A Framework for Smart Object and Its Collaboration in Smart Environment

Kh. Tarik-Ul Islam, Jehad Sarkar, Kamrul Hasan, M Rezwanul Huq, Andrey V. Gavrilov, Young-Koo Lee, Sungyoung Lee
Ubiquitous Computing Lab, Department of Computer Engineering, Kyung Hee University, South Korea
{khtarik, jehad, kamrul, rhuq}@oslab.khu.ac.kr, avg@oslab.khu.ac.kr, yklee@khu.ac.kr, sylee@oslab.khu.ac.kr

***Abstract* — In this paper we present the model of smart collaborative objects that can share the responsibilities of backend server by local processing and collaborating with other objects. We propose the technique to adapt the local inferring using the distributed knowledgebase and a framework for development of context aware software for smart environment We also propose a platform for collaborative object to reduce communication overhead and thus save enormous energy.**

***Keywords* — Digital home network, smart objects and environment, distributed computing, distributed inferring, embedded device.**

1. Introduction

Computing is becoming intrinsically mobile and ubiquitous [1, 2]. Computer-based systems are going to be embedded in all our everyday objects and environments [3,4]. Development of these systems must account following features or requirements:

- Sentience: These applications are context-aware, i.e. have the ability to perceive the state of the environment, through the fusion and interpretation of information collected from diverse sensors and other smart objects.
- Autonomy: Components of these applications will be capable of acting in a decentralized fashion, based on the acquisition of information from the environment and on distributed knowledge.
- Decentralization: There is no single central server that does intensive computation for the clients. A typical application consists of components that might be scattered across geographical regions, e.g., street, buildings, cities, countries, and continents. And distribution of these components may be changed during their lifetimes.
- Adaptivity: These applications will have to cope with changing conditions during their lifetimes. Not only the applications to be designed to evolve, but also their underlying support must be adaptable as well.
- Proactivity: These applications are able to act in anticipation of future goals or problems without direct human intervention. They should have a certain degree of intelligence, and be able to decide what action to take from gathered sensor data.
- Time and Safety Criticality: These applications interact with physical environments and are required to provide real-time services to human users. To do so some system or middleware modules, e.g., resource management and configuration, timing failure detection and Quality of Service (QoS) management can be used.

In this paper we present our performance about features of organization of context and communication in smart collaborative objects. As example of smart environment we basically consider health care system. This work is part of investigations conducting in Ubiquitous Computing Laboratory of KHU [8, 9].

2. Model of Smart Collaborative Object

Functionality of health care system may be relatively divided into the following areas:

- Special medicine tasks (e.g. diagnostics, registration, management of course of (medical) treatment and so on);
- Special tasks solving by mobile robots (e.g. delivery of needed objects and medicine, help to move or make any actions for disable patients);
- Common tasks for smart environments, dealing with different kinds of monitoring.

First two of these tasks are not dealing with smart objects and are not focused in this paper. Examples of tasks from third area, our interested area, are following:

1. Monitoring for behavior (activity) of patient in a ward
    a. Recognize and prediction of activity (e.g. reading of book, watch of TV, having a breakfast)
    b. Warning about activities unallowed by doctor
    c. Send message to doctor or nurse about unallowed activity (e.g. drinking or smoking)
    d. Recognize activity as symptom of impairment of health (e.g. toss and turn, cough, fall)
    e. Send message to doctor or nurse about symptom of impairment of health
2. Advising of activity of patient in a ward
    a. Recognize and prediction of activity
    b. Providing information in response to the request from patient (e.g. schedule of treatment, program of TV)
    c. Reminding to patient about necessary activities which are recommended by doctor (autoreminder) (e.g. to apply determined medicine)
    d. Prompting about the requirements of medicine or other objects needed for patient (e.g. book, tooth-brush)
    e. Control of environment for help of activity of patient (e.g. control of lighting, hitting)

The components of smart collaborative objects (SCO) are depicted in Figure 1. It includes sensor devices to make observations of phenomena in the physical world. Sensor measurements are processed by the perception subsystem that associates sensor data with meaning, producing observational knowledge that is meaningful in terms of the applications domain.

