Behavior Modeling and Reasoning for Ambient Support: HCM-L Modeler

Fadi Al Machot, Heinrich C. Mayr, and Judith Michael

Application Engineering Research Group, Alpen-Adria-Universität Klagenfurt, Austria
{fadi.almachot,heinrich.mayr,judith.michael}@aau.at

Abstract. This paper introduces the architecture and the features of the HCM-L Modeler, a modeling tool supporting the Human Cognitive Modeling Language HCM-L and a comprehensive reasoning approach for Human Cognitive Models based on Answer Set Programming. The HCM-L tool has been developed using the ADOxx® meta modeling platform and following the principles of the Open Modeling Initiative: to provide open models that are formulated in an arbitrary, domain specific modeling language, which however is grounded in a common ontological framework, and therefore easily to translate in another language depending of the given purpose.

Keywords: Modeling Tool, Modeling Language, Behavior Modeling, Ontology, Model Mapping, Knowledge Base, Ambient Assistance, Reasoning, Answer Set Programming.

1 Introduction

Health is essential for individuals as well as for the society. Many everyday activities help to keep healthy: Washing hands after using the bathroom, body care, preparing healthy food and doing sports. However, when getting older, such activities in general become harder.

Ambient Assisted Living (AAL, [1]) focusses on softening or even compensating the effects of ageing. AAL research was as a key topic within the 7th Framework Programme of the European Union; and again, health, demographic change and well-being are important topics in the new funding program Horizon 2020. The range of related research projects is wide and covers mobility support, smart homes, fall protection, cognitive games, healthcare, and much more.

To assess the ability of a person doing her or his daily activities, scales like the Physical Self-Maintenance Scale or the scale of Instrumental Activities of Daily Living (IADL) [2]) may be used. Using the former, e.g., the abilities in toileting, grooming, or bathing may be assessed; the IADL rates, among others, the competences in food preparation or the responsibility for the personal medication.

To support such activities is the main objective of the Human Behavior Monitoring and Support (HBMS[1] [3]) project. The key idea here is to use models of a person's

[1] Funded by Klaus Tschira Stiftung gGmbH, Heidelberg.

A. Moonis et al. (Eds.): IEA/AIE 2014, Part II, LNAI 8482, pp. 388–397, 2014.

former individual target-oriented behavior including its context as a knowledge basis for deriving support services when the person's abilities are reduced or (temporarily) forgotten.

Consequently, behavior modeling as well as reasoning for deriving support information from models are key topics of HBMS. As standard modeling languages like UML or BPMN proved to be suboptimal for the given purpose [4] the Human Cognitive Modeling Language (HCM-L) was designed as a lean set of concepts and constructs for integrated human behavior and context modeling. According to the OMI (Open Modeling Initiative [5]) HCM-L is a domain specific language based on common fundamentals, and thus allows for using models as knowledge carriers. It should be intuitively to understand by users of the AAL domain, in particular e.g. by psychologists or care givers.

HCM-L Modeler is a comprehensive tool for creating, managing and transforming models based on HCM-L. It was developed using the ADOxx® meta modeling platform [6]. Given the comprehensiveness of HCM-L, the Modeler might be useful for many application purposes beyond the borders of HBMS.

Modeling and reasoning are closely related and widely used in expert systems. In particular, intensive research is undertaken in the context of Semantic Web[2] [7,8,9]. Ontologies based on Semantic Web provide concise high-level semantic representations. However, parsing larger ontologies for recognition and heuristic problem tasks does not perform well with limited hardware resources [10,11,12].

Other supervised learning methods like Support Vector Machine, Artificial Neural Networks, Decision Trees, Bayesian Networks and Hidden Markov Models need training data that are quite difficult to extract from human cognitive models.

Within this paper we are targeting a reasoning approach for real time intelligent user support based on Answer Set Programming.

The paper is organized as follows: Section 2 introduces exemplarily the main HCM-L concepts and their representation using the HCM-L modeler. Section 3 discusses the architecture of the modeler and outlines its main features and functionality. Section 4 presents the overall architecture and the components of the proposed reasoning approach and a list of the obtained results. The paper closes with a conclusion and an outlook on future work (section 5).

2 HCM-L and Modeler

As Ambient Assistance in healthcare and other AAL domains mainly deals with supporting activities these are in focus when building ontologies (conceptualization) and modeling. Consequently, using HCM-L means modeling (the sequence of) activities together with their pre- and post-conditions first, and then adding models of the activities' personal, environmental, social, and spatio-temporal context [13].

