Context through Answer Set Programming

Angelo Rauseo  
Politecnico di Milano  
Dipartimento di Elettronica e Informazione  
Via Ponzio 34/5 - 20133  
Milano, Italy  
rauseo@elet.polimi.it

Davide Martinenghi  
Politecnico di Milano  
Dipartimento di Elettronica e Informazione  
Via Ponzio 34/5 - 20133  
Milano, Italy  
martinen@elet.polimi.it

Letizia Tanca  
Politecnico di Milano  
Dipartimento di Elettronica e Informazione  
Via Ponzio 34/5 - 20133  
Milano, Italy  
tanca@elet.polimi.it

ABSTRACT

In a world of global networking, the variety and abundance of information generates the need for effectively and efficiently gathering, synthesizing, and querying the available data while removing information noise. The concept of context has been developed and refined since the first approaches to ubiquitous computing [3], the research area of everywhere computing systems, which has the objective to help provide information to users in an almost imperceptible way. At first, the idea of context was limited to time and location; later it was extended also to the other external environmental factors, current trends and phenomena that may change or influence the information and services available to a user. In this work, we refer to context primarily in relation with the effects that this notion has over data.

A system where context awareness is integrated with – yet orthogonal to – data management, allows the knowledge of the context in which the data are used to drive the process of focussing on currently useful information (represented as a view), keeping noise at bay: this activity is called context-aware data tailoring [1]. To do this, we model the context as a first-class citizen, by means of a tree-shaped structure called Context Dimension Tree (CDT). Formally defined in [1], the CDT is composed of two kinds of nodes: black nodes, which represent context dimensions, and white nodes, which represent context values. Context dimensions (black nodes) model the different perspectives from which the domain of interest can be seen with respect to the user, the system and their interactions. The values these dimensions can assume are represented as white nodes. A context is obtained as a set of dimension values, thus of white nodes. The hierarchical nature of the CDT grants different levels of abstraction to represent contexts.

In this paper we propose Answer Set Programming (ASP) as a unified tool to address all the design-time and run-time tasks related to context management. Building on preliminary work presented in [2], at design time we encode contextual information via a disjunctive logic program using the approach of ASP: given a program representing a CDT, each admissible context is represented by a model (also called answer set) of the program. From a CDT many possible contexts can be built, obtained by guessing if a value is a context element or not. Structural constraints to the composition of contexts also apply, along with application-dependent constraints preventing meaningless combinations of white nodes. Candidate contexts, which are candidate models of the program, are checked against the constraints defined, and the possibly multiple admissible contexts correspond to the multiple models of the program in accordance with ASP fundamentals. Still at design time, the designer extends the program with suitable context-aware views over the extensional data, which establish the relationships between each context and the data to be focussed upon. At run time, whenever new context information is acquired from the system conditions and/or detected by sensors, we use ASP techniques to i) validate this information against the admissible contexts produced at design time from the CDT, and ii) evaluate user queries over the current context-dependent views.

Overall, the main contribution of this work is the design of an ASP framework for supporting design-time and run-time data tailoring in context-aware systems. Notice that ASP allows us to retain the orthogonality of context modeling while adopting the same framework as for data representation. Moreover, the multiplicity of answers permits support for the imperfection of the gathered context information since, when multiple (possibly incompatible) contexts are detected, the system will be able to deliver all the answers to the user queries, each answer differently focussed according to one of the validated contexts.

1. REFERENCES