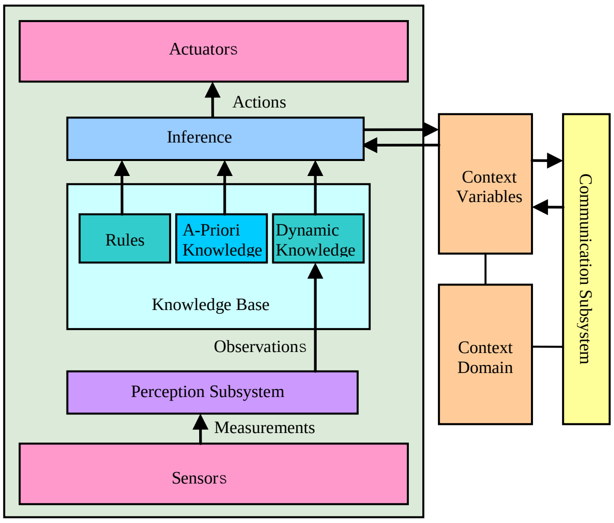

Figure 1. Smart Collaborative Object

Observations are stored and maintained in a knowledge base that reflects the current knowledge of the object about its environment. Perception subsystem may be very simple for simple sensors (e.g. temperature or light sensors) and may be complex based on neural network and other techniques for visual sensor (like camera).

The inference subsystem performs collaborative inference to generate further knowledge using the rules, a-priori knowledge and current knowledge got from perception subsystems and communication subsystem as context variables from another SCO and reasons the actions, context variables and other facts based on the inferred situation.

Actuators are responsible to execute the actions decided by the inference subsystem. Actuators may be generators of sound (or phrases) controller of light-source, motors for any movements.

Domain of context is the description of context variables (CV) been used in Knowledge Base and its role, i.e. facts for condition and facts for conclusion of rules, its location. Context variables are usually depended on location and may be produced by different SCO situated in different places. For example, temperature, lighting or level of sound in different points of hall or corridor, planned meeting in defined room, number of people in room and so on. Sometimes value of context variable may be produced just in one SCO.

Communication subsystem handles all communication-related issues required for smart object collaboration. Cooperating with context domain the communications subsystem adapts the inter object communication to reduce overhead.

Rules may use the context variables for checking of conditions and produce new values of any context variables available either in this SCO or in others. In table 1 some examples of rules are presented.

First three rules are producing actions of SCO. The first rule is dealing with control of local illumination, for example, on table or near bookshelf. Second rule provides control of basic lighting in room. Third rule is dealing with producing of speech phrase in certain situation in room. Forth rule demonstrates forming value of context variable based on other context variables. In this example Boolean context variable "Schedule_is_empty" means that in this room in current time is not planning any arrangement. Fifth rule produces true value of Boolean context variable meaning that patient (in health care system) is outside his ward.

Table 1. Examples of rules on context variables

NN	Condition	Conclusion
1	Lighting = low & Human_is_near	Set_on_light
2	Lighting = on & Room_is_empty	Set_off_light
3	Sound = high & Is_exam & Tearcher_is_absent	Phrase "Hush, please!"
4	Number_of_people = 0 & Schedule_is_empty	Room_is_empty
5	Location = corridor	Patient_is_outside

These examples show that it is needed to connect value of any context variable to determine location in which corresponding rule is executing and context must be concerned to current time.

SCO may be dynamic (mobile) or static. The static SCO is located in one determined place during enough long time. Mobile SCO may be bind with mobile robot, brining objects (cup, book and so on) or hand-held device like PC pocket.

Smart environment is represented as hierarchy of smart collaborative objects. Any SCO in this hierarchy can include set of other smart objects. Example of fragment of such hierarchy is presented in Figure 2. This hierarchy may change in runtime.

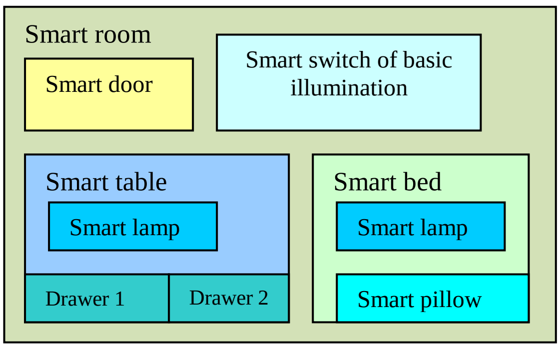

Figure 2. Fragment of hierarchy of SCOs

3. Distributed Context and Perception

In any room or hall several SCOs may produce value of same context variable like temperature or level of lighting etc. So it is important to select which one of them to use corresponding value. For example, a user wants to see the level of temperature using hand-held device. In this case it is worth to get the value from the closest SCO with temperature sensor (Figure 3). Thus we must use the location information of all SCOs generating "temperature" context and the location of hand-held device. This information may be obtained from location estimation system deployed in high level smart objects such as room or building [5]. At that the location of static objects may be estimate preliminary and storing in database and the location of mobile objects are estimated in online fashion.