As this paper addresses the HCM-L Modeler and reasoning, we introduce the underlying modeling language by means of an example instead of a systematic language definition as given in [4]. The graphical elements were chosen following the principles for designing effective visual notations [14].

[2] `http://www.w3.org`

Figure 1 shows a simplified example of the HCM-L key concept *behavioral unit* describing how a person, let's call her Paula, may achieve a given *goal*, here: "get ready for leaving the house" in the morning. As the reader might easily deduce from the figure, a behavioral unit represents an aggregate of *operations* which together lead to reaching a goal in daily life. As such, behavioral units correspond to use cases in business process modeling and can be broken down into different steps: the operations and the flows between these.

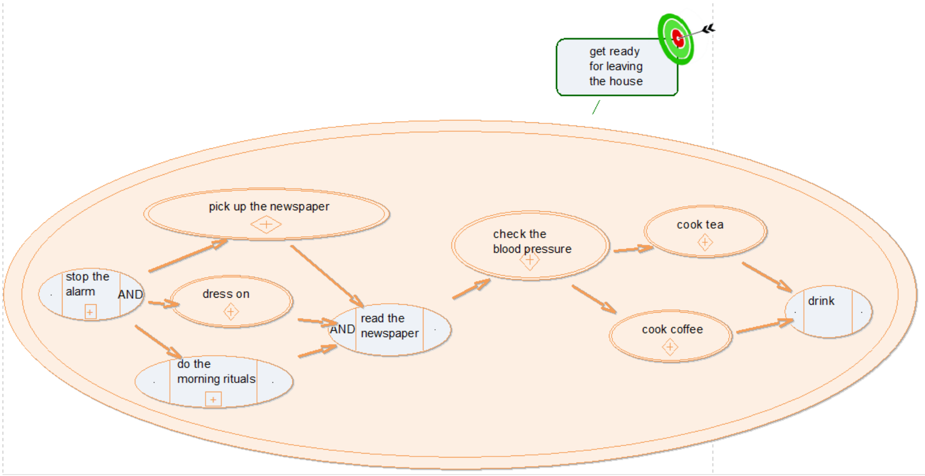

Fig. 1. Behavioral unit ‚morning activity'

Paula starts with stopping the alarm of the alarm-clock. Every morning, in an arbitrary order, she picks up the newspaper from the doormat in front of the entrance door, dresses on and does her morning rituals in the bathroom. After having done all that, she reads the newspaper in the kitchen. After 20 Minutes of reading, she checks her blood pressure. Depending on the result, she cooks and drinks tea (high blood pressure) or coffee (lower pressure).

P*ossible beginning* and *successful ending* operations are just marked implicitly. The former have no *incoming flow*, the latter no *outgoing flow*. If one of the successful ending operations (there might be more than one) is executed, the goal is fulfilled.

Operations can have simple or complex pre - and post-conditions; in this case they are grey-shaded and have condition compartments. Precondition (AND) of operation 'read the newspaper' defines, that this operation can only be executed, if all three predecessor operations are executed completely. How these operations are performed will be defined in the *instruction* attribute of each operation. That Paula has to check the blood pressure at least 20 minutes after she has started to read the newspaper will be defined in the *time space* attribute.

The diamond at the bottom of an operation symbol like in ‘check blood pressure’ or ‘dress on’ indicates that the resp. operation is seen and modeled as a behavioral unit again. Clicking on the diamond thus opens the related model (see figure 2): To check the values, Paula takes the blood pressure monitor out of the top draw of her drawer cabinet in the kitchen, secures the cuff, presses the start button and after some seconds hears a beep signal and can record the results on her tablet PC.

The modeling granularity depends on the resp. objectives. In the context of HBMS, the behavioral unit models will be established by transforming and integrating results from sensor/video observations of concrete sequences of actions.

HCM-L modeler also offers a possibility to condense/expand a sequence of operations (square in the bottom of an operation symbol) for enabling clarity of larger graphs. This, however, is a pure syntactical aid and does not define a hierarchy of behavioral units.

