Context-Based Reasoning in Smart Buildings

Pedro Fazenda[1], Paulo Carreira[2], and Pedro Lima[1]

[1] Institute for Systems and Robotics, IST, Portugal
[2] The Data Management and Information Retrieval group, INESC-ID, Portugal

Abstract. Smart buildings integrate various systems to effectively manage resources in a coordinated manner in order to maximize technical performance, operating cost savings and tenant comfort. These buildings are expected to extend beyond simple automation to include advanced user interfaces, and automatic building management capable of interacting in real-time. It is not yet clear, however, how to design and implement applications with the entire building structure, services and processes. We discuss the importance of considering *context* in the operation of smart buildings, and present context-based reasoning as a modeling paradigm to create a general purpose applications.

1 Introduction

Indicators show that there is a high cost-effective potential for energy savings in buildings [1], responsible for approximately 40% of the global energy usage [2]. Smart Buildings (SB) have been waved as a solution to increase energy efficiency in buildings. In contrast to the definition of Artificial Intelligence [3], in buildings the term "smart", synonymous with "intelligent", has a functional definition: "intelligent" is typically associated with the integration and automation of systems and functions which operate in ways that provide a responsive, effective and supportive environment, within which organizations can meet their performance objectives [4].

SBs are supported by a number of technologies, included in the automated building management system (BMS), that aim at the well being of occupants, promoting a comfortable environment while ensuring an efficient use of building resources.

The ideas described for SBs fall into a wider concept defined as Ambient Intelligence (AmI) [5], a term widely used to signify a vision in which environments support the people who inhabit them by incorporating data acquisition, computation, intelligence and behavior to everyday objects in an interconnected and unobtrusive way. One important part of AmI is that environments should be capable of anticipating the needs of its inhabitants and respond in a timely and user-friendly way. Advances in technology are opening doors for entire new concepts and applications and, in the limit, buildings may even be able to recognize and respond to user emotion.

1.1 Building systems

The deployment of AmI in buildings has been hindered, not only by the lack of a well defined and globally accepted standard to interconnect building systems, but also by the absence of a common platform that organizes all these different systems with associated knowledge, control strategies, services, variables, models, *etc.* A SB is a very complex system [6]. It can have multiple spaces, tenants, human-machine interfaces, distributed systems, sensors, and a set of observed variables with a significant size that require controlling and monitoring (*e.g.* temperature and humidity in each room). Many variables and models are correlated (*e.g.* the thermal behavior of adjacent spaces) and may depend on context (*e.g.* the temperature variation inside a room depends on a context defined by a set of variables like door/window open/closed). To make things harder, we have to consider that new components can be added at any time (*e.g.* a new energy meter or meteorological station).

Most software architectures for SBs are programmed in a modular way. This modularity deals with the complexity of the BMS's domain by dividing its operation into a number of interdependent services that are able to control building systems and functions such as: lighting, HVAC, access control, room-operations, floor-operations, *etc.* These modules, responsible for each control logic, are largely deployed in isolation and do not take into account a great deal of contextual information that could be useful for their operation. For example, an elevator group scheduler could balance between energy efficiency and quality of service (associated with the expected waiting times), depending *e.g.*, on a holiday or a normal working day.

In this paper we discusses another type of modularity: the operation of each service depends on a set of active *contexts*. These contexts organize knowledge and the necessary reasoning mechanisms to act on the buildings in order to accomplish greater energy savings than the ones we would accomplish with simple automation rules.

1.2 Context-awareness

The awareness of context about the environment, discussion, or problem in hand, allows many important aspects of human interaction to remain implicit. Contexts act like adjustable filters creating a knowledge frame that enables the correct semantics to be assigned to terms therefore enabling a minimal amount of information exchange towards effective communication. This means defining, at each step, which knowledge pieces must be taken into account explicitly (contextualized knowledge) and which pieces are not directly necessary or already shared (contextual knowledge). Human communication uses linguistic expressions that are rather highly contextualized and many misunderstandings, in human discourse, take place when communicants are not in a common context. A context inherently contains much knowledge about a situation and environment of a problem. For example, an area in a supermarket, where temperature values are abnormally different from the rest of the building, correspond, with high probability, to the cold section. In another type of service building, a similar situation may correspond to a datacenter.

