

Towards interactive smart spaces

Ekaterina Gilman^{a,*}, Oleg Davidyuk^{a,b}, Xiang Su^a and Jukka Riekkii^a

^a *Department of Computer Science and Engineering and Infotech Oulu, University of Oulu, P.O.Box 4500, University of Oulu, 90014, Oulu, Finland*

^b *ARLES project-team, INRIA Paris-Rocquencourt, B.P. 105, 78153, Le Chesnay, France*

Abstract. Recently, we have been witnessing how various social applications and networking services are being integrated more deeply into our daily lives. Until now, social interaction has been attributed exclusively to humans, while resources and the smart space have supported interaction as passive mediators only. However, the involvement of smart spaces as an active actor in the interaction process facilitates more flexible and user-centered applications for users. This article explores how knowledge-based technologies enable smart spaces to actively take part in the interaction. We argue that smart spaces should be able not only to adapt their behaviour according to the actions of humans and other participants, but also initiate interaction when it is necessary. In order to support this statement, we categorise the types of interaction from the participants' perspective, and review and evaluate the technologies enabling interaction in smart spaces. Furthermore, we present our constructive research on interaction in smart spaces: proof-of-concept prototype applications realizing different architectures and supporting various types of interaction in smart spaces.

Keywords: Social interaction, smart spaces, context-awareness, knowledge-based systems

1. Introduction

Social interaction is vital for humans, for their well-being, productivity, and health [10]. Computer technologies support our social interaction by offering hardware infrastructure and social applications. Mobile phones, tablets, and other mobile devices have rich communication facilities and allow users to be continuously available and use different social services like social networks (Facebook), geospatial applications (Foursquare), and games (Gbanga), to name only a few. This kind of mobile software has been coined as Mobile Social Software (MoSoSo) and defined as software supporting social interaction among interconnected individuals [27]. However, MoSoSo does not achieve the full involvement of the surrounding physical space as an active actor in the social interaction, even though this would allow more flexible and user-centered services for users.

In order to involve the surrounding physical space in social interaction, the space should possess the same

kind of interaction capabilities as humans have. That is, changing a space's role from a mediator to an active actor requires perception, cognitive, and acting capabilities. Research on smart environments provides a good basis for enriching physical spaces with computational facilities in such a way that the space is able to acquire information about the events in the space, define meanings for the events, and react accordingly [20,29]. Such environments are called smart spaces. Does this mean that smart spaces can take part in social interaction, support and affect it? Can smart spaces become participants in social interaction and to which extent? We raise these questions with this article.

Traditionally, social interaction is considered from humans' perspective, meaning that social interaction happens only when human beings are involved. However, there are many other interesting forms of interaction in smart spaces. In this article, we define *smart space interaction* as the sequence of bidirectional interactions between actors (or a group of actors) which take into account context, including the actions and behaviour of these actors. Actors in such interaction are independent entities, able to sense and receive infor-

*Corresponding author. E-mail: ekaterina.gilman@ee.oulu.fi.

mation, act on their own, and communicate with each other through the environment (e.g. air, wireless networks, etc.). Moreover, we emphasize an *active role* in such interaction. This means that actors can initiate the interaction towards others. Here, the interacting actors can be humans or computers. One actor can be human and the other the computer, resembling human-computer interaction (HCI). However, smart space interaction is more than HCI as a computer is an active actor in the interaction instead of a mediator. Both actors can even be computers. In such a case, the main difference to plain machine-to-machine (M2M) communication is that the actors are considered as independent entities, able to make decisions on their own.

Generally, interacting actors perform the following actions: observing the overall interaction environment, other participants, and the situation; interpreting the meanings of these observations; reasoning about them; and performing corresponding actions. Behaviour of interacting peers is affected by others. Moreover, this interaction does not progress in isolation: the environmental properties, overall situation, and personal and social factors (social roles, for example) of participants affect it. We consider a smart space as an *active interaction actor*. This means that a smart space can initiate the interaction and humans can perform interaction with the smart space (i.e. humans perceive the social interaction to occur with the smart space).

We define the following key properties for interaction in the smart space: 1) interaction in the smart space is bidirectional interaction, which means that it happens towards both participants; 2) participants adapt their behaviour to the actions of other participants and the situation context (i.e. all information relevant to interaction). This situation context consists of several elements: the physical context, which describes the environment where the interaction occurs; the computational context, describing the computational facilities of the environment; the personal context depicting the information about peer, including own knowledge and experience (i.e. memory); and the social context, which describes the relationship properties the peer might have with other participants and social laws and norms. The social context is the ultimate feature that distinguishes social interaction from other types of interaction.

Semantics is very important for interaction in smart spaces. In other words, interaction in smart spaces is based on giving a meaning to situations and acting according to this meaning. This process of capturing the overall situation context, interpreting it and acting ac-

ordingly is natural for humans, but how can computers (as active interaction actors) do this?

Knowledge-based systems [21] offer the necessary means to support these tasks. Knowledge-based systems provide 1) means to operate with all different objects of the system with unified semantics; 2) flexible solutions to define actions to take according to the situation; 3) inference algorithms to make decisions efficiently.

In this article, we explore how knowledge-based technologies support building smart spaces that are active actors in the interaction. Our hypothesis is that *knowledge-based technologies supported by distributed mechanisms provide an efficient and flexible solution for facilitating smart space interaction; specifically to achieve an active role for the smart space in such interaction.*

Following sections are drawn towards verifying the outlined hypothesis by utilizing a constructive research approach [28]. Based on theoretical studies, the smart space interaction concepts are introduced, followed by verification through prototypes and their evaluation.

The contribution of the article is the following: First, we distinguish the types of interaction in smart spaces from participators' point of view. Second, we explore the technologies necessary to facilitate and support the interaction in smart spaces and with smart spaces. Third, we present our work towards knowledge-based technologies supporting smart space interaction.

The rest of the article is organised as follows: Section 2 outlines related work. We discuss the basics of smart space interaction in Section 3. Section 4 explains the connection between context-awareness and smart space interaction. Section 5 presents the technologies required to achieve interaction in smart spaces. An overview and an analysis of our prototypes are presented in Section 6. Finally, we discuss the findings in Section 7.

2. Related work

Considerable research and development effort has been made to support user interaction. Plenty of technologies and framework solutions can be used to simplify the development of such systems. In this literature review, we explore prominent approaches, technologies, and applications for supporting interaction in smart spaces.

Applications supporting social interaction nowadays attract significant commercial and research inter-

est. Social networks (Facebook), virtual worlds (Second Life¹), etc. allow people to be in contact and share their social context. These applications provide mediated social interaction among users. With recent advances in mobile technologies, mobile devices are increasingly becoming associated with their owners. This is because of their constant availability for users, connectivity and computational facilities. Hence, mobile devices are an attractive platform for development of social and smart space applications.

Equipping mobile devices with sensing technologies enhances context-based social interaction. For instance, location-based social applications utilize location context obtained from GPS sensors (e.g. Foursquare²). Proximity of users can be sensed with Bluetooth [16], the activity context can be recognized from the accelerometer sensor, etc. Moreover, by monitoring and analyzing sensor data, mobile phones allow exploration of the real social relations users have, instead of relations reported by the users [17]. Obviously, social mobile applications would benefit from a platform which allows utilizing the mobile phone context [37]. Also Toninelli et al. [57] suggest such middleware to manage mobile social ecosystems.

Besides providing context information, embedded sensors offer another usage scenario. They can be used as components of multimodal interfaces that enable users to interact with applications using, for example, gestures or voice commands. These interfaces are used both to interact with a mobile phone (e.g. Apple Siri³) and even use the mobile phone as a tool to interact with other objects in the environment [34]. NFC⁴ opens up facilities to connect physical everyday objects lacking computational facilities to the digital world. For instance, Broll et al. [8] demonstrate physical mobile interaction with tagged objects on a wall poster by touching them with a mobile phone equipped with the NFC reader. Multimodal interaction is also utilized in game industry (e.g. Kinect⁵).

Smart spaces exploit sensor technologies and infrastructure to support users in their daily activities; however, supporting social interaction has not yet been studied much. Lifton et al. [26] study the bridging of

physical and virtual environments via embedded and actuator networks. Philipose et al. [33] infer users' activities from the objects they interact with by utilizing RFID technology to track user interactions with objects. Wang et al. [63] recognize multi-user activities using a multimodal, wearable sensor platform to capture observations of both users and their interactions. Coupled Hidden Markov Models (CHMMs) are used to model user interactions and recognize multi-user activities. Cook et al. [11] study the detection and understanding of social interaction in smart spaces by collecting data from multiple sensors of the environment, such as motion sensors, light and temperature sensors, contact switches to monitor the usage of the main key items, such as a cooking pot, etc. Oliver et al. [31] present a real-time computer vision and machine learning system for modelling and recognizing human behaviour in a visual surveillance task. They employ HMMs and CHMMs for modelling behaviour and interactions. Chen et al. [9] also use a camera-based system to discover social interaction in real work environments by tracking the location of people and their head positions in the office.

