A State-Based Intention Driven Declarative Process Model

Pnina Soffer, Department of Management Information Systems, University of Haifa, Haifa, Israel

ABSTRACT

Declarative process models support process flexibility, whose importance has been widely recognized, particularly for organizations that face frequent changes and variable stimuli from their environment. However, the currently dominant declarative approaches lack expressiveness for addressing the process context (namely, environment effects) and leading its execution towards a goal. This paper proposes a declarative model which addresses activities as well as states, external events, and goals. The model is based on the Generic Process Model (GPM), extended by a notion of activity, which includes a state change aspect and an intentional aspect. The achievement of the intention of an activity may depend on events in the environment and is hence not certain. The paper provides a formalization of the model and describes an execution mechanism. It emphasizes the usefulness of specifying the intentional aspect of activities, by using it as a basis for semantic validation of the model at design time and for a planning module that can guide execution at runtime. These are illustrated by an example from the medical domain.

Keywords: Declarative Process Model, Generic Process Model, Goal, Intention, Notion of Activity, Process Flexibility

INTRODUCTION

The importance of flexibility in process aware information systems has been widely acknowledged in the past few years. Flexibility is the ability to make changes in adaptation to a need, while keeping the essence unchanged (Regev et. al, 2007). Considering business processes, flexibility is the ability to deal with both foreseen and unforeseen changes, by varying or adapting specific parts of the business process, while retaining the essence of the parts that should not be impacted by the variations (Schonenberg et al., 2008).

DOI: 10.4018/jismd.2013040103

Flexibility is particularly important in organizations that face frequent changes and variable stimuli from their environment. For processes that operate in a relatively stable environment, where unpredictable situations are not frequent, flexibility is not essential, as responses to all predictable situations can be defined. However, in the present business environment, where changes occur frequently and organizations have to cope with a high range of diversity, full predictability is rare.

Facing this reality, approaches have been proposed for enabling flexibility in business processes, as reviewed and classified in (Schenenberg et al., 2008). These include mechanisms of late binding and modeling, where the actual
realization of a specific action is only decided at runtime as implemented in YAWL (Aalst & Hofstede, 2005), and changes that can be made at runtime to a running process instance or to all instances of the process, enabled in ADEPT (Reichert et al., 2003).

One of the promising approaches is declarative process models (e.g., Declare (Pesic et al., 2007)), which have received significant attention in recent years.

While “traditional” process models are imperative (e.g., BPMN), explicitly specifying the execution order of activities through control flow constructs, a declarative process model implicitly specifies the execution procedure by means of constraints: any execution that does not violate constraints is possible. Using such model, the user can respond to each situation that arises, executing an activity chosen from all the ones available in compliance to the specified constraints. Currently, the most common approach to declarative process specification (although not the only one existing) is based on Linear Temporal Logic (LTL), which sets constraints on temporal relations among activities (Pesic et al., 2007). While allowing a high level of freedom, the approach has the following limitations.

First, while the human decision about which action to take is made based on the state at that specific moment, the existing models do not emphasize states. Rather, the leading concept to be modeled and monitored in the model is an activity, and constraints can be specified on the execution of a single activity or on relationships between activity executions. The process state is monitored, mainly as a trace of the activities that have been executed up to a given moment. Constraints can also relate to values of data as conditions for activity execution. However, there is no fundamental view and monitoring of state for leading process execution and decision making.

Second, to respond to changes and events in the environment, these need to be addressed in the model. Generally speaking, the model should be context aware, where context is the set of inputs a process instance receives from its environment (Ploesser et al., 2009). This is particularly important when bearing in mind that flexibility is required in the first place in processes that face frequent changes in the environment.

Finally, an effective selection of action by the human operator of the process should relate to the desired outcomes to be achieved, namely, to a goal. Currently, goals are usually not an integral part of process definitions.

Some proposals of state-based declarative models have also been made (e.g., Hull et al., 2011). However, these are either missing the concept of goal (Hull et al., 2011) or of environment effects et al., 2008). This paper proposes a declarative process model to overcome the three discussed limitations. To develop a consistent and complete model, we rely on the Generic Process Model (GPM) (Soffer and Wand, 2007), which is an ontology-based theoretical process analysis framework. GPM uses states as a leading element in process representation; it has been used for analyzing the context of processes (Ghattas et al., 2009), and it includes goals as basic building blocks of processes. Since GPM emphasizes states and abstracts from activities in process models, in this paper it is amended to cater for activities as well. We model activities by the state change they cause and by the intention that drives them. We use the intentional element of activities to drive validation at process design and as a basis for action selection at runtime.

In what follows, we start by a motivating example, demonstrating the limitations of a representative LTL-based declarative model. We then present the concepts required for our declarative model, first by informally deriving them from GPM, and then as formal definitions that set the basis for an execution mechanism. Then, the use of our concepts at process design and at runtime is presented and demonstrated through application to the running example. This is followed by a discussion, review of related work, and conclusions, also outlining future research directions.
Revisiting Software Engineering in the Social Era
www.igi-global.com/article/revisiting-software-engineering-in-the-social-era/177884?camid=4v1a

Model-Driven Exception Management Case Study
www.igi-global.com/chapter/model-driven-exception-management-case/37031?camid=4v1a