The Knowledge Evolution Group at CENTRIA

João Leite
CENTRIA, New University of Lisbon, Portugal

1 Overview

The *Center for Artificial Intelligence* (*CENTRIA*) is a research center of the New University of Lisbon, funded by the Portuguese Ministry for Science and Higher Education.

The Knowledge Evolution Group is a sub-group of the Knowledge Representation and Reasoning Group at *CENTRIA*, which devotes its efforts to investigating the foundational concepts related to the representation of, and reasoning about, evolving knowledge, i.e. knowledge that changes with time as a consequence of successive updates, and their application in areas such as Multi-Agent Systems, the Semantic Web, and Recommender Systems.

2 Developments

As a result of our investigations, over the last few years, several extensions to *Answer-Set Programming* (*ASP*) were proposed, aimed at addressing the issue of representing and reasoning about evolving knowledge. Two of these extensions are *Dynamic Logic Programming* and *Evolving Logic Programming.*

Dynamic Logic Programming

Dynamic Logic Programming (*DLP*) is an extension of Answer-Set Programming that allows for the specification of logic program updates.

According to *DLP*, knowledge is given by a series of theories, encoded as generalized logic programs (i.e. extended LPs, or answer-set programs, plus default negation in rule heads), each representing distinct states of the world. Different states, sequentially ordered, can represent different time periods, thus allowing *DLP* to represent knowledge undergoing successive updates. As individual theories may comprise mutually contradictory as well as overlapping information, the role of *DLP* is to employ the mutual relationships among different states to determine the declarative semantics, at each state, for the combined theory comprised of all individual theories. Intuitively, one can add newer rules at the end of the sequence, leaving to *DLP* the task of ensuring that these rules are in force, and that previous ones are valid (by inertia) only so far as possible, i.e. they are kept for as long as they are not in conflict with newly added ones, these always prevailing.

Besides its appropriateness for representing knowledge updates, *DLP* contributes to the use of Logic Programming in a modular way. This modularity has been taken one step further with the development of Multi-Dimensional Dynamic Logic Programming, where a theory is given by a set of generalized logic programs combined according to a partial order relation between them.

Evolving Logic Programming

In a nutshell, *Evolving Logic Programming* (*EVOLP*) is a simple though quite powerful extension of ASP that allows for the specification of a program's own evolution. These evolutions arise both from self (i.e. internal to the program) updating, and from external updating originating in the environment.

From the syntactical point of view, evolving programs are just generalized logic programs, extended with (possibly nested) assertions in either heads or bodies of rules. From the semantical point of view, a model-theoretic characterization is offered of the possible evolutions of such programs by means of the so-called *evolving stable models* (or evolving answer-sets) which are sequences of interpretations (or sets of atoms). Each interpretation in the sequence describes, at the corresponding evolution step, what is true, and the possible next-step evolutions.

EVOLP provides a simple and general formulation of logic program updating, which runs close to traditional LP doctrine, setting itself on a firm formal basis in which to express, implement, and reason about evolving knowledge bases, opening up several interesting research topics.

3 Applications

Concurrently with the developments of these Answer-Set Programming extensions, we have also devoted some effort in the investigation of their application focusing in the areas of *Multi-Agent Systems*, the *Semantic Web*, and *Recommender Systems*.

Multi-Agent Systems

Agents are autonomous active problem solving entities situated in open and dynamic environments, often required to exhibit reactive, pro-active, adaptive and social behaviours.

The agent paradigm, commonly implemented by means of imperative languages for reasons of efficiency, has increased its influence in the research and development of computational logic-based systems over the last decade. Since efficiency is not always the crucial issue, but clear specification and correctness is, *Logic Programming* and *Non-monotonic Reasoning* have been able to find in

Multi-Agent Systems a suitable application area filled with challenging issues to be addressed.

Agents must keep beliefs about their goals, intentions, capabilities and the environment in which they are situated. Furthermore, these beliefs must be dynamic, not only because the agent may learn about static features of its environment and new ways to behave, but also because of the intrinsic dynamic character of the environment, which may be populated with other autonomous agents.

We have devoted some effort in the application of *DLP* and *EVOLP* in Multi-Agent Systems. Several different topics in this area were addressed, having as common ground the qualities of Answer-Set Programming and *DLP* as knowledge representation tools, with features of key importance in open and dynamic worlds, such as their ability to deal with incomplete information, and of *EVOLP* as a language in which to represent state transitions and behaviour evolution. In particular, we have shown the benefits of *ASP*, *DLP*, and *EVOLP*, by extending existing Multi-Agent Systems (e.g. *3APL* and *MadAgents*) to allow for enhanced representation of their beliefs, goals and behaviour, and by developing, from scratch, the *MINERVA* agent architecture, designed to provide a common agent framework based on the strengths of Logic Programming, to allow for the combination of several existing non-monotonic knowledge representation and reasoning mechanisms.

Semantic Web

Even though one major goal of the Semantic Web is to bridge the heterogeneity of data formats, languages, and behaviour, and to provide unified views of the Web, the Semantic Web should also be able to support, besides querying, knowledge propagation and change in a semantical way. In other words, the Semantic Web should also be able to support knowledge evolution.