Another important question is how the interaction can be facilitated. Different types of interaction have different challenges. For instance, embedding interaction facilities in physical artefacts rises many challenges, like the invisibility dilemma, which describes the difficulty to announce to users the interaction facilities of artefacts (so that the user can interact with the object) while preserving the artefacts' functionality and outlook as much as possible [25]. Riekkilä et al. [41] and Hang et al. [19] employ visual design to advertise the services related to physical objects and to tell users that these physical objects can be interacted with in a non-ordinary way. Eagle and Pentland [16] present their research for facilitating face-to-face human interaction. Their Social Serendipity application notifies a user when a person nearby has a similar profile. The authors pointed out the need to consider privacy preserving protocols when developing such systems. Also, interesting interaction patterns are suggested based on application, e.g. buses could wait for passengers on a late connection. Beach et al. [4] suggest a similar system. When considering the interaction between objects, most research works concentrate on agent programming and P2P networks. Agents often embody all steps within an interaction loop, such as sensing, analyzing and reacting. Sayouti et al. [46] propose separation of the interaction-related behaviours and functionalities from the algorithmic parts of the

¹<http://secondlife.com/> (accessed June 2012)

²<https://foursquare.com/> (accessed June 2012)

³<http://www.apple.com/iphone/features/siri.html> (accessed June 2012)

⁴NFC is a short-range wireless communication technology which allows reading passive RFID tags and exchanging information between two NFC-enabled phones.

⁵<http://www.xbox.com/en-GB/Kinect> (accessed June 2012)

agents. For instance, such a technique offers means of defining the properties of interactions, such as mutual exclusion (two interactions are not active at the same time in the system).

Knowledge-based systems are known for their advantages when constructing systems adapting to a changing context. They provide key technologies for context modelling and reasoning. However, their application in the social interaction domain is still rare. One of the reasons for this is the computational resource limitations of the interacting devices, which require all context-management processes to be centralized on the computationally rich devices (servers). Nowadays, pads and smart phones are capable of performing resource-demanding knowledge processing tasks, such as context retrieving and reasoning.

Most research focuses on producing knowledge from the users' social context which is obtained from social networks. For instance, Stan et al. [53] create semantic User Interaction Profiles (UIPs) from the content users share in social platforms in order to recommend similar persons. As another modelling example, Xu et al. [66] propose an ontology-based approach for a mobile personalized recommender system, where the user's historical data are dynamically collected and integrated into the domain ontology, which presents the user profile and is used to calculate the set of recommended items. Kabir et al. [23] explore modelling and coordinating social interactions in pervasive environments. Their approach is to model and represent social interactions and coordinate them to resolve task conflicts during multiple concurrent interactions. An interaction is modelled from both the domain and player (actor) perspectives using the social context. Domain modelling handles the interactions with a particular domain or the environment. Player modelling is conducted to capture the interactions of the individuals considering preferences and context. A fuzzy-logic reasoning mechanism was utilized in order to infer the overall importance of the interaction, assisting the actor to resolve conflicts and make decisions. Truong et al. [59] utilize Semantic Web technologies to model the social context.

As can be seen, the reviewed related work is mainly biased to exploring social platform-based services and applications. With recent technological development, it is now possible to equip wearable and surrounding objects with computational facilities and gather plenty of context information, even social context. Hence, many works concentrate on recognizing human-social activities, which is crucial in order to support it. Few

works consider objects as active participants in interaction, even though there is plenty of research on agent programming and P2P networks. This is understandable, as the interaction model was mostly centralized, and surrounding objects were not able to do any intelligent information processing by themselves, hence there was no real interaction. The situation is changing now, since many digital devices (e.g. recent smart phones) are now able to acquire context, process it and perform reasoning tasks. Moreover, lightweight mechanisms and distributed reasoning can enrich less powerful resources of smart space with intelligent behaviour. We explore these technologies in Section 5.3 in more detail.

3. Types of interaction in smart spaces

We categorize interaction in smart spaces into four types. It is important to note that in this classification we consider that all parties are interaction participants, not just mediators or tools. For example, when two persons have a phone conversation, only the persons are actually involved in interaction. Here, the phones are just the tools, medium to support this interaction. In addition, we consider participants to have user roles. Hence, we do not consider other interaction roles, like interaction between the system developer or administrator and the system. We define the following types of interaction in smart spaces:

User-to-User interaction. This is the most traditional form of interaction, which involves several human participants, i.e. social interaction. Physical proximity of participants is not necessary for this type of interaction, for example when writing a letter to a friend. If interacting participants co-present, then, generally this interaction can be verbal (direct words and actions) or non-verbal (facial expressions, gestures, pose, etc.). People perceive these signals, analyze them and act accordingly. This interaction is usually goal-driven, meaning that users interact in order to achieve certain goals. The exact interaction depends on the context, such as whether interaction is formal, the place of interaction, etc. For this type of interaction, a smart space is just the physical environment supporting the interaction.

User-to-Resource interaction. This interaction type assumes communication between a user and a resource of a smart space. With a resource, we refer to a physical artefact in the environment, which is equipped with computational, sensing, acting, and communica-

tion capabilities. Part of these capabilities can be provided by the environment, e.g. the resource can be tracked by a video camera. In other words, a resource has a particular aim to provide, affect and support the interaction. This interaction type is usually goal-driven, meaning that interaction happens in order to achieve a certain result. The interaction itself can be direct, when the user manually interacts with the resource, or mediated, when the interaction is performed via another resource or physical environment. In this User-to-Resource type of interaction, smart space can be considered as a user interface [47] where user actions lead to system and environment responses.

Resource-to-Resource interaction. This form of interaction assumes communication between the resources of a smart space. This interaction type requires that resources satisfy the requirements for actors given in the introduction, and the interaction happens with and depends on other peer behaviour, as well as situation context; otherwise we deal just with M2M interaction. Interaction like this can be tightly or loosely coupled. The former is characterized by a predefined interaction model, while the latter does not have one. This type of interaction is goal-driven, which means that each communication act has an initial purpose behind it. P2P networks and agent programming can serve as an example for loosely coupled communication, when a predefined flow of services can exemplify a tightly coupled interaction model. Smart space consists of this type of interaction in order to achieve some utility for its users.

Facilitated interaction. This kind of interaction refers to active interaction of smart space resources in order to achieve some social utility for users, e.g. initiate interaction. In general, Facilitated interaction unites all other interaction types: User-to-User, User-to-Resource and Resource-to-Resource interaction (and M2M communication when the resources are not active participants). This interaction type emphasizes the importance of facilitating the user interaction from interacting resources.

We do not explicitly define the interaction between services and the overall infrastructure that the smart space offers because it is clear that infrastructure consists of resources providing services. Figure 1 presents the interaction types in a smart space. The figure emphasizes the hierarchy within interaction in the smart space. That is, Resource-to-Resource interaction is the lowest level and basic interaction type in smart space. Users may even be unaware that such interaction exists in the smart space. The User-to-

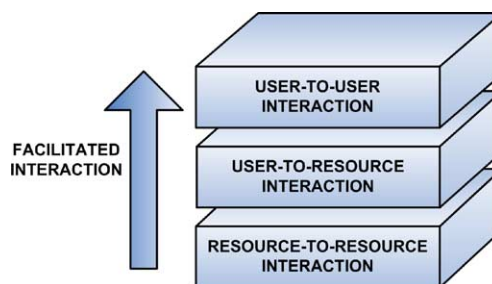


Fig. 1. Smart space interaction types.

Resource interaction type attracts user attention to the situation. User-to-User interaction is the highest level interaction type. All these interaction types can either rely on interactions below or proceed independently. The arrow in Fig. 1 demonstrates the interaction sequence flow in order to achieve Facilitated interaction (from Resource-to-Resource via User-to-Resource to User-to-User). Actually, Facilitated interaction can skip User-to-Resource interaction only when Resource-to-Resource interaction alters the environment in such a way that it encourages users to interact with each other. The categorization of interaction types is shown in Table 1.

4. Context-awareness and interaction

Dey [15] defines context as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. That is, any information which is relevant to the object constitutes the context. As mentioned earlier, in smart space, there can be plenty of context information, which we generalize here as the situation context. Further, we talk about social context in more detail and discuss how interaction and context-awareness are connected.

Social context plays an essential role in describing the situation in which peers interact. Information about social relations can serve as an example of social context for users. For resources of smart space, the communication constraints for specific situations serve as an example of social context. In other words, *social context defines the properties of the relationship and roles of the interacting peers to each other or to a group of peers*. In addition, accepted laws and norms, historical experience, and culture form the social context. Social context depends on the other types of con-

Table 1
Types of interaction in smart space

Type	Volatility level	Direct/Mediated	Goal orientation	Example
User-to-User interaction	High	Both direct (e.g. face-to-face communication) and mediated (e.g. writing email)	Medium (usually goal-driven, however, sometimes goals can evolve from interaction itself)	Traditional face-to-face interaction between humans, such as conversation, argumentation, etc.
User-to-Resource interaction	Low	Both direct and mediated (via another resource or environment, e.g. wall display and mobile phone as its remote controller)	High for direct interaction and can have different forms (Low, Medium, and High) for mediated interaction	Interaction with urban wall display
Resource-to-Resource interaction	Low	Usually direct, but can be mediated, e.g. interaction via changes in environment perceived by other resources (stigmergy)	High	Interaction between agents
Facilitated interaction	Low between resources' interaction, but High between users' interaction	Different combinations exist	High for resources interaction, but Medium for users interaction	Mobile devices notify users when a person nearby has similar interests

text that the smart space provides. For instance, when three colleagues discuss their personal story in a coffee room, it is one social context; but when the same three colleagues are at a work meeting with their boss, the social context is different.