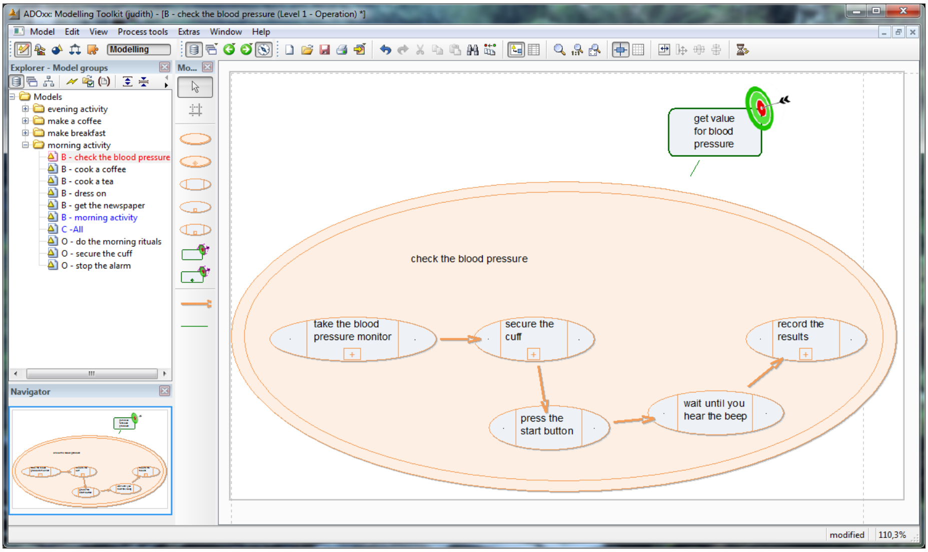

Fig. 2. Behavioral Unit ‚check the blood pressure’

Operations are performed within a personal, environmental, social, and spatio-temporal context [13]. The main HCM-L concepts for context modeling are *thing* and *connection* to describe arbitrary concrete or abstract objects, also persons, or relationships between things, respectively. Figure 3 shows a selection of the context of the “check the blood pressure” operations.

In this example, Paula is modeled as a person (thing box with smiley) with her left and right hand (aggregation). The blood pressure monitor is modeled as a thing, which again is made up of things, e.g. the start/stop button. The blood pressure monitor lays in the drawer cabinet which is located inside the kitchen (a *location thing*, flagged by the map symbol).

There are several relationships between operations and their context: *calling* (a thing initiates an operation), *participating* (things contribute to or are manipulated by an operation), and *executing* (a thing performs an operation).

The main HCM-L concepts are summarized in the meta model as discussed in [4].

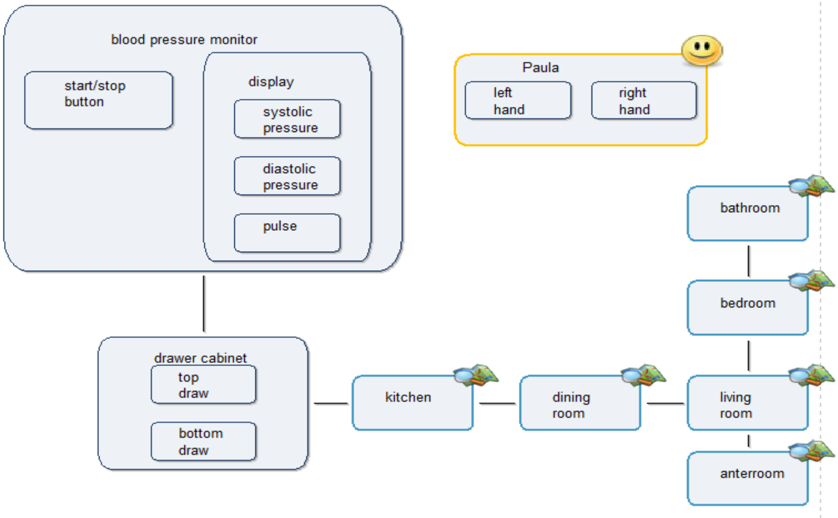

Fig. 3. Some elements of the context model

3 Architecture and Features

HCM-L Modeler is a comprehensive modeling tool for HCM-L including syntax, semantics and consistency checking. In the next development stage it will support complex scenarios, model optimization and advanced reasoning techniques.

HCM-L Modeler was developed using the meta modeling platform ADOxx® (www.ADOxx.org) which implements the upper three layers of the OMG Meta Object Facility (MOF) [15,16,17]. Figure 4 illustrates the overall architecture. On the third MOF layer the ADOxx® Meta^2 Model defines constructs such as Class, Relation Class, Endpoint, or Attribute.

