining the visualization, can be specified and edited. It also provides an implementation of a small web server, which allows stakeholders to view the visualisation in a standard web browser.

The animator component uses a behaviour model in the form of a Labelled Transition System (LTS) to react to events controlled by the animation participants. The animator also has access to the definitions of the fluents used in specifying goals. It keeps track of the values of fluents during an animation and uses these to construct webpages representing the system state. The visualisation specification is a mapping from fluents to visual elements. When the animator produces a visualisation of the system state, it composes all the visual elements that correspond to fluents that are true in that state. The visual elements also include active elements (such as buttons and hyperlinks) that are related to events that are controllable by the participant (see Fig. 1). These active elements allow participants to trigger the events they control.

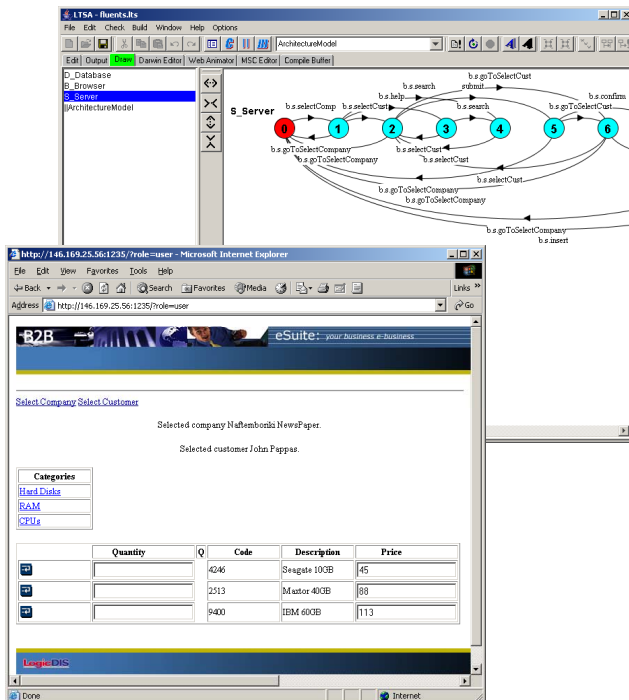


Figure 1 : eCommerce model being animated

3. EVALUATION

The idea of graphic animation based on a behaviour model is not in itself novel. Many verification tools provide the ability to execute a behaviour model as a way of simulating the system being modeled [4,5]

A novelty of our approach is the way that visualisations can be constructed based on abstract system states rather than the concrete states of the model. This allows for greater generality and flexibility, and allows engineers to produce animations that have a concrete relation to the goals of the participating stakeholder. We have found the techniques useful in practice, having applied them to the development of an eCommerce system in a case study with the Greek company LogicDIS.

In [10] we present the theoretical framework that underpins the tool, and also cover the use of scenario-based specifications as the source for behaviour models. The focus of [10] is on validation and elaboration of scenarios and goals. The tool is available for download from <http://www.doc.ic.ac.uk/ltsa>

4. REFERENCES

- [1] R. Chatley, S. Eisenbach and J. Magee, "MagicBeans: a Platform for Deploying Plugin Components" in 2nd International Working Conference on Component Deployment, Edinburgh, 2004.
- [2] R. Darimont and A. v. Lamsweerde, "Formal Refinement Patterns for Goal-Driven Requirements Elaboration" in 4th Symp. on Foundations of Software Engineering, San Francisco, 1996.
- [3] D. Giannakopoulou and J. Magee, "Fluent Model Checking for Event-Based Systems" in ESEC/FSE 2003, Helsinki, 2003.
- [4] D. Harel, et al. "STATEMATE: A Working Environment for the Development of Complex Reactive Systems", IEEE Transactions on Software Engineering, 16 p. 403-414, 1990.
- [5] C. Heitmeyer, C. Kirby, and B. Labaw, "The SCR method for Formally Specifying, Verifying and Validating requirements: Tool Support" in Intl. Conf. on Software Engineering (ICSE/97), Boston, 1997.
- [6] Magee et al., The LTSA site, www.doc.ic.ac.uk/ltsa
- [7] J. Magee and J. Kramer, Concurrency: State Models and Java Programs. New York: John Wiley & Sons Ltd., 1999.
- [8] R. Miller and M. Shanahan, "The Event Calculus in Classical Logic - Alternative Axiomatisations", Linköping Electronic Articles in Computer and Information Science, 4(16), p. 1-27, 1999.
- [9] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap" in Intl. Conf. on Software Engineering (ICSE/00), Limerick, 2000.
- [10] S. Uchitel, R. Chatley, J. Kramer and J. Magee, "Fluent-Based Animation: Exploiting the Relation between Goals and Scenarios for Requirements Validation" in Intl. Cong. on Requirements Engineering, Kyoto, 2004.