A system that uses context in order to provide relevant information or services to users is called context-aware [15]. Generally, interaction in a smart space is context-aware. That is, interacting actors constantly perceive and analyze context, such as the situation, the time of the day, and act accordingly. Hence, all active interacting actors perform the following steps: formalization of the context, its acquisition and its analysis in order to decide what to do when a certain context changes. These steps are well defined by researchers as context modelling, context acquisition, context reasoning, and acting.

Context modelling refers to the activity of building an adequate representation of the part of the real world using concepts and relations between them. A context model assigns semantic meaning to the information relevant to the system in a formal way. Elementary contexts can be organized into high-level, more meaningful contexts. The real world provides such a model for humans, and language and symbols are utilized to describe it. However, there is no such real world model for computational objects, hence, a shared and unified model of the world must be created when computational objects are involved in interaction. Different approaches exist to create context models, for instance Ontology models and Logic based models [54]. Com-

putational components interact based on the meaning defined in this context model.

Context acquisition refers to capturing the overall situation. This can be done using direct interaction of actors with other participants and the environment. Alternatively, the context can be retrieved with different sensing mechanisms. Changing of the situation is considered as a triggering event. That is, it is a signal for an actor that analysis is needed regarding what has happened and how the actor should react to this event. Tran et al. [58] suggest that social context is obtained not only by using sensors, but retrieval *“requires an understanding of constructed relationships, obligations and constraints underlying the interactions between collaborating actors.”* This implies high volatility of the social context, and constant usage of learning techniques in order to provide interacting actors with the adequate social context.

Humans interpret the captured context by using reflex and cognitive mechanisms. Smart space resources interpret the retrieved context with various reasoning techniques which form the rules defining how the interacting actor should respond to different situations [32]. Some of these techniques include historical analysis, like case-based reasoning (CBR) [1], some tackle well the ambiguity of context, like probabilistic reasoning [45]. Reasoning is usually involved in analysis of the situation and making decisions about actor behaviour. It can be difficult to reason about a social context, because of its subjectivity; for instance, interacting actors may have different interpretations of social context.

Finally, an action can be associated with the result of a reasoning process. The action can lead to the actor's own adaptation or to a change in the environment, which in its turn, can lead to a change in the situation which again has to be reasoned.

As described above, smart space interaction is context-aware. Hence, as a full participant of interaction, the smart space should retrieve and analyze the context relevant to the interaction and act according to it. In other words, the smart space has to be context-aware.

Knowledge-based solutions provide great support for developing context-aware applications. Knowledge-based systems distinguish the process of defining the domain knowledge and the process of controlling the system. In the simplest case, such a system consists of two modules - a knowledge base, containing formal concepts, facts and rules of the domain problem, and an inference engine which contains the reasoning algorithms to produce new knowledge from the knowledge base [21]. Hence, these systems can embody the required cognitive capabilities for active interaction actors in a smart space. Interacting actors would be able to retrieve necessary information, "understand the situation", and make decisions on their own.

5. Technologies supporting smart space interaction

5.1. Technology stack

In this section, we discuss the technology stack required to facilitate interaction in smart spaces (Fig. 2). These technologies are needed for User-to-Resource, Resource-to-Resource, Facilitated and mediated User-to-User interaction types. Direct User-to-User interaction can be performed without technology, although technology can support it.

Infrastructure facilitates all interaction in a smart space by providing the necessary hardware, connectivity support and software. Hardware equips the smart space and its actors with the sensing, acting, and computing facilities (sensor networks, computers, digital supplies). Connectivity provides the communication means between such hardware resources, like WiFi and Bluetooth. Software running on the hardware of a smart space realizes the services and applications that provide utilities for the users.

Interfaces enable interacting actors to discover and communicate with each other. For example, Resource-

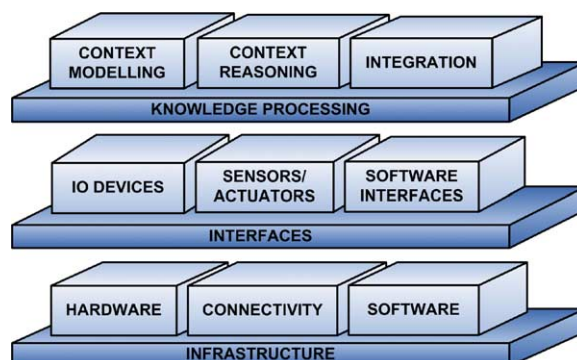


Fig. 2. Technologies supporting smart space interaction.

to-Resource interaction is enabled through various resource communication and access management mechanisms. User-to-Resource interaction is supported by multimodal user interfaces which enable users to interact with resources e.g. using gestures, audio, touch, or traditional IO devices.

Knowledge Processing provides semantics to the interaction. Context modelling defines the common vocabulary of concepts and their relationships for the entire system. All interaction types need a context model, so that the interacting actors have the same meaning for the concepts they operate with. Context reasoning is required for interpreting the information retrieved via interfaces and making decisions. In all types of interaction, actors should constantly reason what actions to do based on certain circumstances. In other words, reasoning utilizes the context model in order to build the rules and logic of interaction. Also, knowledge from a large amount of resources and external knowledge bases can be integrated to provide a more complete view of the situation.

A smart space consists of various technologies and components and hosts plenty of interactions, occurring both synchronously and asynchronously. It is obvious, that a distributed solution should provide more flexible interaction support. Researchers distribute the functionality required for smart space applications to different components [14,42,68]. Kiani et al. [24] consider the scalability advantages distributed solutions provide in order to develop large-scale context-aware systems. Winograd [65] outlines in his essay "*Function should be allocated to processors in whatever way best fits the setting, and the programming metaphor should be based on multiple independent communicating components*". In this article, we support this vision and consider not only the distribution of functions to the different infrastructure components, but

distribution of the technology to different infrastructure components as well. That is, we consider the distribution of knowledge-processing and interface technologies. This distributed solution assumes cooperative behaviour: distributed components collaborate to achieve a common task, because no one actor can have full knowledge about the situation. However, even though the overall system can have a big collaborative task, some actors may compete with each other for resources, for performing subtasks, etc.

Further, we will explore how interfaces and knowledge processing technologies can be distributed in order to support the interaction in a smart space, what challenges can arise, and how they can be solved.

5.2. *Interfaces supporting smart space interaction*

Interfaces support interaction in smart spaces as follows: Resource-to-Resource interaction interfaces enable interacting actors to discover each other and exchange information using various discovery and communication protocols. User-to-Resource interfaces enable users to interact with resources in the smart space using various input and output modalities [2].

Among the most prominent technologies realizing User-to-Resource interaction interfaces are physical browsing solutions which “take advantage of mobile devices physically interacting with tagged objects to facilitate interaction with associated information and services” [8]. A well known early example of a physical browsing solution was proposed by Roy Want et al. in 1999 [64]. Since then, multiple physical browsing solutions have been built for demonstrating various User-to-Resource interaction techniques. In general, physical browsing refers to three interaction techniques, namely touching (e.g. using NFC tags), pointing (e.g. using a laser pointer) [44] and scanning (e.g. using Bluetooth) [8,40]. In addition, User-to-Resource interfaces can be realized using multimodal interfaces that combine two or more input/output modalities, such as speech, touch and gestures [50].

From the users’ perspective, physical browsing is advantageous due to its directness, simplicity and natural characteristics which give users the feeling of trust and being in control over the smart space. Multimodal interfaces, similarly, offer users a natural way of interacting with smart spaces since humans use several senses to perceive the world. From the design point of view, multimodal interfaces and physical browsing make it possible to distribute User-to-Resource interaction between users’ mobile devices and the smart

space resources [7,43]. For example, Turunen et al. [60] demonstrated a prototype of a distributed multimodal interface for smart spaces which was implemented as a combination of a physical browsing solution with gesture and voice recognition technologies.

Realizing User-to-Resource interaction in smart spaces poses various challenges for the smart space’s designers. For example, the smart spaces have to enable and manage users’ concurrent access to resources. This requires the smart spaces to support sharing and collaborative usage of resources, access conflict resolution and timely exchange of information between users [22,43]. Another issue is related to supporting continuous discovery and management of resources. That is, a smart space has to be aware of available resources and their states. This requires the smart space to continuously sense the environment (i.e. perform the resource discovery) in order to detect new resources and changes in the availability and other properties of existing resources. The overall state of the smart space is made available to applications and their users through various distributed or centralized registry mechanisms. This issue can possibly be resolved by using appropriate mechanisms based on dynamic resource discovery protocols [61].

