Knowledge representation for ambient security

Lauro Snidaro and Gian Luca Foresti
Department of Mathematics and Computer Science, University of Udine, Italy
E-mail: snidaro@dimi.uniud.it

Abstract: *Ambient intelligence envisages an articulated, though transparent, interaction between the user and the environment. According to this grand vision, appliances and systems embedded in the environment have to react to the user's presence and provide services in a customized fashion. Therefore, ambient intelligence systems should be endowed with context awareness capabilities in order to provide the proper responses for each user. This paper specifically shows how the system can be instructed to recognize events occurring in the observed environment for security purposes.*

Keywords: ontologies, security, situation assessment

1. Introduction

Ambient intelligence (AmI) is foreseen as a revolutionary way in which machines will interact between themselves and with the user. To describe this, enticing scenarios have been developed in which the user wears appliances that are able to communicate with systems and devices present in the environment in order to provide information and receive services (Ducatel *et al.*, 2001).

In recent years, most of the literature has proposed ways to cope with the three main components of AmI: ubiquitous computing, ubiquitous communication and intelligent user interfaces. That is, computing paradigms have been proposed, communications issues have been discussed, and smart interfaces have been designed. Not much attention has been devoted to the core of an AmI system, i.e. the reasoning capabilities that should provide the user with the hassle-free comfort of the right information/service at the right moment in the right place (Abowd & Mynatt, 2000).

To be really effective, this reasoning capability has to take into account a huge amount of information such as available services, preferences and profile of the user, current situation, activity and intentions of the user, and tasks of the system. Therefore, the system, intended as a distributed entity, has to achieve a strong perception and awareness of its own current status, its goals and capabilities. Capabilities are related to the users who can exploit them through services, which in turn have also to be understood by the system.

Ontologies are rapidly becoming the foremost means to express knowledge in a principled way, and only recently they have been considered in the development of context-aware systems (Wang *et al.*, 2004). This paper shows how recent developments in ontologies can shed new light on the difficult task of building intelligent environments.

2. Intelligent environments

According to Mark Weiser's view on ubiquitous computing (Weiser, 1999), endorsed and evolved by the AmI paradigm (Ducatel *et al.*, 2001), intelligent environments should embed electronic devices able to provide customized information and services to the users.

As noted in Abowd and Mynatt (2000), this should not be taken as a mere spatial extension of the access to services like email and personal calendar available nowadays, but a completely new way of perceiving reality. This could mean enhancing human capabilities like memory or maintaining implicit information sharing between groups, the general underlying idea being the untangling of the human from the desktop. This calls for a myriad of electronic equipment, seamlessly interacting with each other, going from wearable and portable devices to intelligent spaces (i.e. intelligent classrooms etc.). This new level of interaction is meant to be almost transparent, if not completely invisible, so that the users can focus on their activities or duties.

To accomplish this, research has still to make significant progress in the fields of natural interfaces, communications and, most of all, context awareness. The remainder of the paper will cover the last two points within an ontology perspective.

3. Ontologies

An ontology is a formulation of the entities that are relevant to a domain as well as the relationships between these entities (Gruber, 1993). It can be used to categorize both physical and non-physical entities: the former include objects and aggregates of objects, and the latter refer to attributes, properties, concepts, relations, to describe temporal processes like events or time spans (Little & Rogova, 2005).

In addition, the following considerations hold for ontologies (Smirnov *et al.*, 2005).

1. Ontologies are believed to be a way to overcome the problem of semantic heterogeneity.
2. Ontologies provide means for describing sensor data, objects, relations and general domain theories in the form of knowledge.
3. They provide reusable knowledge.
4. Knowledge represented by ontologies is shareable and understandable for both humans and computers.

Ontologies originated from philosophy, but they have recently been adopted by the scientific community as a way to model knowledge. The main thrust was perhaps given by the World Wide Web Consortium (W3C) that is now focused on the development of the ‘Semantic Web’ (Berners-Lee *et al.*, 2001). This effort has yielded the Web Ontology Language (OWL) which is currently emerging as a standard, among many others (McGuinness & van Harmelen, 2004). The Semantic Web project aims to develop a semantic layer above the current web in order to allow intelligent searches and advanced services based on inferences and reasoning over meta-data.

Specific tools like Protégé (Noy *et al.*, 2001) are under rapid development and offer a wide range of functionalities, from design of classes and concepts to visualization, querying and inferencing.

Ontologies can holistically be used in designing and developing an AmI system in its entirety. However, in the following only distributed communication and context awareness will be discussed as they are central in the development of a real system.

3.1. Communication

The OWL language was originally devised to have heterogeneous web agents understand web content and interact with each other and with web services. The effort of the W3C resulted in a language structured on XML and based on the Resource Description Framework (RDF) (Beckett, 2004). Note that an ontology language goes beyond the scope of a pure XML schema. While XML can be used to define entities and their attributes or composition, an ontology language is able to describe the meaning of the relationships that may exist between entities. Ontology languages therefore provide means to express knowledge about how the different classes of objects relate to one another (Kokar *et al.*, 2004). The meta-data along with a formal taxonomy provide the means for different intelligent systems to exchange information and perform inferences over the contents.