The Meta^{1}Model (on the MOF M2 layer) corresponds to the meta model of the language, a modeling tool is to be developed for, i.e. in our case the HCM-L meta model. For that purpose, the corresponding ADOxx® tier provides the concept of library, representing a collection of meta models conforming to the Meta^{2}Model and formulated in the ADOxx® Library Language (ALL).

Models on MOF M1 layer (HCM-L models) are stored in a model repository, i.e. a generic model storage configured by a meta model library [16]. They can be exported in two formats, the ADOxx® Description Language (ADL) or in XML [17].

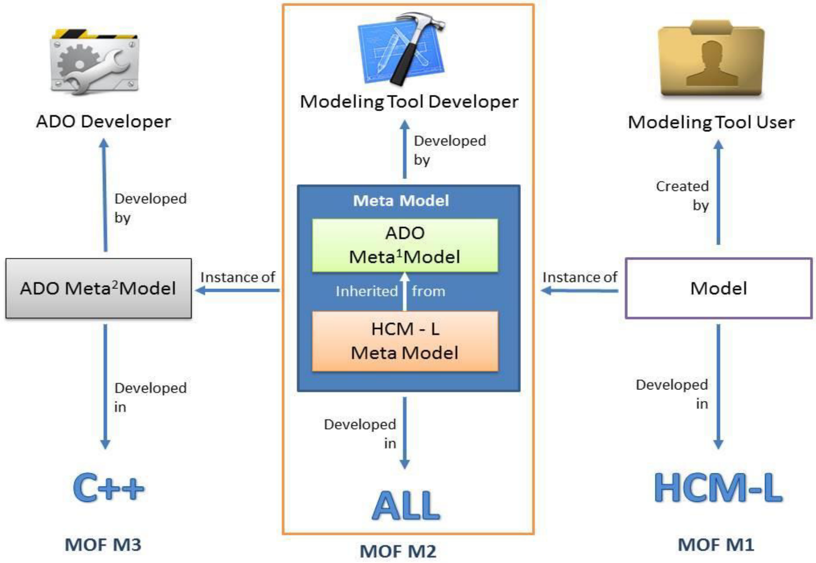

Fig. 4. HCM-L in the ADOxx® meta model hierarchy

We now proceed to the description of some key features and their implementation.

Objects and Relations

ADOxx® offers different types of meta constructs [18], out of which we selected three for the HCM-L modeler:

- *D_Construct*: Super class for a graph-base pre-defined meta model.
- *D_Container*: Container class providing the relation *is_inside*, i.e. O *is_inside* C means that the x/y coordinates of object O lie within the drawing area container C.
- *D_Aggregation*: Inherits from D_Container, hence also provides the *is_inside* relation; in addition, it enables a self-defined drawing area, e.g. a resizable rectangle.

Relation class is a construct that is used as a template for creating directed relations between objects; consequently, a relation class is defined between classes. As the relations are a directed, they have a from-side and a to-side.

Visualization

For the definition of the graphical elements ADOxx® provides the GRAPHREP grammar language which can be used to specify all graphical properties (e.g., shape and color) of a modeling language element.

Comprehensive mathematical support for drawing curves and polygons is provided. Furthermore, the developer is supported in controlling the coordinates of element's names, break-line properties and font types.

Consequently, using the HCM-L modeler, all objects and relations can be connected in a flexible way. The breaking lines of object's names give the possibility to resize the elements for an optimal visual form. This flexibility is a very important advantage of HCM-L where different models and sub-models are required for complex scenarios, e.g., the visualization of different levels of an aggregation hierarchy.

Traceability
Traceability relationship types are modeled as instances of the Meta²Model construct "Relation Class". HCM-L modeler uses *Interref Relation Classes* that can be used to connect elements of different models. The definition of such traceability relationships is supported by the so-called *notebook*, and thus can be done by the user.

Querying
An important function of human cognitive modeling tools is the support of querying the model for different analysis purposes. The ADOxx® platform provides the AQL query language that allows queries on models in a style similar to SQL12 [17]. AQL queries can be pre-defined by the developer or may be formulated manually by a user; for that purpose, HCM-L modeler provides an interactive assistant.

Consistency Check
A major issue in modeling processes is the fact that comprehensive consistency checks are difficult, in particular for inexperienced users. However, inconspicuous mistakes in the logic may affect the whole model: contradictory semantics reduce the performance of reasoning processes and yield invalid results. For the HCM-L modeler we considered two main consistency issues: (1) using the right syntax of logical operators and (2) consistent naming of model elements throughout the whole model.