Realizing Resource-to-Resource interaction raises multiple issues related to the communication and discovery mechanisms used by the smart space’s resources. Particularly, resources have to announce their means of communication (e.g. machine addresses, APIs and supported protocols) before they can interact with other resources in the smart space. Usually, this issue is resolved by using a distributed or a centralized registry mechanism. However, centralized solutions may result in performance bottlenecks and may drastically reduce the overall reliability of the smart space. In contrast, distributed registry mechanisms, utilizing P2P or agent technologies, maintain higher reliability, although, often at the expense of greater implementation complexity and longer overall response times [61].

Often, some smart space resources are tiny resource-constrained devices which are unable to utilize traditional discovery and communication protocols. This requires designing lightweight solutions which enable these resource-constrained devices to discover and exchange information with each other. One promising initiative is the Constrained Application Protocol (CoAP) [51] which allows resource-constrained devices to be connected to the Internet. CoAP applies the principles of RESTful design and provides ad-

vanced functionalities, including discovery, caching and event management. Other promising solutions enabling communication between resource-constrained devices are BiTXML⁶ and M2MXML⁷ protocols. BiTXML protocol implements the presentation level of the OSI-based communication stack reference in order to standardize the command and control information exchange specifically for the demands of M2M communication. M2MXML is an open standard XML-based protocol which provides a transport agnostic and extensible framework for M2M communication in wireless networks. Both solutions target various resource-constrained devices such as wireless sensor nodes and portable computing devices.

5.3. Knowledge processing to support interaction

Knowledge Processing technologies enrich smart space interaction with semantics, and provide mechanisms to reason about actions in certain situations. Taking into account the amount and diversity of resources in a smart space, we consider the distributing of knowledge processing as a flexible solution to support interaction in smart spaces.

Knowledge processing distribution deals with the distribution of the knowledge base and inference rules between the system components according to their facilities, locations and policies. For instance, some components can host their own knowledge base and perform reasoning tasks locally. In this section, we discuss Resource-to-Resource interaction support because it is the most challenging case for distributing knowledge processing technologies.

Rich and Knight [39] define the advantages of distributed knowledge processing: system modularity, efficiency, multiple perspectives, heterogeneous reasoning, reliability, and privacy. System modularity brings convenient system development and maintenance when the system is composed of separate components corresponding to their own problem domain and responsible for their own tasks. Each system component can search the local knowledge base and reason more efficiently, hence overall efficiency can be improved. Moreover, different system components can have their own view on the same situation. Hence, multiple perspectives achieve a more complete understanding of the problem. Heterogeneous reasoning, re-

sulting from using different reasoning mechanisms, provides a more flexible solution to the problem. Additionally, some components can perform overlapping functionality; hence, the failure of a single component will not necessarily affect the achievement of the final goal, leading to enhanced reliability. Finally, distributed architecture provides better means for privacy support as sensitive information can be processed locally.

However, design and implementation of a decentralized solution is challenging due to the following issues [39]:

Knowledge representation and awareness. Interacting components must have a correct understanding of the exchanged information, as there is no uniform description of the world (global context model). This is challenging, because interacting actors may represent different features of the same environment, or they can have a different interpretation of the same feature because of perspective or granularity. For instance, location can be given at a room level by one component, but by a geo-point by another. Benerecetti et al. [5] and Bikakis & Antoniou [6] propose the setting up of maps between partial and perspective agent's representations using a Multi-Context Systems approach. This approach assumes that together with the knowledge, agents exchange the mapping rules which bridge the agents' concepts (local context). Distributed logics can also be used to define semantic relations between context models [49].

Awareness refers to making system components aware when something relevant to them occurs. This is needed when a certain important property of the environment cannot be observed by a particular component. In this case, other components must inform the agent about the change or they must act in order to affect the environment in such a way that the agent can sense the change [3,38].

Coordination and cooperation. This issue refers to the coordination and cooperation required for agents to work together. This may include task planning and decomposition, and execution coordination. Process planning techniques can be applied for requirements based composition of complex tasks from atomic ones (task planning). Task decomposition refers to decoupling a complex task into a set of atomic subtasks, and assigning them to the actors. Task assignment can be done, for example, by a coordinator agent, or agents can negotiate for themselves using auction mechanism.

Coordination refers to the management of the task execution process. The approach depends on the com-

⁶www.bitxml.org/ (accessed June 2012)

⁷www.m2mxml.sourceforge.net/ (accessed June 2012)

munication facilities between agents, and the needs for control and synchronization. For instance, one agent can be responsible for synchronizing the behaviour of other agents, like the super-peer in P2P technologies. If agents cannot communicate with each other, the basic assumption can be that actors are rational and behave optimally to fulfil their goals.

Communication. Rich and Knight [39] define two basic communication models for distributed components: blackboard and message passing systems. The blackboard system consists of the Knowledge sources (KSs), the Blackboard and the Control component. KSs are independent modules containing the expertise needed to solve part of the problem. The blackboard is a global data repository containing partial solutions, input data, final solutions and control information. It serves as the communication medium, buffer and triggering mechanism for KSs, which communicate with each other by changing the blackboard. The Control component coordinates the decision making process. The blackboard architecture solves the context distribution problem, as it utilizes a centralized knowledge base – the blackboard [12].

Message passing systems rely on direct communication between components. That is, agents build the models of other agents, and forward problem requests to those they assume can handle them. These systems can utilize modal logic approaches. For instance, Nilsson [30] introduces a knowledge modal operator K that denotes the proposition that agent A knows fact B , $K(A, B)$, axioms and inference rules, so the reasoning can be made. The Belief Desire Intention (BDI) model is another example of introducing modalities [36]. Some agents utilize modal together with temporal logic [18], which introduces the sense of time, such as true now, true always, or true sometimes in the past, present or future.

Distributed algorithms. To be able to solve tasks in a distributed fashion, distributed reasoning algorithms are required. For instance, Viterbo [62] and Ye et al. [67] propose approaches for distributing rule-based reasoning between different nodes. Distributed CBR [1] refers to the distribution of cases between several components. The difficulty appears to be in retrieving the appropriate solution for problems composed from these cases [35]. A number of probabilistic reasoning mechanisms support distributed design by their nature, for instance Neural and Bayesian networks [45]. For example, Schmidt and Abere [48] propose a distributed reasoning mechanism based on a

Bayesian Network and a belief propagation algorithm utilizing P2P technology.

Lightweight solutions. Some devices in a smart space are not able to perform demanding computations due to their resource limitations. These cases require lightweight solutions to represent knowledge and perform reasoning. For instance, Su et al. [56] propose a lightweight mechanism to exchange semantically enriched data. Other researchers propose lightweight reasoning engines capable of running on mobile phones, like Androjena⁸ and KRHyper [52].

6. Prototypes for smart space interaction

In order to exemplify our ideas, we implemented three prototypes which use knowledge-based technologies to support smart space interaction. We gradually implement a more distributed knowledge-based solution to observe the challenges and advantages of distribution.

6.1. Overview of the prototypes

Event Map Application (EMA). The goal of this application is to explore how knowledge-based technologies support smart space interaction. EMA is a context-aware reminder for mobile phone users [55]. The geographical location of the device, a description of points of interest (PoIs) and user tasks provide the context for the application. The social context is formed with the shared tasks created by users. EMA renders maps and reminders to mobile phone screens according to the context. Maps with reminders appear in the foreground only when there is important information to be shown to the users. At other times, users are not disturbed and can perform normal mobile phone activities. The interaction scenario implemented is the following: a wife creates a shared task with her husband to pick up their kids from school. The couple has decided to pick up their kids as soon as possible, so, at the predefined time, the one who is closer to the school is sent a reminder with a map to pick the kids up.

This application was implemented using the client-server application model (see Fig. 3). Thin mobile clients contain only GUIs. The server hosts an OWL ontology knowledge base which contains information about PoIs, users and tasks, and a Jena rule-based rea-

⁸<http://code.google.com/p/androjena/> (accessed June 2012)

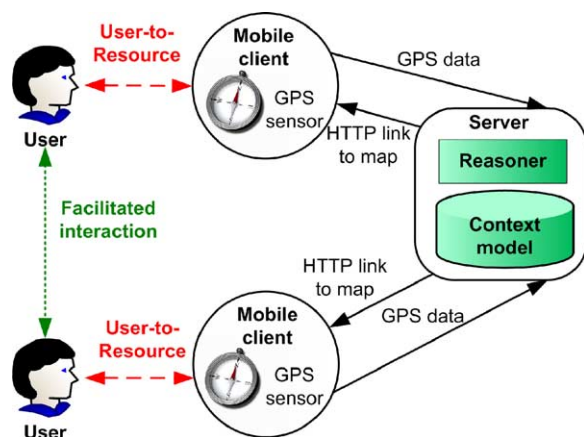


Fig. 3. Interaction in Event Map Application.

soning engine making decisions about actions. Hence, all processing is done on the server side; mobile clients do not make any decisions, they only send and receive the information from the server.

This centralized solution provides better control for social communities (sharing tasks with friends, family, etc.) because it does not involve negotiating and distributing algorithms; it therefore simplifies application control and development. Moreover, this solution saves mobile phones from performing resource demanding reasoning tasks. The main disadvantages of this implementation are: constant network connectivity, limited scalability and limited privacy. Often with mobile applications, it is difficult to achieve constant and reliable network connectivity to the server. This solution creates overhead network traffic when mobile clients send requests to the server with updated GPS coordinates and the server infers no action. Scalability issues should be addressed, as latencies increase proportionally to the size of the knowledge base and amount of the active users in the system. Finally, security mechanisms should be implemented, as personal information is handled at the server.