In the case of an AmI application the contents are given by *a priori* information that constitutes the knowledge base of the system along with incoming contextual information relevant to the situation at hand. A formal ontology can therefore lay a common ground for interoperability between the diverse devices and services that can interact in an intelligent environment (i.e. intelligent building) (Preuveneers *et al.*, 2004). While a number of protocols have been developed to allow communication between distributed and heterogeneous agents (CORBA, COM, SOAP etc.), they do not treat the information being interchanged semantically. They provide only the means to establish a syntactically correct dialogue between the parts, but they do not take meaning into account. In this way, different agents could associate different meanings to the same term. Conversely, different terms could actually refer to the same concept. As will be discussed in the following section, solving this issue is of paramount importance when attempting to assess the current situation based on incoming reports of multiple and heterogeneous sources of information (Ranganathan & Campbell, 2003).

Figure 1 illustrates an ontology-managed information accrual by an agent (or user) who performs the initial request.

The process, inspired by Smirnov *et al.* (2005), goes through the following steps.

1. An agent performs a request that is dispatched through the communication network.
2. The request is processed and relevant knowledge is extracted from the ontology library.
3. Extracted knowledge is an abstract context that needs to be instantiated with actual data to reflect the current situation. Therefore, pertinent information sources are contacted to provide such data.
4. Incoming data are integrated into an operational context that represents the situational picture relevant to the request.
5. Additional information may be obtained from relevant sources to process the request within the operational context.
6. The information produced is supplied to the requesting agent and/or broadcast through the network.

As discussed in the following section, a similar process is used, on a larger scale, to build a global situational picture of the domain.

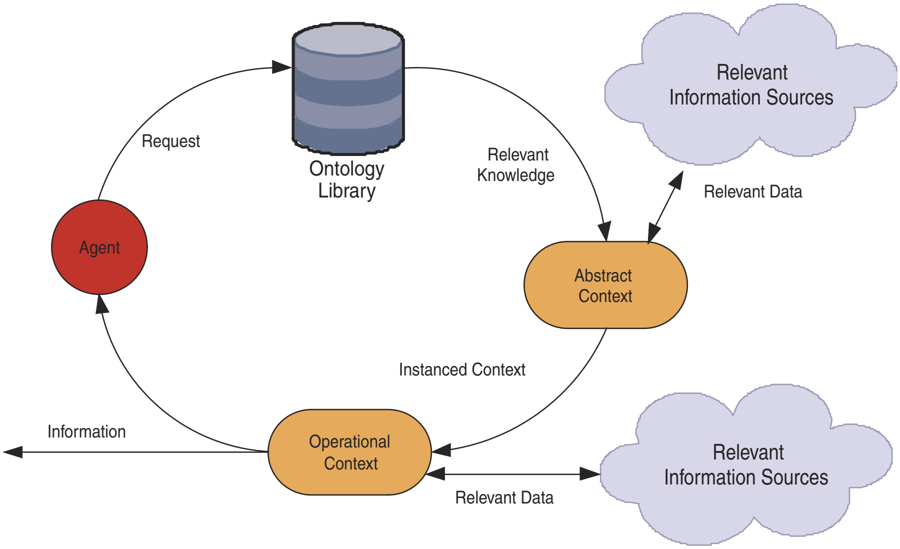

Figure 1: *Ontology-broked information retrieval.*

4. Ontology development

An ontology for a given domain is meant to capture the commonly agreed knowledge about that environment. Generally, ontologies are therefore very application and domain dependent. Nonetheless, a proper hierarchical structuring of the known entities and relationships between them could lead to reusable knowledge across different domains. In particular, as shown in Figure 2, at least two layers of ontologies can be identified for each application: formal ontologies and domain-specific ontologies (Niles & Pease, 2001; Little & Rogova, 2005).

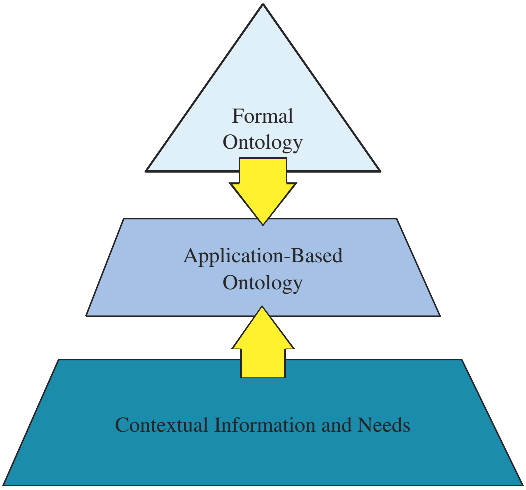

Figure 2: *Formal and domain ontologies (Little & Rogova, 2005). The application-based ontology constrains the formal ontology by incorporating domain-specific knowledge and needs.*

Domain-specific ontologies express the available knowledge on actual entities of a given domain along with information about their properties and relations that may exist between them, whereas formal ontologies define in abstract terms the basic properties of those entities that hold in any case. An application-based ontology can be seen as an instantiation of some of the abstract classes of the formal ontology.

This concept follows the well-known concept of inheritance in object-oriented programming languages, where context-specific classes can be created by deriving and extending library classes.