In the next section we present some of the related work on applications for SBs. In section 3 we discuss the organization of knowledge and strategies and in section 4, how context-based reasoning (CxBR) can be used to organize such knowledge. Section 5 clarifies the concept of context and how it can be applied in different building services. Section 6 concludes.

2 Related Work

Creating applications for SBs is a current topic of research. Most approaches have used decentralized control solutions based on multi-agent systems (MAS) see, *e.g.* [7, 8, 9, 10, 11]. Their solutions consist of using collections of software agents that monitor and control different parts, as well as different aspects of the environmental conditions of the building. They operate and manage particular entities in the building, *e.g.*, offices, meeting rooms, corridors or electrical devices. Tianfield [12] presents a study on the MAS approach to large complex systems. Agent systems have been widely accepted as an effective coarse-granularity metaphors for perception, modeling and decision-making, particularly in systems where humans are integrated mostly because system modeling becomes greatly alleviated. Developing the infrastructure of a MAS includes developing an agent platform, the agents, and agent communication language, the agent-task association and the social communication. With and agreement on language and communication, agents can be reused, taking their behavior and functionality to other MAS. In this work we want include context-based reasoning [13, 14, 15] in a multi-agent architecture for SBs. The idea can also be extended to multi-service architectures, where different services, each with their own execution-context, manage particular parts of a BMS much like a MAS architecture. The term emergent is frequently used to describe behaviors that arise from the interaction of subsystems and are not evident from the analysis of each subsystem. We believe that a notion of context can bring a new organization to these systems that can help avoid some of the most common problems like avoiding and detecting emergent behaviors. Consider the following example: an agent, programmed to optimize the use of natural lighting in a room, will open the window blinds and turn of the lights. This action may inadvertently increase the temperature inside the space due to solar gains. The agent that manages the HVAC will notice this increase and will try to cool down the room, thus spending more energy. Without a link between lighting, energy and temperature, two agents designed to save energy by managing each of their isolated domains, may end up spending even more energy, when working together in the MAS. With strategies organized according to context, a user may easily detect the increase in energy spending in the situation where the blinds are open, because this may be explicitly verified within that context.

Even though a lot of research has been conducted within context-aware systems, the core term context is not yet a well defined concept [16]. In a general idea, context is a structure or a frame of reference. It permits to define which knowledge should be considered, what are the conditions of activation and limits of validity and when to use it at a given time. It is what constrains a problem solving without intervening in it explicitly. Brezillon et al. [13] state the lack of consensus on this work and present some of the definitions that are given in the literature. In section 5 we explain and redefine the definition of context given by Gonzalez et al. [17], and extended it to multi-agent/multi-service systems [3].

3 Knowledge and strategies in Buildings

The organization of knowledge (*e.g.* how energy is used in a certain room), and planning strategies based on that knowledge (that fulfil some expectations like *e.g.*, saving energy) is not an easy task. It should be accomplished in a modular way and should be available where it is needed *i.e.*, global knowledge (type of building, season o the year, *etc*), and knowledge associated with events in a certain area (*e.g.* the schedule of a tenant), should be available for decision making in that area. In this organization we have to consider:

- **Pre-acquired knowledge**. Of a static nature associated with a building and its operation that needs to be known before deploying a BMS. This includes knowledge about:
  - architectural aspects like the buildings' location and a plan of its structure including doors, rooms, materials, glazing, furniture, electrical layout, pipes, *etc*;
  - building systems (with information on service providers) such as elevators, HVAC (including subsystems, ducts and vents), power storage and generation, sensors and actuators, *etc*;
  - the building's function (supermarket, pool, school, *etc*) and associated information like schedules (*e.g.* holidays, working days), description of spaces (amphitheater, classroom, kitchen, *etc*), and other information like: a company or a department occupies a specific part of the building;
  - occupant's activities, and the association of these activities with specific spaces inside the building (sleeping, working, eating, entertaining, *etc*);
  - electric and gas utility rates.
- **Acquired knowledge**. Accomplished through a process of gathering information from the environment to improve the efficiency of a system in achieving certain goals. This includes creating models that can be used to predict and anticipate the behaviour of tenants and explain variables like indoor temperature, power, lighting, humidity, thermal-behavior of spaces, *etc*. There are many types of algorithms and techniques that can be used for this purpose. The learning process is performed throughout the operation

[3] Throughout this paper we will use the term multi-service.

of the building, with the models being continuously adapted and fitted to the observations. A well-defined organization of knowledge must take into consideration the context (*e.g.* holidays, working-days, winter, summer) that help explain these variables (*e.g.* the total amount of energy used over those periods).