As an example, after clicking on the button "pre-defined queries", HCM Modeler yields a menu of different consistency checks for every model and sub model.

Transformation
ADOxx® offers the possibility to import and export models in a generic XML format. This feature is adopted by HCM-L modeler in order to allow transforming models to other formats, as used e.g. by inference or reasoning tools.

Reasoning Support
Both, model and rule based reasoning approaches for ambient support require the extraction of different features out of the given overall model. HCM-L modeler, among others, offers the possibility to calculate the frequency of specific activities based on the user history: every operation is supported by a percentage value which will be used for reasoning purposes (see next section).

In addition to that, HCM-L modeler calculates for each operation the "importance value" based on the user history, and "cost value" based on the similarity between the current user profile and other users. In the next section these values will be discussed in detail.

4 Ambient Support by Reasoning Based on HCM-L Models

For ambient support using conceptual cognitive models different requirements have to be covered, e.g., reasoning over time, constraints, and optimization.

Answer set programming (ASP) [19,20] is a form of declarative programming oriented towards difficult (primarily NP-hard) search problems. It is based on the stable model (answer set) semantics of logic programming. In ASP, time is usually represented as a variable the values of which are defined by an extensional predicate with a finite domain. Finite temporal intervals can be used to reason in ambient support. Optimization is indicated via maximization and minimization. Adding a constraint to a logic program P affects the collection of stable models of P in a very simple way: it eliminates the stable models that violate the constraint.

Figure 5 shows the components of the proposed ambient support reasoning approach. First we use the HCM-L modeler to model and design the conceptual cognitive model, and then we generate the required reasoning parameters that will be added to the model in our HCM-L tool automatically. Finally, we export the model in XML format as input for the answer set programming solver (i.e., its knowledge base).

Fig. 5. Components of the ambient support reasoning approach

The purpose of our reasoning module is to support users (patients or old people) in choosing the next operation (activity) when desired. In the case depicted in Fig. 1, the reasoning system would have to propose Paula on of the operations (1) picking up a newspaper, (2) doing morning rituals or (2) dressing on.

For that purpose, an optimization problem is to be solved based on three priority measures: (1) the importance of performing an operation according to the user history; (2) the cost value of choosing an operation based on the similarity between the current user profile and other users; (3) the time when the operation should be performed. Consequently, an operation is represented in our knowledge base as follows:

```
operation(Id).
operation_time(OperationId,Time).
user_hist_importance(OperationId,ImportanceValue).
cost(OperationId,CostValue).
bad_timing(OperationId).
user_current_time(Time).
```

The cost value is calculated based on the determined set of similar users. A common measure for such similarity is Pearson's correlation coefficient [16]. Based on the user history, a matrix R is established consisting of the scores of operations the users perform. The score is incremented by 1 each time the user chooses the particular operation. Dividing this score by the total number of operations a particular user is doing per day gives the probability of choosing that specific operation. Consequently, the cost value is the complementary probability of the resp. score probability.

The user_hist_importance is chosen to be the average value of performing the given operation over the last 20 days; it also could be fed into the model by the modeler.

To decide about timing we define two cases for the given example: if the operation is done in the morning then it is considered as good timing otherwise as bad.

The Optimization Process Based on ASP

To find the optimal solution for our optimization problem, we consider the reasoning values of the operations discussed previously:

```
1. #maximize[operation(X):user_hist_importance(X,Y)=Y @3].
2. #minmize[operation(X):cost(X,Y):user_hist_importance(X,Z)=Y/Z@2].
3. #minimize {bad_timing @1}.
```

Line 1–3 contribute optimization statements in inverse order of significance, according to which we want to choose the best operation. The most significant optimization statement (line 1) gives the main priority to the habits' history of the user. Line 2 is to minimize the cost per operation with respect to the importance of user's history. Line 3 serves to minimize the number of operations with bad timing.

Obtained Results

To check our ASP-based reasoning approach, we performed several tests on an embedded platform. In particular, we used pITX-SP 1.6 plus board manufactured by Kontron[3]. It is equipped with a 1.6 GHz Atom Z530 and 2GB RAM. We use Clingo[4] as ASP solver which is an incremental ASP system implemented on top of clasp[3] and Gringo[3] solvers [20]. Clingo is written in C and runs under Windows and Linux. We measured the execution time of the ASP solver on our embedded platform. The knowledge base consists of 10, 30 and 40 facts and supported by the previous optimization rules as discussed above. The overall execution time was between 0.4 and 0.6 seconds. I.e., that the reasoning system can run on smart phones and support the user in real time. 40 as a maximum number of facts mean that the user could choose between up to 8 operations which are much more than usual in everyday situations.