In the case of an AmI application for intelligent building, for example, the formal ontology could define basic concepts such as *building* and *user*. Figures 3 and 4 show class trees taken from the Suggested Upper Merged Ontology (SUMO) library (Niles & Pease, 2001) and visualized through the Jambalaya Protégé plug-in (Storey et al., 2002). The SUMO library is one of the freely available upper-level ontologies. Figure 3 shows the abstract concepts at the very first two levels of the hierarchy from which every other concept is derived. Figure 4 shows how the 'building' class is derived from the basic concepts.

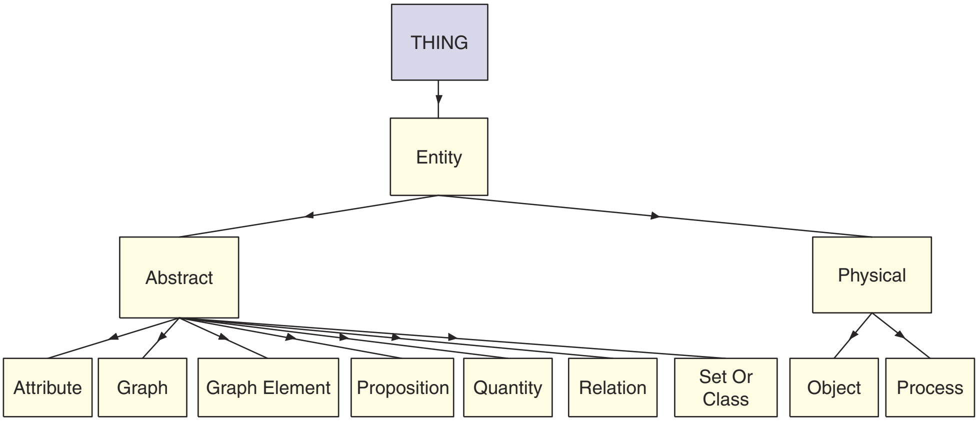

Figure 3: *Abstract classes of the SUMO ontology. Arcs indicate class/subclass relations.*

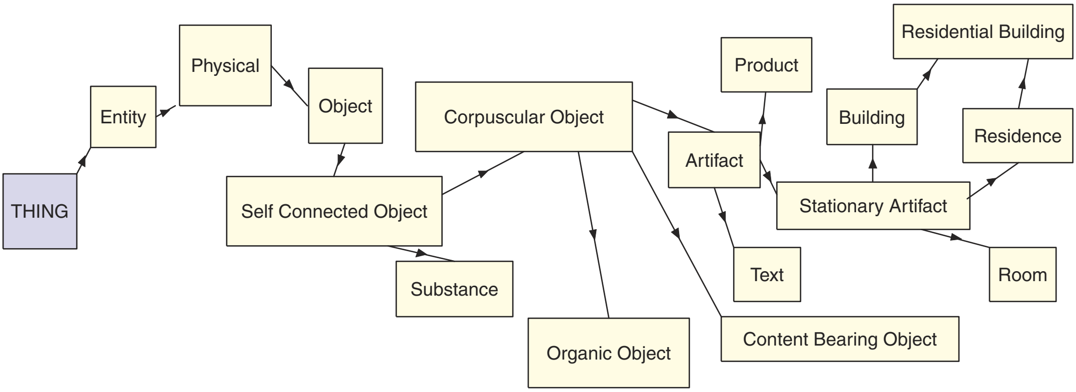

Figure 4: *Tree view of part of the SUMO ontology containing the 'building' class.*

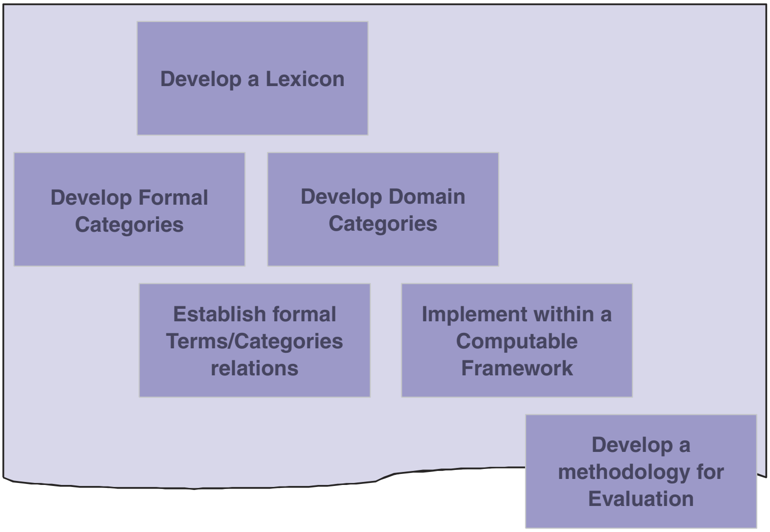

Figure 5: *Methodology for ontology-based application development (Little, 2003).*

The domain-specific ontology specifies and constrains the concepts of the formal ontology to the situation at hand. For example, intelligent buildings should possess computing capabilities and should provide advanced IT services (i.e. automatic and customized access control) along with usual basic functionalities like heating and lighting.