This application participates in different kinds of interactions, supported by the infrastructure, interface and knowledge processing technologies (see Section 3). The infrastructure provides network connectivity, devices, sensors, the server and the external map rendering service. Interfaces support M2M communication and direct User-to-Resource interaction. The server and mobile clients communicate via API interfaces when a GUI interface initiates User-to-Resource interaction - this happens when users approach certain PoIs. In this prototype, knowledge-processing technologies are responsible for Facilitated interaction.

That is, the interaction of the users involved in a shared task arises from interaction of the mobile clients with the server.

QuizBlasters (QB). QB is a context-aware learning application, which combines elements of a treasure-hunt and a multiplayer action game [13]. This application features a multimodal context-aware interface (iCompose) which enables users to compose applications from resources in the smart space. The application supports two modes, the learning mode for a single user, and the multiplayer action mode. The learning mode motivates users to collect and answer different learning tasks (i.e. quizzes). Users collect quizzes by touching NFC tags placed in the smart space using their mobile phones that are equipped with NFC readers. By completing these quizzes in a timely manner, users make progress and unlock various virtual bonuses. The multi-user gaming mode allows users challenge each other by playing an action game on a wall display. During this game, the players utilize the bonuses and weapons which they earn by completing quizzes in the learning mode. In this game, multiple players destroy each other's avatars with different weapons. Users control their avatars on a wall display using their mobile devices as remote controllers. In order to play the game, two or more users need to compose the application using their mobile devices and a wall display. In this case, users choose opponents for the game by bringing their mobile phones into contact. Likewise, users choose a wall display by touching it with a mobile phone.

QB utilizes multiple kinds of context, including the users' physical location, study performance, daily schedule, and proximity to other users. The social context specifies the users' roles (e.g. students or a teacher), and the relationships between them (e.g. friends or classmates). Particularly the social context specifies the so called "rules of engagement", i.e. when, where and with whom the players can play the multi-user game. These rules enforce a certain behaviour and order (i.e. restrict undesirable usage of certain resources, restrict gaming during classes, etc.).

This application was implemented using a hybrid centralized-distributed architecture, as shown in Fig. 4. The application's interface is distributed between the smart space (NFC-based quiz tags in the environment), the users' mobile devices and wall displays. In addition, the knowledge processing functionality is distributed between multiple mobile devices and the server as follows: each mobile device hosts a reason-

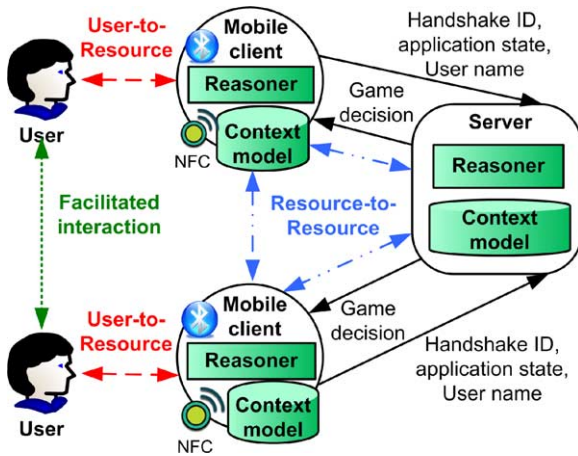


Fig. 4. Interaction in QuizBlasters application.

ing engine which is capable of performing decision-making independently of the server. The latter also hosts a reasoning engine. This configuration makes it possible to preserve a context that is sensitive to the users on their mobile devices as, in most of cases, their reasoning engines are able to perform reasoning locally and involve the server only if required. Although distributed, this reasoning process still relies on the server when multiple mobile devices need to synchronize their decision-making processes.

This application supports both Resource-to-Resource and User-to-Resource interaction. User-to-Resource interaction is implemented using a multimodal interface comprised of a physical browsing interface and a traditional graphical user interface, combined with knowledge based technologies. The latter provide semantics and decision-making capabilities for interaction. For instance, if a student tries to compose a multi-user game, the student's local reasoning engine permits or rejects this operation depending on whether the student has a class at that time. In addition, the iCompose application facilitates User-to-User interaction through the messaging mechanism, which allows users 1) to detect potential players in the smart space using a Bluetooth-based discovery protocol and 2) to broadcast these players' "invitation messages", which allow them to join a game on a nearby wall display. Finally, the iCompose interface supports Facilitated interaction. For example, when composing the multi-user game, reasoning engines, running on the users' mobile devices, enable them to choose a wall display and players for the game.

Share Interests (SI). This application enables us to explore knowledge processing distribution. SI is a mo-

bile context-aware application which uses Bluetooth to detect people with common interests. Users specify their hobbies and the types of alerts they expect to receive. These hobbies are described using private and public profiles as specified later in the text. Users also create the rules which detect common interests with others. These are logical IF-THEN rules and they are included in profiles. When the application is running, mobile phones scan for users nearby. If some users are detected, then the application exchanges the rules in order to find common interests. We utilize the rule distribution algorithm suggested by Viterbo [62] in order to exchange the data defined in the rules' conditions, instead of whole profiles. If a match is detected, then, both users receive a common interests alert.

This application has a distributed architecture (see Fig. 5). The RDF-based knowledge base and the Androjena rule-based reasoning component are hosted in each mobile device, hence all decisions are performed locally. This allows users to have full control over the application. In SI, the knowledge base is separated into public and private profiles. The private profile contains user-sensitive facts and rules. The public profile, in turn, contains the facts and rules which can be exchanged with other users. The public profile has a predefined context model which strictly defines concepts and their relations. This context model is the same for all clients. The private profile can contain a context model unique for each client, so that it can be expressed in any user-defined format. Common interests are expressed using IF-THEN rules, where conditions can be presented as the conjunction of elementary conditions. Users also specify the rules for handling Bluetooth requests, such as when they are accepted and when they are rejected. The session mechanism regulates the synchronization of communication between users.

The SI application demonstrates how Facilitated interaction can be supported. That is, users' mobile devices interact in order to facilitate interaction between their users. Mobile phones perform their job unobtrusively. Users are notified only when a match is found. Interfaces support User-to-Resource interaction by providing an unobtrusive alerting mechanism. Knowledge processing and interface technologies support Resource-to-Resource interaction. Mobile devices interact with each other on behalf of their users; they process Bluetooth requests, exchange rules and facts, detect matches and inform users.

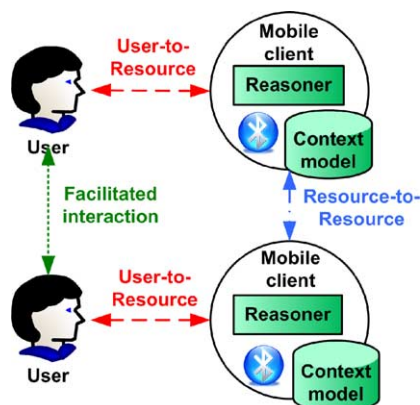


Fig. 5. Interaction in Share Interests application.

6.2. Analysis of the prototypes

These three implemented prototypes explore Facilitated smart space interaction. In EMA, user interaction arises from direct communication between mobile phones and the decision-making server. This application demonstrates mediated User-to-User interaction, which means that there is no direct face-to-face interaction between humans; the interaction is mediated via rules and messages of the application. In the QB application, both direct and mediated User-to-User interaction evolves from direct interaction between users' mobile clients and infrastructure (servers and services). SI contributes to direct User-to-User interaction from direct interaction between several mobile phones. In this application, mobile clients make decisions themselves and notify their users only when a match is found. Although the interactions in these three applications are goal-driven, QB enables users to initiate interaction simply by exploring the environment and trying different interface modalities, like touching phones together.

Our prototypes consider the smart space with its resources as an active participant of the interaction. For instance, the QB application motivates users to interact with the infrastructure and its resources. SI and EMA applications constantly participate in decision-making in order to provide social utility for users. The environment is not isolated from the applications, but it is an active part of them. All three applications utilize user context in decision-making. The developed applications possess a certain level of autonomy and intelligence to assist users and provide certain utilities for them.

We compare our three prototypes with the related work according to the degree of context-awareness and

smart space interactions these systems support. Smart space interaction evaluation allows us to explore the interaction types that the system provides. Analysing the support for context-awareness allows us to evaluate the technologies and solutions used to achieve the interaction. Finally, we are interested in seeing if there is some correlation between these two characteristics. We compare four works with our prototypes: we analyze the social application by Eagle and Pentland [16] to find people with similar interests. Truong et al. [59] present a solution to design and implement collaborative systems, and a meeting scheduling prototype, which we also analyze. Kabir et al. [23] propose an approach to model and coordinate social context. Their prototype application is the third focus of our analysis. Beach et al. [4] present the WhozThat identity sharing protocol, making it possible to share the social context from social networks. We analyze their context-aware music player prototype. All these systems are socially-oriented and achieve Facilitated interaction.