– **Operation strategies** (including optimization). Technical difficulties in creating SBs also include the fact that the set of all possible behaviors, given all possible inputs, is significantly large. It can also be from dealing with several different types of data (discrete/real valued, complex-structured, states, transitions, *etc*) and multiple goals (*e.g.* energy efficiency and comfort) depending on context (working hours, holidays, emergency, *etc*). Operation strategies can also be partitioned into a hierarchy of levels and contexts. For example, at the highest operation level of a BMS, a building manager can be informed that energy is being lost because the building is not sufficiently airtight (with detailed information); or some operational parameters of a chiller can be adjusted. At a lower (or local) level, a window can be closed because the HVAC is *on*. Some local decisions may depend on higher level strategies: *e.g.* a smart thermostat in a room will not turn the cooling/heating *on/off*, if the HVAC system is powered down, after a certain hour, in certain weather conditions.

To avoid ending up with a data rich but information limited environment, conceptual modeling of information must be part of the engineering process, to describe the general knowledge of each domain (HVAC, elevator, room, company located on the 5th floor, *etc*). Conceptual models serve to organize information in a way that can also help e.g., system operators understand the full context of some type of event that is occurring in some part of a network or process. This organization is necessary to support the ability to provide the right information at the right moment to the right decision maker. For example, if something is wrong with the HVAC system, then a message can be sent to an entity responsible for managing this system with detailed information. High-level contextualized information services are often needed along with supportive sensor data or trends to provide context e.g., a malfunction X in the HVAC happened due to a situation Y, as shown by some sensor values Z. The goal is also to facilitate data mining, information publishing, and the application of automatic learning and decision support tools to facilitate system management. For example, a room management service can learn that energy is being wasted when a window opened, while the HVAC is *on*. If such a situation happens frequently, the service may point out that fact by emailing the tenant with detailed information about how much energy is being wasted. At the building level, a building manager can be informed on how much energy is lost in the entire building due to to opened windows, including the corresponding economical costs.

4 Context-Based in Smart Buildings

The concept of context can provide a model to partition the operation of a complex system into "scenarios", where knowledge, strategies, parameters and objectives, are organized. To clarify the concept, lets consider the use of context in the following applications:

- **Problem diagnosis**. In problem diagnosis, context can be used, for example, to reduce the search-space when trying to detect the source of an identified problem. Gonzalez et al. [17] give the following example: a dead battery in a car that has been parked overnight has entirely different diagnostic implications than one that discharges while the car is in operation. This idea can be generalized to buildings. If a certain condition is being verified like *e.g.*, an unusual amount of energy is being used in a certain area, understanding the context in which this condition happens can be fundamental to identify the problem and take corrective measures.
- **Comparing performance**. Taking context into consideration can be very important when comparing entities according to certain performance metrics. In buildings, for instance, when comparing and analyzing the performance in terms of energy use between two different schools, a lot can be gained if context is taken into consideration. Facts like: level of education (primary/secondary school, university), type of school (*e.g.*, economics, dance, military), division of an academic year, *etc*, are important to extract more reliable conclusions.
- **Organizing knowledge**. Previous known expert knowledge about the operation of a particular building can be encoded in a context-based model. Context can be used to explain observed variables and organize models that predict the behavior of those variables. For example in a school, the energy used may depend on the division of the academic year (Christmas break, vacations, exams, holidays, exams, instructional days, *etc*); on the season, location, and other facts that can be previously known. Creating models within each specific context (from, *e.g.*, a time series obtained form an energy meter) can gain a lot from these divisions by minimizing the need of explanatory variables. This is a natural way of including previous known knowledge in the process of modeling variables from the observed environment, creating more reliable models. Following an hierarchy of contexts (*e.g.* building-operation, floor-operation, room-operation), information can also be organized according to locality and resolution (energy used by the entire building, floor or room).
- **Organizing strategies and behaviors**. Multi-context systems support the development of modular architectures. Following some of the arguments used for organizing knowledge, strategies and behaviors can also be organized according to context. Contextual information can help an agent focus attention on appropriate goals to achieve in certain situations. For example, at night a building strategy can be storing thermal energy and shifting energy demand to off-peak time periods, when utility rates are lower; in a

normal working hour, a room-behavior can be regulating natural light with shading devices; in the advent of an emergency situation like, *e.g.*, a fire, the building will assume a totally different set of behaviors and objectives.