A general methodology for the development of ontology-base applications (Little, 2003) is shown in Figure 5. The first four steps deal with the design phase of the system and have already been discussed. The implementation phase involves the non-trivial choice of a computable framework in which the modelled ontologies could maintain the required expressive power while at the same time allowing reasoning capabilities. This can be achieved by structuring the implementation in several communicating layers. In this case, computational performance could be an issue, though, and the proper architecture should therefore be carefully designed. The last step is meant to assess the completeness and consistency of the ontology.

This is an open problem in the research community as, currently, there is no agreed recipe to address it.

5. Situation awareness

In order to provide adequate and timely services to the users, intelligent environments should be able to develop an understanding of the current situation based on prior knowledge and incoming data. The system should therefore be able to build a dynamic situational picture as a result of reasoning about objects, attributes, aggregates, relationships and their behaviour over time within a specific context (Little & Rogova, 2005). Only recently, ontologies have been considered in developing situation-aware AmI systems (Chen *et al.*, 2003; Ranganathan & Campbell, 2003; Wang *et al.*, 2004). However, those papers do not discuss evaluating the process of building the situation picture for consistency and effectiveness.

Building an accurate situational picture of an intelligent environment is certainly no easy task. In fact, the following characteristics apply to a wide range of domains:

- generally noisy, dynamic, and complex real-world environments,
- numerous and heterogeneous sources of information,
- simultaneous interaction with many users.

At the same time the system has to deal with imposed resource and time constraints.

Composing a consistent and effective picture of the environment status requires the fusion of different pieces of information (Abowd & Mynatt, 2000). This important aspect cannot be overstated and is grounded on the most recent research focus on Level 2 Data Fusion (Llinas *et al.*, 2004).

The process, outlined in Figure 6, involves a continuous consistency check of the picture created so far based on incoming data. If, according to available knowledge in the ontology library, the picture is not deemed sufficiently representative (i.e. due to poor or scarce data), the system tries to collect more information and goes through a refinement step. When the situation has been assessed, the system evaluates whether a reaction has to be provided or not. In any case, thesystem has to take into account if the decision is effective and compliant with pending tasks and goals.

In the context of an intelligent building, for example, this would mean localizing and identifying the persons present in the rooms, understanding their activities, and creating a

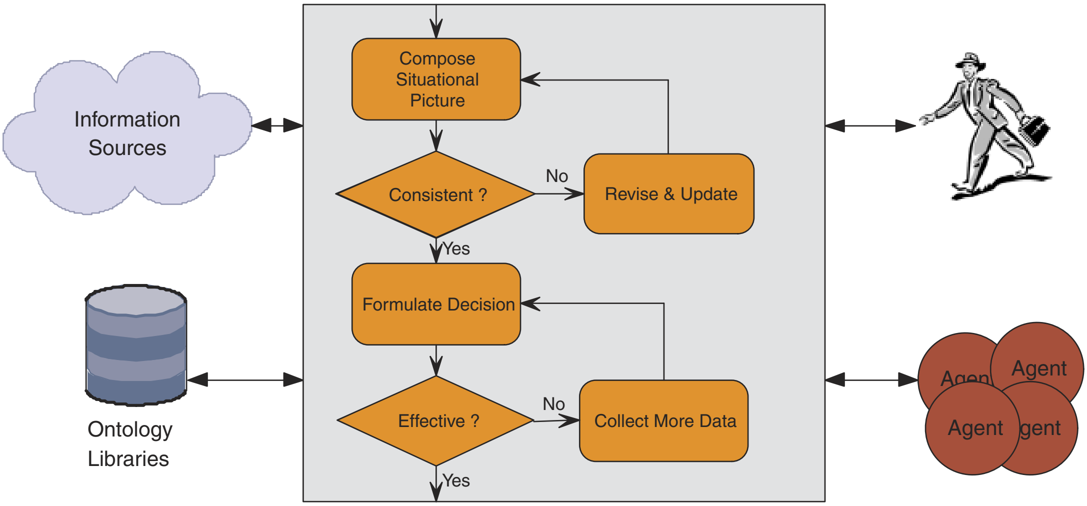

Figure 6: *Situation assessment process.*

situational picture by integrating available knowledge in the ontology library with collected data incoming from heterogeneous sources (i.e. cameras, portable devices, direct interaction of the user etc.). Localization could indeed play a key role in understanding activities and intentions (Snidaro *et al*., 2005); therefore an ontology for an intelligent building should specify the role and rules associated to every area and room. In this way, proximity to a given area could be used to perform basic inferences about intentions.

However, as pointed out in Abowd and Mynatt (2000), position and identity are not sufficiently expressive pieces of information to infer, for example, the intentions of human beings. Along with answering the who, where, when, what questions, the system should be able to understand why a certain entity is performing a certain action in a given place at a given time. To this end, the starting point for this kind of high-level reasoning resides in the capability of the system to detect and recognize events and to understand the relations and implications between them.

5.1. Event representation

Building a situational picture requires the system to be able to assess the current state of the observed environment. More specifically for security purposes, the system should be able to detect and recognize events. These can be subdivided into atomic and complex. The former can be considered as the variation of an entity's state, while the latter are a sequence of atomic activities. In the surveillance domain, several attempts have been made to model events: in Oliver *et al.* (2000) coupled hidden Markov models are used to learn interaction semantics, in Ivanov and Bobick (2000) hidden Markov models are coupled with context-free grammars, and in Hongeng *et al.* (2004) hierarchical decomposition and single/multiple thread terminology, to describe single and concurrent events, are advocated. The Video Event Representation Language (VERL), to explicitly manage the semantics of events and to instruct the system to recognize them, is presented in François *et al.* (2005).