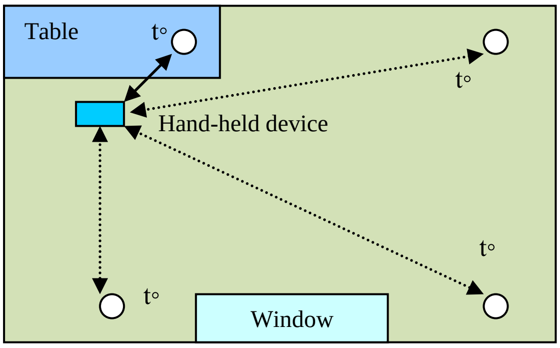

Figure 3. Possible distribution of sources of context variable "temperature" in room

Location estimation of objects may be absolute and relative. Absolute location is determined in coordinates inside high level smart objects. Relative location of object is estimated as relationship with other objects and may be represented by linguistic variables, e.g. *near(table), left_to(table), in_front(door)* and so on. This location estimation most of the cases is provided basically by visual sensors. The usage of such linguistic variables (e.g., *left_to* or *right_to*) demands definition of topological scale, which may be connected with position of camera or current position of SCO that makes decision using the corresponding variables.

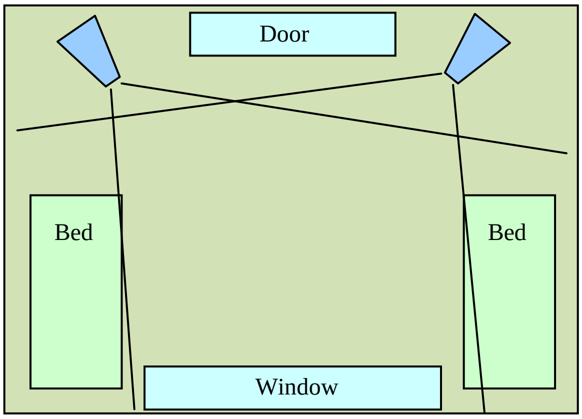

Figure 4 Example of distributed vision in ward

If some SCOs have camera as sensors or one SCO (e.g. room) has several cameras we have opportunity to implement so-called distributed vision [6]. Distributed vision provides possibility to build a more precise scene of a room to estimate an exact relative location (Figure 4). For example, view from different camera in different points ensures visibility of all objects independently on its relative location, capability to track movement of human being in room, like in [6], more exactly estimate relative location of objects.

In case of distributed vision (figure 4) context variables connecting with cameras participating in it are produced by higher level SCO than SCO including these cameras as sensors. Such context variables may be following: "target of movement of patient = bed" (prediction or hypothesis), "in hand of patient = toothbrush", "expression on face = sad". These context variables must be produced by SCO "room" but not SCO or SCS "eye" (including one camera).

4. Communication in Smart Collaborative Objects

Individual objects can only make limited observations of their environment, mainly due to intrinsic limitation of attached sensors and available perception algorithms. As a consequence objects need to communicate to reason about changes in the environment.

We propose a collaborative communication platform based on context-sharing and cross-object reasoning. Objects take decision based on context-values and context-dependencies. If a local decision need to take i.e. the interested contexts are available in local space then no communication is required. But if a common decision need to take as the contexts are distributed over objects inter object communication is necessary. It is therefore a key decision for the communication subsystem to decide which contexts can be obtained from which object. To reduce the strength of communication we propose to parameterize it by the respective contexts. The communication subsystem uses a backward-chaining algorithm with choice points and meta-information about contexts to decide which objects should cooperate in communication. Depending upon the situation a forward-chaining technique also can be used.

In case of context delivery to reduce the communication overhead we propose to adapt context delivery scheme proposed in [8] where the communication is divided into multicast and unicast channel. The hot context and cold context are sent in multicast and unicast channel respectively.

5. Conclusion

In this paper we present the model of smart collaborative objects that can share the responsibilities of backend server by local processing and collaborating with other objects. We also develop the technique to adapt the local inferring using the distributed knowledgebase. We also propose a communication platform for collaborative object to reduce communication overhead and thus save enormous energy. Another additional advantage of our scheme is that locally processing at each sensing entity gives us the benefit of handling each sensor in its own specific way which will aid in resource conservation on part of the smart object.

6. Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement).

References

[1] Siegmeund, Frank: '*Cooperating Smart Everyday Objects – Exploiting Heterogeneity and Pervasiveness in Smart Environments'*: PhD dissertation, Swiss Federal Institute of Technology Zurich (ETH Zurich), 2004

[2] G. Cabri, L. Leonardi, M. Mamei, F. Zambonelli, "Location-dependent Services for Mobile Users", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems And Humans, 33(6):667-681, Nov. 2003

[3] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "*Connecting the Physical World with Pervasive Networks*", IEEE Pervasive Computing, 1(1):59-69, January 2002.

[4] Maomao Wu, Adrian Friday, Gordon Blair, Thirunavukkarasu Sivaharan, Paul Okanda, and Hector Duran Limon. Novel Component Middleware for Building Dependable Sentient Computing Applications. Proc. 0f ECOOP'04, 2004.

[5] Uzair A, A Gavrilov and SY Lee. *Modular Multi Layer Perceptron for Location Awareness.* 2006 IEEE World Congress on Computational Intelligence. 2006 July, 16-21, 2006, Vancouver, BC, Canada. – Pp. 3465-3471.

[6] Atsushi NAKAZAWA, Hirokazu KATO and Seiji INOKUCHI. *Human Tracking Using Distributed Vision Systems*. Proc. of 14th Int. conf. on Pattern Recognition ICPR-98, August, 1998, Australia, Vol.1. – Pp. 593-596.

[7] Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt:' Gaia: *A Middleware Infrastructure to Enable Active Spaces*.' In IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.

[8] Hung Q. Ngo, Anjum Shehzad, Kim Anh Pham, Maria Riaz, Saad Liaquat, and S. Y. Lee, "Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework". The 2004 International Conference on Embedded & Ubiquitous Computing (EUC2004), Springer-Verlag, Lecture Notes in Computer Science, pp.672-681.

[9] Lenin Mehedy, Sungyoung Lee, Salauddin Muhammad Salim Zabir and Young Koo Lee, "*Scalable and Adaptive Context Delivery Mechanism for Context Aware Computing*", The Special Issue of the International Journal of Pervasive Computing and Communications - JPCC-2007, 2007.

[10] Lenin Mehedy, Md. Kamrul Hasan, Young-Koo Lee, Sungyoung Lee, Sangman Han: Hybrid Dissemination Based Scalable and Adaptive Context Delivery for Ubiquitous Computing. The 2006 IFIP Int. Conf. on Embedded and Ubiquitous Computing, 2006, August, 01-04, Seoul, Korea, LNCS 4096 – 0987.

[11] Takeshi Okadome, Takashi Hattoi, Kaoru Hiramatsu,and Yutaka Yanagisawa, Project Pervasive Association: Toward Acquiring Situations in Sensor Networked Environments. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.3B, March 2006. – Pp. 134-139.

[12] Marcelo Kallmann and Daniel Thalmann. A Behavioral Interface to Simulate Agent-Object Interactions in Real Time**.** Proceedings of the Computer Animation. 1999. – Pp. 138-145.

[13] Jérôme Maisonnasse , Nicolas Gourier, Oliver Brdiczka, Patrick Reignier and James L. Crowley. DETECTING PRIVACY IN ATTENTION AWARE SYSTEM. 2nd International Conference on Intelligent Environments. Athens, July, 2006. – Vol. 2, Pp. 231-239.

[14] Martin Strohbach, Gerd Kortuem, and Hans Gellersen. Cooperative Artefacts — A Framework for Embedding Knowledge in Real World Objects. Workshop on Smart Object Systems, in Proc. of UBICOMP'2005, Tokyo, 2005.

[15] Albrecht Schmidt, Martin Strohbach, Kristof van Laerhoven, and Hans-W. Gellersen. Ubiquitous Interaction - Using Surfaces in Everyday Environments as Pointing Devices.

[16] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, and J.A. Stankovic**,** "An Advanced Wireless Sensor Network for Health Monitoring," at *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, Arlington, VA, April 2-4, 2006.

[17] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, A. Wood, "Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges," in *High Confidence Medical Device Software and Systems (HCMDSS) Workshop*, June 2-3 Philadelphia, PA, 2005.