5 Conclusion and Future Work

As has been shown, HCM-Modeler is a powerful and comprehensive tool for developing, managing and exchanging models written in HCM-L. As HCM-L focuses on behavior and its context, HCM-Modeler might be used for a wide variety of applications in Ambient Assistance, healthcare and other process-oriented domains.

The Answer Set Programing paradigm proved to be an appropriate solution for solving heuristic problems. Furthermore, we showed that ASP allows solving such problems in real time which is important for the given application domain.

Currently we are designing usability experiments with end-users in order to reveal improvement potential for the modeler's interface.

References

1. Steg, H., et al.: Europe Is Facing a Demographic Challenge. In: Ambient Assisted Living Offers Solutions. VDI/VDE/IT, Berlin (2006)
2. Lawton, M.P., Brody, E.M.: Assessment of older people: Self-maintaining and instrumental activities of daily living. Gerontologist 9, 179–186 (1969)

[3] http://www.kontron.com
[4] http://potassco.sourceforge.net

3. Michael, J., Grießer, A., Strobl, T., Mayr, H.C.: Cognitive Modeling and Support for Ambient Assistance. In: Mayr, H.C., Kop, C., Liddle, S., Ginige, A. (eds.) UNISON 2012. LNBIP, vol. 137, pp. 96–107. Springer, Heidelberg (2013)
4. Michael, J., Mayr, H.C.: Conceptual Modeling for Ambient Assistance. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 403–413. Springer, Heidelberg (2013)
5. Karagiannis, D., Grossmann, W., Höfferer, P.: Open Model Initiative: A Feasibility Study. Dpmt. of Knowledge Engineering, University of Vienna (2002), http://www.openmodels.at
6. Karagiannis, D.: Business Process Management: A Holistic Management Approach. In: Mayr, H.C., Kop, C., Liddle, S., Ginige, A. (eds.) UNISON 2012. LNBIP, vol. 137, pp. 1–12. Springer, Heidelberg (2013)
7. Kofod-Petersen, A., Mikalsen, M.: Context: Representation and Reasoning. Special Issue of the Revue d'Intelligence Artificielle on "Applying Context-Management" (2005)
8. Moody, D.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Software Eng. 35, 756–779 (2009)
9. OMG Meta Object Facility (MOF) Core Specification Version 2.4.1: http://www.omg.org/spec/MOF/2.4.1/PDF/ (last visited on September 29, 2013)
10. Schwarz, H., Ebert, J., Lemcke, J., Rahmani, T., Zivkovic, S.: Using expressive traceability relationships for ensuring consistent process model refinement. In: Proc. Int. Conf. on Engineering of Complex Computer Systems (ICECCS), pp. 183–192 (2010)
11. Fill, H.G., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. Enterprise Modelling and Information Systems Architectures 8(1) (2013)
12. ADOxx® Software Engineer Tutorial Package: http://www.adoxx.org/live/tutorial (last visited on September 29, 2013)
13. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003) ISBN 978-0-521-81802-5
14. Turaga, R., et al.: Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology 18(11), 1473–1488 (2003)
15. Batsakis, S., Petrakis, E.G.M.: Sowl: Spatio-temporal representation, reasoning and querying over the semantic web. In: Proc. 6th Int. Conf. on Semantic Systems. ACM (2010)
16. King, B., et al.: Statistical reasoning in the behavioral sciences. Wiley.com (2011)
17. Batsakis, S., Petrakis, E.: SOWL: Spatio-temporal representation, reasoning and querying over the semantic web. In: Proc. 6th Int. Conf. on Semantic Systems. ACM (2010)
18. Sensoy, M., et al.: OWL-polar: A framework for semantic policy representation and reasoning. In: Web Semantics: Science, Services and Agents on the WWW 2012, pp. 148–160 (2012)
19. Gentner, D., et al.: Cognitive Approaches for the Semantic Web. Dagstuhl Reports 2.5, 93–116 (2012)
20. Al Machot, F., et al.: Real time complex event detection for resource-limited multimedia sensor networks. In: Proc. 8th IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (2011)
21. Gebser, M., et al.: A user's guide to gringo, clasp, clingo, and iclingo (2008)