- **Sensing and Perception**. To understand how context is important for this item, consider the example on how humans focus their attention. A magician or a pickpocket can take a wallet/watch away from the person's pocket/hand by manipulating this focus and attention. By showing something interesting with one hand, or by pushing the person, they can avoid being detected by distracting the person's attention away from the item that they want to obtain. People sense the environment depending on the surrounding context - giving more attention to certain details and relaxing on others. In buildings, we can imagine a situation like, for example, a fire, where all the focus of sensing is towards satisfying objectives within that context (*e.g.* check if there are locked spaces with people inside and notify the fireman of this situation).
- **Human-machine interfaces**. When considering, for example, the ability to recognize human emotions. This user-centric contextualized information can be used for decision support: *e.g.*, if at a certain moment the user is angry and stressed, then he is probably not very receptive to any notifications about efficiency performance inside the building. The concept is called Affective computing and it concerns enabling systems recognize human emotion and act accordingly. Emotionally intelligent buildings may have a clear advantage when it comes to human-computer interaction.

5 Context-based Reasoning

Part of the architectural design of a building service (*e.g.*, a service that manages the operation of a room by controlling the HVAC and lighting) is designing the CxBR model, *i.e.*, identifying the context set(s), transition rules, dependencies and relations between contexts. The classical frame problem [18] is closely related to this issue. The design process has to include the experience of human experts to model the necessary knowledge associated with the operation of particular types of buildings, equipments, systems, *etc.* Context-encapsulated knowledge appears as a chunk of reasoning that can be re-used in several designs and implementations. A context is a 3-tuple (Ak, Tk, Dk) composed of the following elements:

- Ak **- Action knowledge**. Required for the agent to carry out the behavior encapsulated within the context. It represents the agent's functional intelligence within its given environment for a specific situation. This knowledge can be previously coded with logic rules, or learned using reinforcement learning, neural networks, evolutionary algorithms, *etc.*
- Tk **- Transitional knowledge**. That indicates when a transition to another context is warranted. It can be expressed as IF(**conditions**) then(**activation**) transition rules or any other type of triggering mechanism using, *e.g.*, neural networks.

– *Dk* **- Declarative knowledge**: Describing some aspects of the context. For buildings, this can be used, *e.g.*, to include some of the pre-acquired knowledge, suited for the context.

5.1 Context Hierarchy

A CxBR model can include a context hierarchy as shown in Figure 1. The model can be used to partition knowledge into sub-levels, making it available in the context where it "makes sense". A multi-level hierarchy represents a *vertical* relationship between groups in a set $G = \{g_1, \ldots, g_n\}$. A group $g_i \in G$ contains a set of mutually exclusive contexts $C_i = \{c_0^i, \ldots, c_n^i\}$ and, an active context c_a^i in g_i, is active within the context of its parents *i.e.* it will inherit active, transitional and declarative knowledge from selected contexts in groups that are hierarchically above g_i. c_a^i can redefine or specialize behaviors and/or contain the functionality required to perform specific sub-tasks.

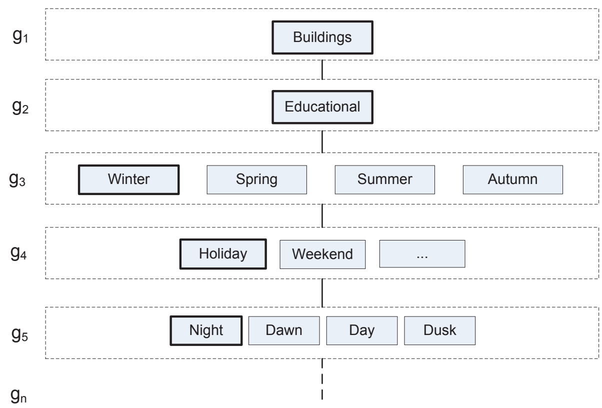

Figure 1: Part of a context hierarchy for an educational building. Highlights show an example of the active contexts at a certain time instance.