This language represents a convenient way to express the knowledge of the system on relevant activities in the observed environment. Indeed, the system can be explicitly taught to discriminate between normal and anomalous behaviour. Table 1 shows how an 'identity fraud' event can be defined.

In the table the malevolent act of attempting to gain entry into a protected facility through a stolen access card is described. The event takes place when entity x approaches door d of facility f, card c is swiped at d (and c is a valid card to get into f through d), but the biometric control subsystem fails to verify the identity contained in c with the biometric features extracted from entity x.

The VERL language constitutes a potentially interesting choice for event representation; however, there is no working implementation of it. The level of expressiveness of VERL in fact requires the OWL-Full declination of the OWL language which is not computable. A partial implementation exists that defines some of the syntactic elements of VERL in OWL; however, there is no mechanism that defines complex events and no mention of how they should be derived from instances of simple events.

5.2. Rule languages

OWL is an appropriate language for expressing taxonomies in terms of classes, subclasses and relations between them. However, it falls short in describing rules that can be computed by a reasoner to derive knowledge. Rule-based systems are successfully employed across a variety of domains (e.g. economic models, clinical diagnoses, process control) in a multitude of implementations with very limited support for interoperability. The widespread interest in ontologies and means to exchange knowledge bases has recently pushed endeavours toward adoption of a common rule base that could be computed by different rule engines. The effort has led to the development of several rule languages such as Rule Markup Language,

Table 1: *Identity fraud event in VERL*

```
SINGLE-THREAD(Identity-Fraud(ent x, card c, facility f)
  AND(
      Sequence (
      approach (x,d),
      swipe(c,d),
      NOT(verifyIdentity(x,c))),
      portal-of(d,f),
      key-of(c,d)
      ))
```

Semantic Web Rule Language (SWRL), Metalog and ISO Prolog, among others (O'Connor *et al.*, 2005).

The SWRL is intended to be the rule language of the Semantic Web. It is based on OWL DL, OWL Lite and the Rule Markup Language. All rules are expressed in terms of OWL concepts (classes, properties, individuals), and this allows the integration of taxonomic knowledge and rule bases about a given domain under the common hood of OWL. This means that rules can be used to infer new knowledge from existing OWL ontologies. The widespread adoption of OWL is confirmed by the availability of a multitude of editors to help build OWL ontologies. In particular, the already mentioned Protégé editor provides also a 'SWRLTab' to assist the definition of SWRL rules (O'Connor *et al.*, 2005).

The following logic expression is an example of an SWRL translation for the complex event described in Table 1:

$$\begin{aligned} &Person(?x) \wedge Door(?y) \wedge isCloseTo(?x,?y) \wedge \\ &\wedge Card(?z) \wedge isSwipedOn(?z,?y) \wedge \\ &\wedge isOpenedBy(?y,?z) \wedge \\ &\wedge isNotVerifiedBy(?x,?z) \Rightarrow \\ &\Rightarrow isViolating(?x,?y) \end{aligned} \tag{1}$$

Since this expression is eventually translated into OWL by the SWRLTab editor, properties of the involved entities can be seamlessly retrieved through formal or domain ontologies (Section 4). The above rule constitutes a hypothetical proposition where the antecedent is given by the conjunction of several predicates. If all the predicates in the antecedent are verified, i.e. the video-surveillance system reports that a person x is close to a door y, and a magnetic card z is swiped on door y, and z is a valid card for door y, but the identity stored on the card is not matching the one provided by the biometric control system failing to recognize x, then the consequent follows verifying the predicate that the person x is attempting an unauthorized access to door y.

The SWRLTab Protégé plug-in also provides high-level Java application programming interfaces (APIs) called 'SWRL Factory' that allow the creation and modification of SWRL rules programmatically. In this way, external applications can access/create/modify/delete rules in the rule base. However, the APIs do not provide inferencing capabilities. Inferencing is provided by the external reasoner Jess[1] through the 'SWRL Bridge' which is part of the Protégé-OWL APIs. These APIs allow the interaction with the rule engine and can be used to compute inferences.

Note that rules such as (1), which exemplifies the definition of the complex event 'identity fraud' for a surveillance system in an intelligent building, provide an effective way of integrating knowledge coming from heterogeneous sources. In the example, the antecedent is composed of atomic (or simple) events detected by different types of sensors (e.g. surveillance camera, magnetic card reader, biometric system). The recognition of atomic events can be performed per sensor through techniques such as Bayesian

---

[1] *http://www.jessrules.com/*

networks (Hongeng *et al.*, 2004). A number of features can be extracted from the data produced by a sensor (e.g. the position of a person can be extracted from the video stream provided by a surveillance camera) and they can be used to build Bayesian networks for each atomic event of interest. Each network will provide a Boolean output stating if the atomic event has occurred or not. These binary outputs will then be fed to the reasoner to check if any rule in the rule base can be fired.

