LEADSTO: A Language and Environment for Analysis of Dynamics by SimulaTiOn
(Extended Abstract)

Tibor Bosse[1], Catholijn M. Jonker[2], Lourens van der Meij[1], and Jan Treur[1]

[1] Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse, lourens, treur}@cs.vu.nl
http://www.cs.vu.nl/~{tbosse, lourens, treur}
[2] Nijmegen Institute for Cognition and Information, Division Cognitive Engineering,
Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
C.Jonker@nici.ru.nl

Abstract. This paper presents the language and software environment LEADSTO that has been developed to model and simulate dynamic processes in terms of both qualitative and quantitative concepts. The LEADSTO language is a declarative order-sorted temporal language, extended with quantitative means. Dynamic processes can be modelled by specifying the direct temporal dependencies between state properties in successive states. Based on the LEADSTO language, a software environment was developed that performs simulations of LEADSTO specifications, generates simulation traces for further analysis, and constructs visual representations of traces. The approach proved its value in a number of research projects in different domains.

1 Introduction

In simulations various formats are used to specify basic mechanisms or causal relations within a process, see e.g., [1], [2], [3]. Depending on the domain of application such basic mechanisms need to be formulated quantitatively or qualitatively. Usually, within a given application explicit boundaries can be given in which the mechanisms take effect. For example, "from the time of planting an avocado pit, it takes 4 to 6 weeks for a shoot to appear".

In such examples, in order to simulate the process that takes place, it is important to model its *dynamics*. When considering current approaches to modelling dynamics, the following two classes can be identified: *logic-oriented* modelling approaches, and *mathematical* modelling approaches, usually based on difference or differential equations. Logic-oriented approaches are good for expressing qualitative relations, but less suitable for working with quantitative relationships. Mathematical modelling approaches (e.g., Dynamical Systems Theory [3]), are good for the quantitative relations, but expressing conceptual, qualitative relationships is very difficult. In this

M. Ali and F. Esposito (Eds.): IEA/AIE 2005, LNAI 3533, pp. 363 – 366, 2005.

article, the LEADSTO language (and software environment) is proposed as a language combining the specification of qualitative and quantitative relations.

2 Modelling Dynamics in LEADSTO

Dynamics is considered as evolution of states over time. The notion of state as used here is characterised on the basis of an ontology defining a set of properties that do or do not hold at a certain point in time. For a given (order-sorted predicate logic) ontology Ont, the propositional language signature consisting of all *state ground atoms* (or *atomic state properties*) based on Ont is denoted by APROP(Ont). The *state properties* based on a certain ontology Ont are formalised by the propositions that can be made (using conjunction, negation, disjunction, implication) from the ground atoms. A *state* S is an indication of which atomic state properties are true and which are false, i.e., a mapping S: APROP(Ont) → {true, false}.

To specify simulation models a temporal language has been developed. This language (the LEADSTO language) enables one to model direct temporal dependencies between two state properties in successive states, also called *dynamic properties*. A specification of dynamic properties in LEADSTO format has as advantages that it is executable and that it can often easily be depicted graphically. The format is defined as follows. Let α and β be state properties of the form 'conjunction of atoms or negations of atoms', and e, f, g, h non-negative real numbers. In the LEADSTO language the notation $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, means:

If state property α *holds for a certain time interval with duration g, then after some delay (between e and f) state property* β *will hold for a certain time interval of length h.*

An example dynamic property that uses the LEADSTO format defined above is the following: "observes(agent_A, food_present) $\twoheadrightarrow_{2, 3, 1, 1.5}$ belief(agent_A, food_present)". Informally, this example expresses the fact that, if agent A observes that food is present during 1 time unit, then after a delay between 2 and 3 time units, agent A will believe that food is present during 1.5 time units. In addition, within the LEADSTO language it is possible to use sorts, variables over sorts, real numbers, and mathematical operations, such as in "has_value(x, v) $\twoheadrightarrow_{e, f, g, h}$ has_value(x, v*0.25)".