5.2 Operational Semantics

Exercising the CxBR model is the process of activating the set of contexts that best suits the situation in hand. This activation allows the active contexts to take over and control the execution a process, defining behaviors, constraints, and other context-dependent characteristics. The process must survey the environment as well as its internal state (including transition knowledge) to determine the conditions where the current context is deactivated and a new context is activated. In Figure 1, if context "Night" is activated then, following an hypothetical scenario, contexts "Holiday", "Winter", "Educational" and "Building" are also activated *i.e.*, the entire path up to the root of the hierarchy tree. A context can override behaviors, add behaviors, redefine attribute values and add knowledge to what it inherits from its parent contexts. Activating the correct context within some processes can be a hard problem. A process that manages the operation of an office room, *e.g.*, may be directly associated with an observable or partially observable state composed by the set of variables that are important for the operation of that room: (door/window opened/closed, temperature, humidity, ocuppied/empty, *etc*). The temperature inside the room behaves differently if a door/window is opend/closed or if the room is empty or occupied. In such a situation, context can be defined *e.g.*, by a set of explanatory variables that can somehow be used to explain or to predict changes in the values of other variables of the state.

G exists within the domain of a service s. At certain instance t, there is a set $C^t_{active} = (c^1_a, \ldots, c^n_a)$ that contains all the active contexts that exist in G. This set is continuously updated, as the following example shows:

$$C^t_{active} = \{Buildings, Educational, Spring, Holiday, Dusk\}$$
$$\downarrow$$
$$C^{t+1}_{active} = \{Buildings, Educational, Spring, WorkingDay, Night\}$$

Service s_i has its own execution thread(s) and its control is a function of C^t_{active}:

$$\texttt{Control of } s_i = \Gamma(C^t_{active})$$

where Γ is the CxBR framework operating within s_i. Figure 2 shows a representation of the framework, including inputs and outputs.

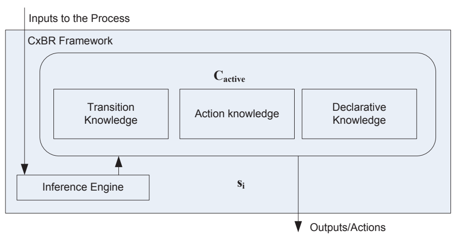

Figure 2: CxBR framework operating within a service s_i.

Distributed applications for a BMS can be composed by multi-distributed context-aware services. The interaction/inter-dependency between these services can be represented by a directed graph. The elements of the graph belong to the set of services $S = \{s_1, \ldots, s_n\}$ that operate with the BMS and the edges represent some type of context or action dependency. Figure 3 shows an example that includes services to manage a building-central (*e.g.*, one that contains the set of contexts represented in Figure 1), a floor, a department and two rooms.

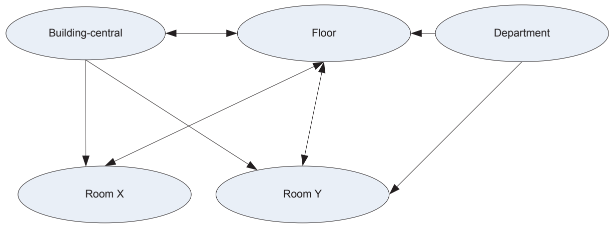

Figure 3: A service dependency graph.

Most actions assumed at the highest level (in the graph, probably the most connected vertex) affect the operation of all services: if the HVAC is turned *off*, then there can be no room-level HVAC strategies in operation within any other service. Most information and knowledge that exists within this service, can also used by several others: season of the year, building characteristics, *etc*.

Behaviors of a room-service can depend on a floor-level strategy or on other information like *e.g.*, information specific to a certain department of a company that is located at that building. For example, it may make sense to turn the HVAC *off* if a department meeting is scheduled to happen on another room. The operation of a floor-service can depend on the current context of each room on that floor. To model, *e.g.*, the thermal-behavior of all spaces, within that floor, it will need to know if windows or doors are opened/closed and the temperature/pressure difference between those spaces.

6 Conclusions and future work

We need the necessary foundations to acquire and organize knowledge and create the necessary reasoning mechanisms to act on the building and accomplish greater energy savings than the ones we could accomplish with simple automation rules. A building is a large complex system and there has been no common platform that organizes all these different systems with associated knowledge, control strategies, services, information, variables, models, *etc*.