5.3. Uncertainty

As discussed in the previous section, complex events are defined through logic expressions that can assume only Boolean values. Indeed, the verification of atomic events is expressed by predicates and this could be a limiting condition in many real-world scenarios. Whatever technique is used to recognize atomic events, then its outputs have to be, in most cases, thresholded to yield a Boolean value. Take for example the proximity condition expressed by the predicate *isCloseTo* in (1). To detect the atomic event that a person is close to a given object, the current distance between the person and the object must be compared against a preset threshold. In real-world scenarios, hard thresholds are troublesome in the vast majority of cases. In the example above, rule (1) would not fire if the video-surveillance system had over-estimated the distance between the person and the door yielding a value above the threshold.

It would be better then to assign a degree of truth to expressions such as the antecedent of (1), instead of computing its Boolean value. This can be done by assigning probability coefficients to each term and then evaluating the overall degree of truth of the weighted expression. A fuzzy approach can also be foreseen by exploiting a specific extension of the Jess engine called 'FuzzyJess' (Orchard, 2001) that provides the functionalities needed to handle fuzzy logic.

5.4. Video annotation

An AmI system which uses video sensors to collect information about the environment should maintain this information in a structured and exchangeable way that can be shared throughout the system for distributed processing. To this end, a Video Event Markup Language (VEML) is also described in François *et al.* (2005). However, in our opinion, it is not clear why this solution should be adopted in lieu of the MPEG-7 standard which already has these capabilities and is being internationally accepted. In addition, the VEML is specifically designed for video annotation and it is thus limited to this medium. In contrast, MPEG-7, also called 'Multimedia Content Description Interface', standardizes the description of generic multimedia content supporting a wide range of applications (Manjunath *et al.*, 2002). Descriptions comprise descriptors (D) and description schemes (DS) that can span from low-level audio and video (e.g. scale, timbre, instruments, colour, shape, texture) to high-level content meanings (semantics).

Here, the purpose is to annotate video streams coming from the sensors and locally processed with descriptions of detected events whose semantics have been previously defined in VERL. In the following example, the 'identity fraud' event of Table 1 is detected and recognized in the video stream produced by a surveillance camera (Table 2). Because of lack of space, only the first, namely 'Entity approaching', of the sequence of events leading to an identity fraud is described.

MPEG-7 descriptions can be transmitted in textual format or in binary format. That is, the standard describes how to encode and decode textual XML descriptions into binary format that could be efficiently compressed and streamed over the network. The 'Binary format for MPEG-7' (BiM) was specifically designed to cater for possibly scarce network or storage resources. As already mentioned, this is certainly the case for distributed applications for AmI.

An MPEG-7 description can be represented in a tree structure which has a bijective correspondence with the nested syntax of an XML file. Nodes in the tree represent information, while links stand for the 'containment' relation. Both 'Textual format for MPEG-7' (TeM) and

Table 2: *Example of MPEG-7 annotation*

```
<Mpeg7>
 <Description xsi:type="ContentEntityType">
  <MultimediaContent xsi:type="VideoType">
   <Video>
    <TemporalDecomposition>
     <VideoSegment>
      <MediaTime>
       <MediaTimePoint>T00:25:04:0F25</MediaTimePoint>
      </MediaTime>
      <MovingRegion id="Person_5">
       <!-- moving region description -->
      </MovingRegion>
          <Semantic>
            <SemanticBase xsi:type="AgentObjectType" id="id_1">
            </SemanticBase>
            <SemanticBase xsi:type="EventType" id="id_2">
              <Label>
                <Name>Entity Approaching</Name>
              </Label>
              <Event>
                <Label>
                  <Name>Approaching</Name>
                </Label>
                <SemanticPlace>
                  <Label>
                    <Name>Facility B</Name>
                  </Label>
                </SemanticPlace>
                <SemanticTime>
                 <Time>
                   <TimePoint>Mon Jul 10 07:16:27 CEST 20</TimePoint>
                  </Time>
                 </SemanticTime>
              </Event>
            </SemanticBase>
            <SemanticBase xsi:type="ObjectType" id="id_3">
              <Label>
                <Name>door</Name>
              </Label>
            </SemanticBase>
            <Graph>
              <Relation
         type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
        source="#id_1" target="#id_2" />
              <Relation
               type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:instrumentOf"
              source="#id_3" target="#id_2" />
            </Graph>
          </Semantic>
     </VideoSegment>

     <!-- more video segments... -->

     <VideoSegment>
      <MediaTime>
       <MediaTimePoint>T00:25:14:0F25</MediaTimePoint>
      </MediaTime>
          <Semantic>
            <SemanticBase xsi:type="EventType" id="id_10">
              <Label>
                <Name>Identity fraud</Name>
              </Label>
              <Event>
                <Label>
                  <Name>IdentityFraud</Name>
                </Label>
               </Event>
             </SemanticBase>
           </Semantic>
         <VideoSegment>
    </TemporalDecomposition>
   </Video>
  </MultimediaContent>
 </Description>
</Mpeg7>
```