When taking the Web as a "living organism" consisting of autonomous knowledge sources, appropriate languages for the specification of updates and evolution are in order. These are needed, for example, to specify the update of the contents of some Web site (e.g. as a reaction to some user's actions), to specify the change in the behaviour of some Web site (e.g. in adaptive Web systems as a reaction to the change of the user model). Furthermore, these languages must cater for changes to be propagated to other Web sites.

We have been actively developing languages for the specification of reactivity and change that are suitable to deal with the particular issues found in the Semantic Web. For this purpose, general Event-Condition-Action rule-based languages have been investigated, where events may either be (internal) update events, or (external) message events, and actions may include (local) updates of the knowledge, remote actions (potentially structured by transactions), synchronization actions, and possibly "blackbox" actions.

Recommender Systems

Recommender systems are programs that help the user in navigating large volumes of available information, attempting to provide a solution to the user's needs, or suggesting items that the user may like.

To choose or discard existing items, recommender systems adopt many different approaches such as collaborative filtering, content-based filtering, etc. The use of explicit models for both products and user preferences is often suggested as a possible way to improve the quality of recommendations

We have proposed and investigated the introduction of *DLP* in recommender systems as a means for users to specify and update their individual models and preferences, with the purpose of enhancing recommendations. Valuable features of *DLP*, most of which are inherited from ASP, include: a simple extendable language; a well defined semantics; the possibility to use default negation to encode non-deterministic choice; the combination of both strong and default negation; easy connection with relational databases; and support for explanations.

4 Current and Future Activities

Our current and near-future projects include the development of *EVOLP* along three axes: (1) Declarative goal-driven knowledge evolution, (2) event-driven knowledge evolution, and (3) knowledge evolution under conditions of uncertainty. In parallel, we will continue to investigate the application of these developments in, at least, the areas of Multi-Agent Systems, the Semantic Web and Recommender Systems.

Declarative goal-driven evolution

Goals partially define the behavior of pro-active agents, and impose structure on the evolution of knowledge. Declarative goals require confirmation that a goal is met or notification that it is impossible to meet, and mechanisms for revising conflicting goals and recommending which conflicting goals should be altered. *EVOLP* appears ideally suited to provide a declarative representation of goals and their evolution. Furthermore, the extension of *EVOLP* with events (see below) may provide the capacity for representing richer, more complex goals, by allowing for temporal aspects necessary for procedural knowledge and the construction of plans. Our hope, then, is to combine current, more reactive behavior of *EVOLP* with a new representation framework for goals, plans, and pro-active behaviour.

Event-driven evolution

Event-condition-action rules are an important paradigm in logic-based reasoning and reactive systems. Recently, Event-driven-Architectures are returning as an important variant of Service Oriented Architecture, particularly in enterprise integration applications. For instance, a range of middleware capabilities for event transformation, aggregation, split and composition, and complex event processing are written into standards such as COBRA and JMS, and into commercial systems. We are working to provide *EVOLP* with event specification and processing capabilities by focusing on extending the expressiveness of trigger conditions for updates and actions within *EVOLP*.

Probabilistic evolution

EVOLP can represent fixed and conditional constraints, and compute possible event-outcomes when partial knowledge yields incomplete constraints. But there are other sources of uncertainty besides incomplete knowledge. Uncertainty may attach to particular pieces of knowledge, to possible event-outcomes, or to the desirability of an event-outcome given changes in its assessment of likelihood. To represent such cases of uncertainty it is natural to turn to probability. But there are many interpretations of probability, and a probabilistic semantics for uncertain knowledge representation depends on the interpretation we adopt and the type of uncertainty we wish to represent. We are investigating which probabilistic features we can import into *EVOLP*. Although several probabilistic interpretations of logic programming exist, none are for languages with the expressive capacity of *EVOLP*. The introduction of probability to *EVOLP* is very challenging in large part because the flexibility of the *EVOLP* language invites several distinct representations of uncertainty.

5 Wrap Up

This overview is just a partial take of the research being conducted at *CENTRIA* in the area of Knowledge Representation and Reasoning. Furthermore, the current topics of research are evolving, as expected, driven both by our own evolving interests, as well as those brought by new members of our research group.

If you are interested in any of these topics, or see any opportunities for collaboration, we would be very happy to hear about it and would most welcome your contact (see contact details below).

If you would like to study with us, we have opportunities both at MSc level as well as PhD level. At MSc level, the New University of Lisbon (and *CENTRIA*) is one of the five institutions involved in the European Master Program in Computational Logic (link below). At PhD level, opportunities are discussed on a case-by-case basis. In either case, we welcome your contact.

More information on all the topics mentioned, including the group's publications, can be found at the author's and *CENTRIA*'s web-sites.

- Author's web-site:
  `http://centria.di.fct.unl.pt/~jleite`
- *CENTRIA*'s web-site:
  `http://centria.di.fct.unl.pt`

- Author's email address:
  `jleite@di.fct.unl.pt`

- European Master in Computational Logic:
  `http://ssdi.di.fct.unl.pt/masters/mcl/`