Next, a *trace* or *trajectory* γ over a state ontology Ont is a time-indexed sequence of states over Ont (where the time frame is formalised by the real numbers). A LEADSTO expression $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, holds for a trace γ if:

$\forall$t1: [$\forall$t [t1−g ≤ t < t1 ⇒ α holds in γ at time t] ⇒ $\exists$d [e ≤ d ≤ f & $\forall$t' [t1+d ≤ t' < t1+d+h ⇒ β holds in γ at time t']

An important use of the LEADSTO language is as a specification language for simulation models. As indicated above, on the one hand LEADSTO expressions can be considered as logical expressions with a declarative, temporal semantics, showing what it means that they hold in a given trace. On the other hand they can be used to specify basic mechanisms of a process and to generate traces, similar to Executable Temporal Logic (cf. [1]).

The LEADSTO language has been used in a number of research projects in different domains. It has been used to analyse and simulate behavioural dynamics of agents in cognitive science, biology, social science, and artificial intelligence. For publications about these applications, the reader is referred to the authors' homepages.

3 Tools

The LEADSTO software environment consists of two programs: the *Property Editor* and the *Simulation Tool*. The Property Editor provides a user-friendly way of building and editing LEADSTO specifications. It was designed in particular for laymen and students. The tool has been used successfully by students with no computer science background and by users with little computer experience. By means of graphical manipulation and filling in of forms a LEADSTO specification may be constructed.

The Simulation Tool can perform the following activities:

- Loading LEADSTO specifications, performing a simulation and displaying the result.
- Loading and displaying existing traces (without performing simulation).

Apart from a number of technical details, the simulation algorithm is straightforward: at each time point, a bound part of the past of the trace (the maximum of all `g` values of all rules) determines the values of a bound range of the future trace (the maximum of `f + h` over all LEADSTO rules).

Figure 1 gives an example simulation trace within the domain of psychotherapy. It demonstrates the power of LEADSTO to combine quantitative concepts with qualitative concepts. The result is an easy to read (important for the communication with the domain expert), compact, and executable representation of an informal cognitive model.

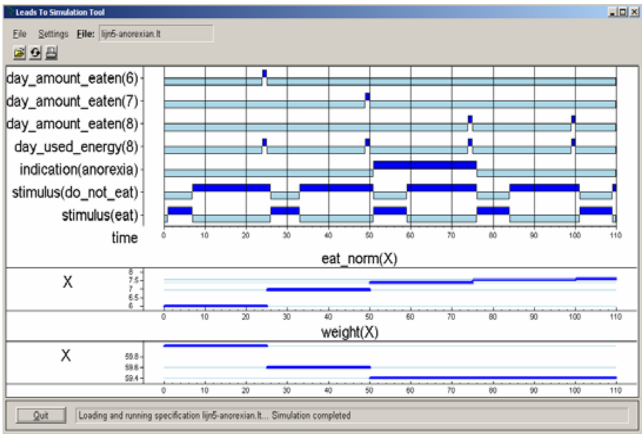

Fig. 1. Example simulation trace

4 Conclusion

This article presents the language and software environment LEADSTO that has been developed especially to model and simulate dynamic processes in terms of both qualitative and quantitative concepts. It is, for example, possible to model differential and difference equations, and to combine those with discrete qualitative modelling approaches. Existing languages are either not accompanied by a software environment that allows simulation of the model, or do not allow the combination of both qualitative and quantitative concepts.

Dynamics can be modelled in LEADSTO as evolution of states over time, i.e., by modelling the direct temporal dependencies between state properties in successive states. The use of durations in these temporal properties facilitates the modelling of such temporal dependencies. Main advantages of the language are that it is executable and allows for graphical representation.

The software environment LEADSTO proved its value for laymen, students and expert users in a number of research projects in different domains.

References

1. Barringer, H., M. Fisher, D. Gabbay, R. Owens, & M. Reynolds (1996). *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press Ltd. and John Wiley & Sons.
2. Forbus, K.D. (1984). *Qualitative process theory*. Artificial Intelligence, volume 24, number 1-3, pp. 85-168.
3. Port, R.F., Gelder, T. van (eds.) (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, Mass.