Figure 6 visually compares the above mentioned systems with our prototypes according to the context-awareness features supported. All analyzed systems utilize social context, represented with different approaches, in order to support users. For instance, Eagle and Pentland [16] and SI utilize profiles with public features in order to find the similar interests among users. Beach et al. [4] use the profile information found in social networks. In contrast, QB, Kabir et al. [23], and Truong et al. [59] explicitly model the social relations of users. This approach is the most advanced, as it considers the high volatility of the social context.

Among reasoning mechanisms, rules are used most often. This allows users to express their preferences more intuitively, as well as utilize advanced rule-based technologies for decision-making. EMA, SI, and Truong et al. [59] utilize semantic-based rule-technologies to perform reasoning. Kabir et al. [23] implement different rule approaches in their system: fuzzy rules deal with conflicts, constraint rules are used to specify temporal relations, and Drools rules define context changes. QB implements logical programming rules. Both Eagle and Pentland [16] and Beach et al. [4] implement their own algorithms in order to trigger certain system behaviour. This approach is less flexible, in our opinion, as it limits the system maintenance. If we consider the architecture of context-awareness technologies, then a centralized schema, where the knowledge base and the reasoner are located

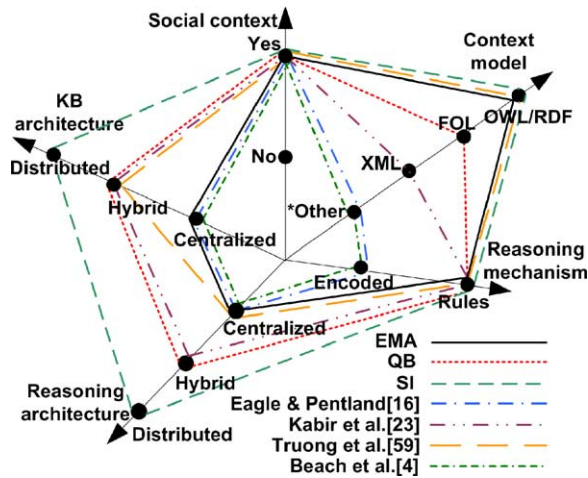


Fig. 6. Context-awareness support (*No detailed information).

at the central node, is the most often used; however, all work referred to here mention the limitations of such an approach. As can be observed from Fig. 6, the higher the polygon area, the more advanced and flexible technologies are used in the system to support the context-awareness.

Figure 7 presents the smart space interaction support provided by analyzed systems. We apply the interaction categories introduced earlier to describe the interaction capabilities of these systems. We also analyze whether interaction is Automatic, Semi-Autonomous or Modifiable. Automatic interaction means that the interaction pattern is hardcoded and trigger-based. Semi-Autonomous interaction means that interacting actors make decisions by themselves. Modifiable interaction denotes that autonomous decisions can be modified so that the behaviour of the interaction changes when necessary. The further to the right the system is located on the black horizontal arrow of Fig. 7, the more autonomous and flexible it is. From the analyzed systems, QB, SI and Kabir et al. [23] implement direct Resource-to-Resource interaction, the rest perform M2M communication. Kabir et al. [23] and SI allow modification for resource interaction behaviour. At the opposite end, resources in EMA, Beach et al. [4], Eagle and Pentland [16], and Truong et al. [59] have predefined triggers in order to communicate with other resources. These systems present a thin client solution, where decision-making is performed at the server. For instance, EMA constantly sends GPS data to the server, and the server always responds on whether to act and how. QB performs Semi-Autonomous Resource-to-Resource interaction, mean-

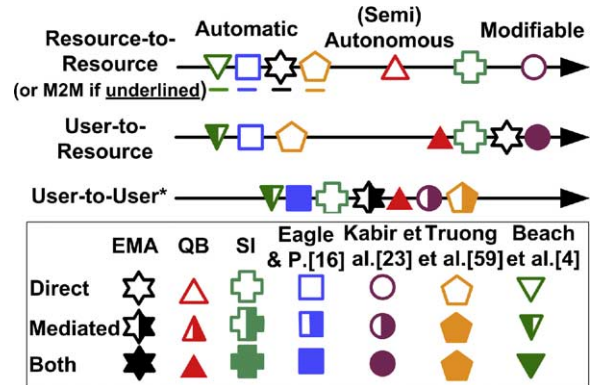


Fig. 7. Smart space interaction support (Categories, marked with * are not classified as Automatic/(Semi)Autonomous/Modifiable).

ing that resources decide on their own about whether to initiate the interaction, based on context.

Different forms of User-to-Resource interaction can be observed. In Kabir et al. [23] and QB, User-to-Resource interaction can be both direct and mediated. This means that in these systems users can directly interact with resources or via other resources in the environment. Other systems present either mediated User-to-Resource interaction (Beach et al. [4]) or direct (Eagle and Pentland [16], Truong et al. [59], SI, EMA). Some systems allow changing the behaviour by modifying the resource rules (Fig. 7), like in Kabir et al. [23], QB, SI, and EMA. Eagle and Pentland [16] and Beach et al. [4] are little more restrictive in this sense, as reasoning is performed at the server.

Different forms of User-to-User interaction can be found as well. SI presents direct User-to-User interaction. QB and Eagle and Pentland [16] allow both direct and mediated User-to-User interaction. Other systems present the mediated form of User-to-User interaction, where users interact via resources.

As mentioned above, all analysed applications demonstrate Facilitated interaction, its mediation and autonomy characteristics are composed from corresponding Resource-to-Resource, User-to-Resource and User-to-User interactions. As Fig. 7 demonstrates, the implemented prototypes possess some intelligence in order to perform decision-making by the resources.

Kabir et al. [23], QB, and SI are the most advanced systems for interaction support. More advanced smart space solutions can be developed based on Beach et al. [4] and Truong [59], as we have evaluated just prototypes for their frameworks. EMA and Eagle and Pentland [16] implement basic interaction support nec-

essary to serve applications. Of course, smart space interaction support is a matter of the scenario and selected design solutions. However, we can observe some correlation between the context-awareness technologies used and the system autonomy level for interaction support. Systems which implement more advanced context-awareness technologies (Fig. 6) are mostly placed towards the right (Fig. 7). This seems to be logical, as advanced solutions provide flexibility and maintenance advantages over hardcoded approaches.

All implemented prototypes demonstrate that interface design is especially important for User-to-Resource interaction. To support direct User-to-Resource interaction, both the EMA and SI applications provide unobtrusive alert mechanisms for their users. Similar approaches can be observed in related work (e.g. Eagle and Pentland [16], Truong et al. [59]). The interaction behaviour can be changed by rules or tasks modifying GUI. Kabir et al. [23] provide similar functionality. The interface in QB is unique: it is distributed between the environment, mobile phones and the infrastructure, and supports several modalities, such as touch-based and manual control. The challenge for such interface design is that users should perceive such interaction as natural and not requiring specific skills to be learnt. Resource-to-Resource interface design requires well defined protocols. In our applications, we utilize XML, JSON and Entity Notations [56] to construct such data-exchange protocol between resources. All our applications use socket and HTTP interfaces for Resource-to-Resource and M2M communication.

Knowledge-based technologies play a key role in our applications. Also, only two among the analysed work do not use them explicitly (see Fig. 6). Generally, each prototype has a different architecture for the reasoning mechanism. Distributing the reasoning functionality greatly decreases the network overload. Our QB and SI applications have faster response times, and also scalability, reliability and privacy support advantages over EMA. However, these advantages are achieved at a price of more difficult and time-consuming implementation and thick client models, consuming the resources of mobile phones.

Our prototypes demonstrate an evolution from a fully centralized design approach through hybrid to a distributed knowledge-based technologies approach. Obviously, each design solution has its own advantages and disadvantages. Based on our experience, we highlight the following: 1) Developers must pay at-

ention to scalability issues for centralized solutions. This is an obvious observation; however, in EMA, we encountered increasing latencies with an increasing amount of users. 2) An approach relying on connectivity requires careful handling for network disconnections, for example, through consistency support. For instance, EMA relies on constant network connectivity; however, it is not possible to guarantee that due to the mobile nature of the application and its geographical usage scale. Also, the data-exchange protocol should produce compact messages and should not be expensive for mobile users. 3) One of the main advantages of a distributed solution is privacy support. We observed that distributed knowledge-processing technologies support this well. Our QB and SI applications did not need any extra solutions to support privacy, as all decision-making involving sensitive information was done locally. Obviously, EMA requires such data-protection mechanisms. 4) Knowledge-based management solutions (such as cleaning, archiving, etc.) must be considered when implementing distributed knowledge-processing technology for resource limited devices. Indeed, the memory space of such devices is limited. The SI application does not use much dynamic context; hence, the size of the knowledge base is not an issue for this application. The QB application implements a cleaning mechanism, which deletes irrelevant or outdated facts.

7. Discussion

In this article, we have considered a smart space as an active participant of interaction. We clarify the types of interaction possible in such a smart space. Also, we consider the importance of the context in such interaction, and discuss the social context. In addition, this article identifies necessary technologies to facilitate smart space interaction. Moreover, we exemplify our concepts with three prototypes.