In the last few years frameworks like the Robot Operating System [4] have been introduced to the robotics community as a common development platform for robots that provides hardware abstraction, low-level device control, implementation of commonly-used functionality message-passing between processes, *etc*. A similar platform is necessary for smart buildings. Such a software framework, for smart building software development, would enable programmers to reuse drivers and create optimization algorithms with an abstraction over the underlying hardware. We need a framework that is specific for buildings (that can use infrastructure/communication protocols like BACnet, Zigbee, *etc*) and to create such a platform, we have to know how to cope with the dimension of the system and consider the heterogeneity and complexity of a building environment.

[4] http://www.ros.org/wiki/

In this paper we discussed the importance of using a context-based architecture to support some of the aforementioned requirements that are necessary to create smart buildings. We proposed a modeling paradigm that needs to be elaborated and tested. Our vision includes working on a framework similar to the robot operating system, but for buildings. A clear strategy on how to structure such a operating system to fit a building environment and building management requirements is needed. We believe that this vision of creating a building operation system has a lot to gain with previous work on software architectures for context-aware applications.

Acknowledgements

This material is based on work supported under a Portuguese National Science and Technology Foundation Graduate Research Fellowship, by FCT grant number SFRH/BD/60481/2009. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science and Technology Foundation, or the Portuguese government.

We would like to thank all the reviewers for their inputs.

References

[1] Rod Janssen. Towards energy efficient buildings in europe. Technical report, Final Report, EuroAce - The European Alliance of Companies for Energy Efficiency in Buildings, June 2004.

[2] Abdeen Mustafa Omer. Energy, environment and sustainable development. *Renewable and sustainable energy reviews*, pages 2265–2300, 2008.

[3] Peter Norvig Stuart Russell. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, ISBN-10: 0137903952, ISBN-13: 978-0137903955, 2002.

[4] Clements-Croome T.D.J. What do we mean by intelligent buildings? *Automation in Construction*, 6(5):395–400, 1997.

[5] Fariba Sadri. Ambient intelligence: A survey. *ACM Comput. Surv.*, 43(4):36:1–36:66, October 2011.

[6] J. Ottino J. Guckenheimer. Foundations for complex systems research in the physical sciences and engineering. Technical report, Report from an NSF Workshop, September, 2008.

[7] Diane J. Cook, Michael Youngblood, III Edwin O. Heierman, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. Mavhome: An agent-based smart home. *Pervasive Computing and Communications, IEEE International Conference on*, 0:521, 2003.

[8] Magnus Boman, Paul Davidsson, Nikolaos Skarmeas, Keith Clark, and Rune Gustavsson. Energy saving and added customer value in intelligent buildings, 1998.

[9] Paul Davidsson and Magnus Boman. Saving energy and providing value added services in intelligent buildings: A mas approach. In *Agent Systems, Mobile Agents, and Applications, LNCS*, pages 166–177. Springer-Verlag, 2000.

[10] Bing Qiao, Kecheng Liu, and Chris Guy. A multi-agent system for building control. In *Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, IAT '06, pages 653–659, Washington, DC, USA, 2006. IEEE Computer Society.

[11] Alain Schfer Ueli Rutishauser. Adaptive building automation - a multi-agent approach, 2002.

[12] Huaglory Tianfield. A study on the multi-agent approach to large complex systems. In Vasile Palade, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2773 of *Lecture Notes in Computer Science*, pages 438–444. Springer-Verlag.

[13] Patrick Brézillon. Context in problem solving: a survey. *Knowl. Eng. Rev.*, 14(1):47–80, May 1999.

[14] Richmond H Thomason. Representing and reasoning with context. In *Artificial Intelligence and Symbolic Computation Proceedings of the International Conference AISC98*, Plattsburgh New York, 1998.

[15] John McCarthy and Sasa Buvac. Formalizing context (expanded notes). Technical report, Stanford, CA, USA, 1994.

[16] Marius Mikalsen and Anders Kofod-Petersen. Representing and reasoning about context in a mobile environment. *REVUE D'INTELLIGENCE ARTIFICIELLE (RIA*, 19:479–498, 2005.

[17] Avelino J. Gonzalez, Brian S. Stensrud, and Gilbert Barrett. Formalizing context-based reasoning: A modeling paradigm for representing tactical human behavior. *Int. J. Intell. Syst.*, 23(7):822–847, July 2008.

[18] John Mccarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463–502. Edinburgh University Press, 1969.