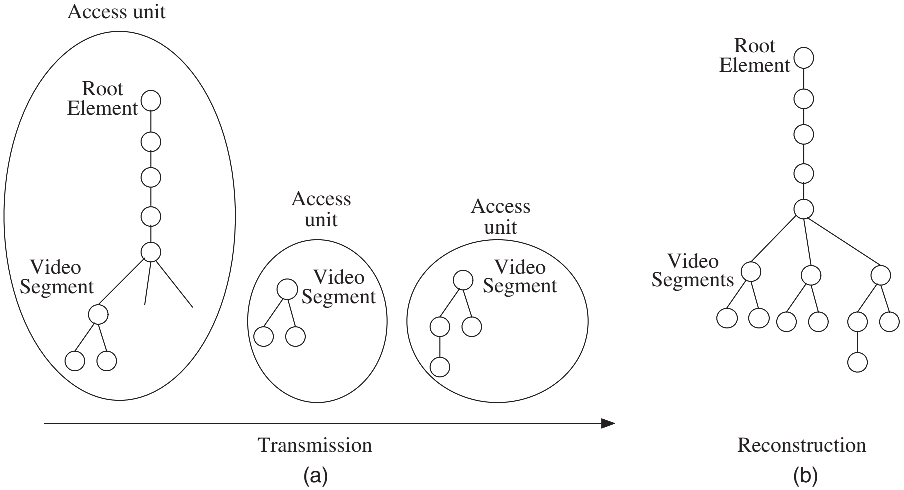

Figure 7: *Description trees and access units. (a) Fragments of a description tree are sent over the network by first-layer nodes. (b) Second-layer nodes compose the fragments into a full description.*

BiM allow dynamic and incremental transmission of description trees. This means that the standard allows the decomposition of a full description into several fragments (sub-trees), to encapsulate the fragments into access units, which can be sent over the network separately, and to reconstruct the full description at the destination as shown in Figure 7. This feature is particularly relevant for a system that elaborates video content in real time, thus building a description as the composition of several temporally separated fragments.

6. Conclusions

In this paper, the need for a formal way to represent knowledge about a domain in an AmI application is discussed. This knowledge, along with reasoning capabilities, represents the basis for building the degree of awareness an AmI system should show with respect to situations, users, services and itself. This issue is here discussed in the context of a video security subsystem for an intelligent building. We show how relevant events in the application's domain can be defined and recognized through SWRL rules. We also show how video streams can be annotated with structured and extensible metadata conforming to the MPEG-7 standard that can be interchanged between the heterogeneous devices constituting an AmI system.

Acknowledgements

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project 'Ambient intelligence: event analysis, sensor reconfiguration and multimodal interfaces' (2007–2008).

References

Abowd, G.D. and E.D. Mynatt (2000) Charting past, present, and future research in ubiquitous computing, *ACM Transactions on Computer–Human Interaction*, **7** (1), 29–58.

Beckett, D. (2004) RDF/XML syntax specification (revised), *Technical Report*, W3C, February.

Berners-Lee, T., J. Hendler and O. Lassila (2001) The Semantic Web, *Scientific American*, **284** (5), 34–43.

Chen, H., T. Finin and A. Joshi (2003) An ontology for context-aware pervasive computing environments, *Knowledge Engineering Review*, **18** (3), 197–207.

Ducatel, K., M. Bogdaniwicz, F. Scapolo, J. Leijten and J.-C. Burgelman (2001) Scenarios for ambient intelligence in 2010, *Technical Report 10*, ISTAG, February.

François, A.R.J., R. Nevatia, J. Hobbs, R.C. Bolles and J.R. Smith (2005) VERL: an ontology framework for representing and annotating video events, *IEEE MultiMedia Magazine*, **12** (4), 76–86.

GRUBER, T.R. (1993) A translation approach to portable ontology specification, *Knowledge Acquisition*, **5**, 199–220.

HONGENG, S., R. NEVATIA and F. BREMOND (2004) Video-based event recognition: activity representation and probabilistic recognition methods, *Computer Vision and Image Understanding*, **96** (2), 129–162.

IVANOV, Y.A. and A.F. BOBICK (2000) Recognition of visual activities and interactions by stochastic parsing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8), 852–872.

KOKAR, M.M., C.J. MATHEUS, K. BACLAWSKI, J.A. LETKOWSKI, M. HINMAN and J. SALERNO (2004) Use cases for ontologies in information fusion, in *Proceedings of the 7th International Conference on Information Fusion*, Philadelphia, PA: International Society of Information Fusion, 415–421.

LITTLE, E. (2003) A proposed methodology for the development of application-based formal ontologies, in *Proceedings of the Workshop on Reference Ontologies and Application Ontologies*, Hamburg, 15–18 September.

LITTLE, E.G. and G.L. ROGOVA (2005) Ontology meta-model for building a situational picture of catastrophic events, in *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia, PA: International Society of Information Fusion.

LLINAS, J., C.L. BOWMAN, G.L. ROGOVA, A.N. STEINBERG, E.L. WALTZ and F.E. WHITE (2004) Revisiting the JDL data fusion model II, in *Proceedings of the 7th International Conference on Information Fusion*, P. Svensson and J. Schubert (eds), Philadelphia, PA: International Society of Information Fusion, Vol. II, pp. 1218–1230.

MANJUNATH, B.S., P. SALEMBIER and T. SIKORA (eds) (2002) *Introduction to MPEG-7: Multimedia Content Description Interface*, New York: Wiley.