We have defined four types of interaction possible in a smart space. This categorization could help to set up guidelines for smart space systems design. We find Facilitated interaction most interesting. This type of interaction could be an ideal goal for research targeting social interaction support. How can human social interaction be facilitated through interactions between the resources of a smart space? What utility does interaction among smart space resources bring for users? These difficult questions raise challenging research issues. For instance, privacy support and the feeling of

being in control over the situation are relevant examples in this context.

Different technologies support smart space interaction. We categorized them into three groups. If we consider particularly users, then, for smart space interaction, they utilize mainly the Interfaces. Users provide information, consume, and interact through natural and multimodal interfaces. Knowledge management and Infrastructure lie in the background and are not observable for users and perhaps that is as it should be. However, for the interaction of resources in a smart space, all of these groups of technologies move to the foreground. It is vitally important for resources to agree on the concepts they utilize, to define the interfaces to communicate, etc. There are many challenges which were partly discussed in Section 5.

Three prototypes were implemented in order to exemplify our views. All these prototypes implement different types of smart space interaction. Moreover, they all achieve Facilitated interaction type. That is, all of our prototypes achieve some social utility for their users as they do facilitate interaction. We gradually implemented more and more distributed knowledge-based solutions into the prototypes to see whether it brings flexibility for our applications. It is obvious that a distributed knowledge-processing solution requires more work, as lightweight mechanisms should be implemented. However, with our prototypes, we clearly see the advantages of such a distributed design (refer to Section 6 for more details). We do not argue that our architectural solution is the only right one. As we discussed in Section 6, every architecture has its advantages and disadvantages, and heavily depends on the resources used and scenario.

Interactions in a smart space are dynamic and volatile. They evolve from changes in the smart space, user context, events, etc. We find distributed knowledge-based systems to be a flexible solution for pervasive scenarios. First, this approach enriches the system components with the semantics and logic of interaction; hence, each interacting component can participate in many interesting interaction scenarios, make its own decisions and act based on certain circumstances. Moreover, interacting components can initiate (achieving an active role in smart space interaction) and even facilitate interaction among users, as it was demonstrated with the prototypes. As technological development continues, we may expect more and more devices that able to perform computational activities; hence, we will be able to enrich them with the semantics of the situations they are used in. This

way, the system can acquire knowledge from the environment, and not only the data, by studying objects and their semantic relations. This could improve the learning capabilities of the system and devices as well.

The work towards interactive smart spaces is ongoing. Researchers work to recognize human social activities in smart spaces. When these activities can be recognized, it will be possible to provide support and services for them – considering the smart space as an active actor. Specifically, lightweight and distributed solutions would help to enrich the surrounding objects with the intelligence needed for smart space interaction.

Acknowledgments

We would like to thank all our colleagues for implementing the prototypes mentioned in this article: Iván Sánchez Milara, Jussi Mäkipelto, Mikko Pyykkönen, Paweł Kwiatkowski, Tomasz Latkowski, Alma Pröbstl, Bartłomiej Wójtowicz, Kun Wang and Yili Jin. Also, we would like to thank all test users who helped us to analyze the designed applications. We thank Dr Pentti Luoma for fruitful discussions about aspects of social interaction. This work was carried out in the Pervasive Service Computing project funded by the Ubiquitous Computing and Diversity of Communication (MOTIVE) program of the Academy of Finland. Ekaterina Gilman would like to thank GETA (the Graduate School in Electronics, Telecommunications and Automation), the Walter Ahlström and Tauno Tönnig foundations for funding support. Xiang Su would like to thank the Infotech Oulu Graduate School and the NOKIA Foundation for funding.

References

- [1] A. Aamodt and E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AI Communications* 7(1) (1994), 39–59.
- [2] Z. Antoniou and S. Varadan, Intuitive mobile user interaction in smart spaces via NFC-enhanced devices, in: *Proc. of 3rd Int. Conf. on Wireless and Mobile Communications*, March 2007, pp. 86.
- [3] P. Barron and V. Cahill, Using stigmergy to co-ordinate pervasive computing environments, in: *Proc. of 6th IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 2004, pp. 62–71.
- [4] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han, WhozThat? Evolving an ecosystem for context-aware mobile social networks, *IEEE Network* 22(4), 50–55.

- [5] M. Benerecetti, P. Bouquet, and M. Bonifacio, Distributed context-aware systems, *Human-Computer Interaction*, **16**(2–4) (2001), 213–228.
- [6] A. Bikakis and G. Antoniou, Distributed defeasible contextual reasoning in ambient computing, in: *Proc. of the European Conference on Ambient Intelligence*, E. Aarts et al., eds, LNCS, Vol. 5355, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 308–325.
- [7] M. Blumendorf, D. Roscher, and S. Albayrak, Dynamic user interface distribution for flexible multimodal interaction, in: *Proc. of the Int. Conf. on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ACM, NY, USA, 2010, Article 20, 8 pp.
- [8] G. Broll, E. Rukzio, M. Paolucci, M. Wagner, A. Schmidt, and H. Hussmann, PERCI: Pervasive service interaction with the internet of things, in: *IEEE Internet Computing*, Nov./Dec. 2009, pp. 74–81.
- [9] C.-W. Chen, R.C. Ugarte, C. Wu, and H. Aghajan, Discovering social interactions in real work environments, in: *Proc. of the IEEE Int. Conf. on Automatic Face & Gesture Recognition and Workshops*, March 2011, pp. 933–938.
- [10] S. Cohen, Social relationships and health, *American Psychologist* **59**(8), (2004), 676–684.
- [11] D. Cook, A. Crandall, G. Singla, and B. Thomas, Detection of social interaction in smart spaces, *Journal of Cybernetics and Systems, special issue on social awareness in smart spaces* **41**(2) (2010), 90–104.
- [12] D.D. Corkill, Collaborating software: Blackboard and multi-agent systems & the future, in: *Proc. of the International Lisp Conference*, 2003, Invited paper.
- [13] O. Davidyuk, E. Gilman, I. Sánchez, J. Mäkipelto, M. Pyykkönen, and J. Riekkii, iCompose: Context-aware physical user interface for application composition, *Central European Journal of Computer Science* **1**(4) (2012), 442–465.
- [14] A.K. Dey, Providing architectural support for building context-aware applications, PhD Dissertation, Georgia Institute of Technology, 2000.
- [15] A.K. Dey, Understanding and using context, *Personal and Ubiquitous Computing* **5**(1) (2001), 4–7.
- [16] N. Eagle and A. Pentland, Social serendipity: Mobilizing social software, *Pervasive Computing, IEEE* **4**(2) (2005), 28–34.
- [17] N. Eagle, A. Pentland, and D. Lazer, Inferring social network structure using mobile phone data, *Proc. of the National Academy of Sciences* **106**(36) (2009), 15274–15278.
- [18] E.A. Emerson, Temporal and modal logic, in: *Handbook of Theoretical Computer Science*, J. van Leeuwen (ed.), Elsevier, 1990, pp. 996–1072.
- [19] A. Hang, G. Broll, and A. Wiethoff, Visual design of physical user interfaces for NFC-based mobile interaction, in: *Proc. of the 8th ACM Conf. on Designing Interactive Systems*, ACM, NY, USA, 2010, pp. 292–301.
- [20] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kad-doura, and E. Jansen, The Gator Tech Smart House: A programmable pervasive space, *Computer* **38**(3) (2005), 50–60.
- [21] A.A. Hopgood, The state of artificial intelligence, in: *Advances in Computers*, Vol. 65, Elsevier, 2005, pp. 1–75.
- [22] M. Jurmu, H. Kukka, S. Hosio, J. Riekkii, and S. Tarkoma, Leasing service for networks of interactive public displays in urban spaces, in: *Proc. of the 6th Int. Conf. on Advances in Grid and Pervasive Computing*, May 11–13, LNCS, Vol. 6646, Springer, Berlin/Heidelberg, 2011, pp. 198–208.
- [23] M.A. Kabir, J. Han, and A. Colman, Modeling and coordinating social interactions in pervasive environments, in: *Proc. of the 16th IEEE Int. Conf. on Engineering of Complex Computer Systems*, April 2011, pp. 243–252.
- [24] S.L. Kiani, B. Moltchanov, M. Knappmeyer, and N. Baker, Large-scale context-aware system in smart spaces: Issues and challenges, in: *Internet Communications (BCFIC Riga)*, 2011 Baltic Congress on Future, Feb. 2011, pp. 173–180.
- [25] M. Kranz, P. Holleis, and A. Schmidt, Embedded interaction – interacting with the internet of things, in: *IEEE Computing, Special Track on Internet of Things*, April–June 2010.
- [26] J. Lifton, M. Laibowitz, D. Harry, N.-W. Gong, M. Mittal, and J.A. Paradiso, Metaphor and manifestation cross-reality with ubiquitous sensor/actuator networks, *Pervasive Computing, IEEE* **8**(3) (2009), 24–33.
- [27] G. Lugano, Mobile social software: Definition, scope and applications, in: *eChallenges conference*, Holland, October 2007.
- [28] K. Lukka, The constructive research approach, in: *Case Study Research in Logistics*, L.Ojala and O.-P. Hilmola, eds, 2003, pp. 83–102.
- [29] M.C. Mozer, Lessons from an adaptive house, in: *Smart environments: Technologies, protocols, and applications*, Hoboken, NJ, D. Cook and R. Das, eds, J. Wiley & Sons, 2005, pp. 273–294.
- [30] N.J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1998.
- [31] N.M. Oliver, B. Rosario, and A.P. Pentland, A Bayesian computer vision system for modeling human interactions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8) (2000), 831–843.
- [32] M. Perttunen, J. Riekkii, and O. Lassila, Context representation and reasoning in pervasive computing: A review, *International Journal of Multimedia and Ubiquitous Engineering* **4**(4) (2009), 1–28.
- [33] M. Philipose, K. Fishkin, M. Perkwowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel, Inferring activities from interactions with objects, *IEEE Pervasive Computing* **3**(4) (2004), 50–57.
- [34] S. Pirttikangas, I. Sánchez, M. Kauppila, and J. Riekkii, Comparison of touch, mobile phone, and gesture based controlling of browser applications on a large screen, in: *Adjunct Proceedings Pervasive Sydney*, Australia, 2008, pp. 5–8.
- [35] N.M.V. Prasad, V. Lesser, and S. Lander, Retrieval and reasoning in distributed case bases, *Journal of Visual Communication and Image Representation*, Special Issue on Digital Libraries, **7**(1) (1996), 74–87.
- [36] A.S. Rao and M. Georgeff, BDI agents: From theory to practice, in: *Proc. of the 1st Int. Conf. on Multi-Agent Systems*, 1995, pp. 312–319.
- [37] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, Context-Phone – A prototyping platform for context-aware mobile applications, *IEEE Pervasive Computing* **4**(2) (2005), 51–59.
- [38] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva, Cognitive stigmergy: A framework based on agents and artifacts, in: *Environments for Multi-Agent Systems III*, D. Weyns et al., eds, LNCS, Vol. 4389, 2007, pp. 124–140.
- [39] E. Rich and K. Knight, *Artificial Intelligence*, McGraw-Hill, Inc., 1991.
- [40] J. Riekkii, T. Salminen, and I. Alakarppa, Requesting pervasive services by touching rfid tags, *IEEE Pervasive Computing* **5** (2006), 40–46.