MCGUINNESS, D.L. and F. VAN HARMELEN (2004) OWL web ontology language overview, *Technical Report*, World Wide Web Consortium, http://www.w3.org/TR/2004/REC-owl-features-20040210.

NILES, I. and A. PEASE (2001) Towards a standard upper ontology, in *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, C. Welty and B. Smith (eds), 2–9.

NOY, N.F., M. SINTEK, S. DECKER, M. CRUBEZY, R.W. FERGERSON and M.A. MUSEN (2001) Creating Semantic Web contents with Protégé-2000, *IEEE Intelligent Systems*, **16** (2), 60–71.

O'CONNOR, M.J., H. KNUBLAUCH, S.W. TU, B. GROSSOF, M. DEAN, W.E. GROSSO and M.A. MUSEN (2005) Supporting rule system interoperability on the Semantic Web with SWRL, in *Proceedings of the 4th International Semantic Web Conference (ISWC2005)*, 974–986.

OLIVER, N.M., B. ROSARIO and A.P. PENTLAND (2000) A Bayesian computer vision system for modeling human interactions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8), 831–843.

ORCHARD, R. (2001) Fuzzy reasoning in JESS: the fuzzyj toolkit and fuzzy-jess, in *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS 2001)*, 533–542.

PREUVENEERS, D., J. VAN DEN BERGH, D. WAGELAAR, A. GEORGES, P. RIGOLE, T. CLERCKX, Y. BERBERS, K. CONINX, V. JONCKERS and K. DE BOSSCHERE (2004) Towards an extensible context ontology for ambient intelligence, in *Second European Symposium on Ambient Intelligence*, P. Markopoulos, B. Eggen, E. Aarts and J.L. Crowley (eds), LNCS 3295, Berlin: Springer, 148–159.

RANGANATHAN, A. and R.H. CAMPBELL (2003) A middleware for context-aware agents in ubiquitous computing environments, in *International Middleware Conference*, M. Endler and D. Schmidt (eds), LNCS 2672, Berlin: Springer, 143–161.

SMIRNOV, A.V., M.P. PASHKIN, N.G. CHILOV, T.V. LEVASHOVA and A.A. KRIZHANOVSKY (2005) Ontology-driven information integration for operational decision support, in *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia, PA: International Society of Information Fusion.

SNIDARO, L., C. MICHELONI and C. CHIAVEDALE (2005) Video security for ambient intelligence, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, **35** (1), 133–144.

STOREY, M.-A., M. MUSEN, N. NOY, C. BEST, R. FERGERSON and J. SILVA (2002) Visualization support for ontology authoring and knowledge acquisition, *Technical Report*, CHISEL Group, Department of Computer Science, University of Victoria.

WANG, X.H., T. GU, D.Q. ZHANG and H.K. PUNG (2004) Ontology based context modelling and reasoning using OWL, in *PERCOMW '04: Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Washington, DC: IEEE, 18–22.

WEISER, M. (1999) The computer for the 21st century, *ACM SIGMOBILE Mobile Computing and Communications Review*, **3** (3), 3–11.

The authors

Lauro Snidaro

Lauro Snidaro received the Laurea degree and doctoral degree in computer science from the

University of Udine, Italy, in 2002 and 2006 respectively. He is currently a post-doctoral researcher in computer science and a member of the AVIRES Laboratory in the Mathematics and Computer Science Department at the same university. His main interests include data and information fusion, computer vision and artificial neural networks. He actively publishes scientific papers in international journals and refereed international conferences. He has experience in European projects since 2001 and participated directly in the MISS project (FP6-SUSTDEV-516235). He serves as a reviewer for several international journals and conferences.

Gian Luca Foresti

Gian Luca Foresti was born in Savona (Italy) in 1965. He received a Laurea degree *cum laude* in Electronic Engineering and a PhD in Computer Science from the University of Genoa, Italy, in 1990 and in 1994, respectively. In 1994, he was visiting Professor at the University of Trento, Italy. Since 2000, he has been Professor of Computer Science in the Department of Mathematics and Computer Science, University of Udine, where he is also Director of the Artificial Vision and Real-time Systems Laboratory. His main interests involve (a) multisensor data and information fusion, (b) computer vision, (c) image processing and understanding and (d) artificial neural networks and pattern recognition. The techniques he proposed found applications in the following fields: video-based surveillance systems, autonomous vehicle driving, road traffic control, human behaviour understanding and visual inspection. Professor Foresti is author or co-author of more than 200 papers published in international journals and refereed international conferences. He was general chairman and member of the technical committees at several conferences where he has been co-organizer of several special sessions on data fusion, image processing and pattern recognition. He was finance chair of the 2005 IEEE Conference on Image Processing, Genoa, Italy. He has contributed in several books in his area of interest and he is co-author of the books *Multisensor Surveillance Systems: The Fusion Perspective* (Kluwer, 2003) and *Ambient Intelligence: A Novel Paradigm* (Springer, 2005). He has been Guest Editor of a Special Issue of the *IEEE Transactions on Systems, Man and Cybernetics* on ‘Ambient intelligence’ and of a Special Issue of the *Proceedings of the IEEE* on ‘Video communications, processing and understanding for third generation surveillance systems’. He serves as a reviewer for several international journals, and for the European Union in different research programmes. He is a member of IAPR and a senior member of IEEE.