- [41] J. Riecki, I. Sanchez, and M. Pyykkönen, Remote control for pervasive services, *International Journal of Autonomous and Adaptive Communications Systems* 3(1) (2010), 39–58.
- [42] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R.H. Campbell, and K. Nahrstedt, A middleware infrastructure for active spaces, *IEEE Pervasive Computing* 1(4) (2002), 74–83.
- [43] D. Roscher, M. Blumendorf, and S. Albayrak, A meta user interface to control multimodal interaction in smart environments, in: *Proc. of the 14th Int. Conf. on Intelligent User Interfaces*, ACM, NY, USA, 2009, pp. 481–482.
- [44] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, and J.S.-Y. Chin, An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning, in: *Proc. on the 8th Int. Conf. on Ubiquitous Computing*, 2006, pp. 87–104.
- [45] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach 2/E*, Prentice Hall, 2002.
- [46] A. Sayouti, F. Qrichi Aniba, and H. Madromi, Interactions between agents as shared resources in multi-agents systems, in: *New Technologies, Mobility and Security*, Nov. 2008, pp. 1–4.
- [47] A. Schmidt, M. Kranz, and P. Holleis, Interacting with the ubiquitous computer: towards embedding interaction, in: *Proc. of the Joint Conf. on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies*, ACM, NY, USA, 2005, pp. 147–152.
- [48] R. Schmidt, and K. Aber, Efficient Peer-to-Peer belief propagation, in: *Proc. of OTM Confederated Int. Conf., CoopIS, DOA, GADA, and ODBASE 2006*, R. Meersman and Z. Tari, eds, LNCS, Vol. 4275, Springer-Verlag, Berlin, Heidelberg, pp. 516–532.
- [49] L. Serafini and A. Tamilin, DRAGO: Distributed reasoning architecture for the semantic web, in: *Proc. of 2nd European Semantic Web conf. on The Semantic Web: Research and Applications*, A. Gómez Pérez, and J. Euzenat, eds, LNCS, Vol. 3532, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 361–376.
- [50] R. Sharma, V.I. Pavlovic, and T.S. Huang, Toward multimodal human-computer interface, *Proc. of the IEEE* 86(5) (1998), 853–869.
- [51] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, *Constrained Application Protocol (CoAP)*, Internet-Draft (work in progress) <draft-ietf-core-coap-08>, Sensinode, Universitaet Bremen TZI, SkyFoundry (accessed January 2012).
- [52] A. Sinner and T. Kleemann, KRHyper – In your pocket, in: *Automated Deduction – CADE-20*, R. Nieuwenhuis, ed., LNCS, Vol. 3632, Springer, Berlin/Heidelberg, 2005, pp. 737.
- [53] J. Stan, V.-H. Do, and P. Maret, Semantic user interaction profiles for better people recommendation, in: *Proc. of the Int. Conf. on Advances in Social Networks Analysis and Mining*, July 2011, pp. 434–437.
- [54] T. Strang and C. Linnhoff-Popien, A context modeling survey, in: *Proc. of the Workshop on Advanced Context Modelling, Reasoning and Management within 6th Int. Conf. on Ubiquitous Computing*, Nottingham, England, 2004.
- [55] X. Su, E. Gilman, P. Kwiatkowski, T. Latkowski, A. Pröbstl, B. Wójtowicz, and J. Riecki, Knowledge-based systems for ambient social interactions, in: *Ambient Intelligence*, D. Keyson et al., eds, LNCS, Vol. 7040, Springer, Berlin/Heidelberg, 2011, pp. 61–71.
- [56] X. Su, J. Riecki, and J. Haverinen, Entity notation: Enabling knowledge representations for resource constrained sensors, in: *Personal and Ubiquitous Computing*, 2011, pp. 1–16.
- [57] A. Toninelli, A. Pathak, and V. Issarny, Yarta: A middleware for managing mobile social ecosystems, in: *Advances in Grid and Pervasive Computing*, Riecki et al., eds, LNCS, Vol. 6646, Springer, Berlin/Heidelberg, 2011, pp. 209–220.
- [58] M.H. Tran, J. Han, and A. Colman, Social context: Supporting interaction awareness in ubiquitous environments, in: *Proc. of the Int. Conf. on Mobile and Ubiquitous Systems: Networking & Services*, July 2009, pp. 1–10.
- [59] H.-L. Truong, S. Dustdar, D. Baggio, S. Corlosquet, C. Dorn, G. Giuliani, R. Gombotz, Y. Hong, P. Kendal, C. Melchiorre, S. Moretzky, S. Peray, A. Polleres, S. Reiff-Marganiec, D. Schall, S. Stringa, M. Tilly, and H.Q. Yu, inContext: A pervasive and collaborative working environment for emerging team forms, in: *Proc. of the Int. Symp. on Applications and the Internet*, July 2008, pp. 118–125.
- [60] M. Turunen, A. Kallinen, I. Sanchez, J. Riecki, J. Hella, T. Olsson, A. Melto, J.P. Rajaniemi, J. Hakulinen, and E. Makinen, Multimodal interaction with speech and physical touch interface in a media center application, in: *Proc. of the Int. Conf. on Advances in Computer Entertainment Technology*, ACM, 2009, pp. 19–26.
- [61] C.N. Ververidis and G.C. Polyzos, Service discovery for mobile Ad Hoc networks: a survey of issues and techniques, *Communications Surveys & Tutorials, IEEE* 10(3), 30–45.
- [62] J. Viterbo, Decentralized reasoning in Ambient Intelligence, PhD dissertation, Pontificia Universidade Católica do Rio de Janeiro (PUC-Rio), 2009.
- [63] L. Wang, T. Gu, X. Tao, J. Lu, and M. Tscheligi, Sensor-based human activity recognition in a multi-user scenario, in: *Ambient Intelligence*, B. de Ruyter et al., eds, LNCS, Vol. 5859, Springer, Berlin/Heidelberg, 2009, pp. 78–87.
- [64] R. Want, K.P. Fishkin, A. Gujar, and B.L. Harrison, Bridging physical and virtual worlds with electronic tags, in: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems: The CHI is the Limit*, ACM, 1999, pp. 370–377.
- [65] T. Winograd, Architectures for context, *Human-Computer Interaction* 16(2) (2001), 401–419.
- [66] Y. Xu, J. Gu, J. Yang, and Z. Zhang, An ontology-based approach for mobile personalized recommendation, in: *Proc. of the Int. Conf. on Services Science, Management and Engineering*, July 2009, pp. 336–339.
- [67] J. Ye, J. Li, H. Shi, X. Gu, and Z. Zhu, DFRe: A distributed fuzzy reasoning engine for personalization recommendation, in: *Proc. of 3rd Int. Conf. on Pervasive Computing and Applications*, Vol. 2, 2008, pp. 576–581.
- [68] A. Zimmermann, Context management and personalisation: A tool suite for context- and user-aware computing, PhD dissertation, Fraunhofer FIT, 2007.