A SYSTEMATIC METHODOLOGY FOR AUGMENTING QUALITY OF EXPERIENCE IN SMART SPACE DESIGN

JING ZENG, LAURENCE T. YANG, HUANSHENG NING, AND JIANHUA MA

Jing Zeng is with Huazhong University of Science and Technology.

Laurence T. Yang is with Huazhong University of Science and Technology and St. Francis Xavier University.

Huansheng Ning is with the University of Science and Technology Beijing.

Jianhua Ma is with Hosei University.

ABSTRACT

Smart space is gaining popularity in academia and industry due to its deep influence on humans, computers, and physical objects. However, quality of experience has attracted less attention in the design of smart space, which results in poor user satisfaction in actual design solutions. In this article, we propose a systematic method for augmenting smart space design, which allows the refinement of high-level user needs into the underlying design solution by leveraging an intermediate representation model, meeting nearly optimal user satisfaction including locality satisfaction, energy consumption satisfaction, QoS satisfaction, and human computer interaction satisfaction. Also, we present the design platform of smart space, which achieves the design flow combined with the space model and platform library. Finally, we use a home health care application to demonstrate the effectiveness and feasibility of our methodology.

INTRODUCTION

A smart space is a physical space equipped with sensors, actuators, and pervasive devices. It allows proactive service leveraging information and communication technologies to be provided in order to meet users' computing needs according to context information [1]. Smart space design has gained intensive attention in academia and industry; a detailed survey for smart space can be found in [2]. Smart space design involves various research issues; quality of experience (QoE) of smart space is a critical issue that significantly affects the future design of smart space.

QoE of smart space refers to the factor of human impact on design and is associated with user perception, experience, and expectations of application and network performance [3]. Unfortunately, most existing research [4, 5] is aimed at securing applications and developing specific application functions. Few current investigations in the design of smart space consider the user QoE, which poses various challenges. First, sensors, actuators, and routers are installed in improper places, which results in poor user experience in the design solution of smart space. Second, high energy consumption may increase user power cost, which also lowers user experience. Third, users' expectations of QoS are also significant, so they should be seriously considered (e.g., price, latency, and availability).

To address the challenges, we propose a systematic design methodology for augmenting smart space design. Leveraging an intermediate representation model (IRM), we can achieve nearly optimal QoE design for a given platform library (the set of reusable computing components) and constraints. The design is a mapping process from user defined specification to underlying architectures. Multiple QoE criteria are considered in the design procedure, including locality satisfaction, energy satisfaction, QoS satisfaction, and human-computer interaction (HCI) satisfaction.

THE SYSTEMATIC DESIGN METHODOLOGY

Our design methodology is motivated by platform-based design [6], which advocates that the design process can be abstracted as the exploration in the design space with given constraints. As Fig. 1 shows, in our design framework, initially, the space specification is captured by the IRM; then the software defined specification is mapped into the design platform of smart space leveraging the IRM. Essentially, the mapping process can be abstracted as the multi-objective combinatorial optimization (MOCO) problems to be solved. Also, the IRM can formally represent the smart space flow relations with control flow and corresponding physical flow, data flow, and human flow. The design platform of smart space consists of multiple platform libraries, and we also take the space model and constraints into consideration. Hence, the mapping is further interpreted as constrained optimization problems, where they encompass object placement, system synthesis, and HCI synthesis.

The object placement achieves space objects allocation including physical objects and cyber objects to guarantee user maximal locality satisfaction. The system synthesis dynamically

 1536-1284/15/$25.00

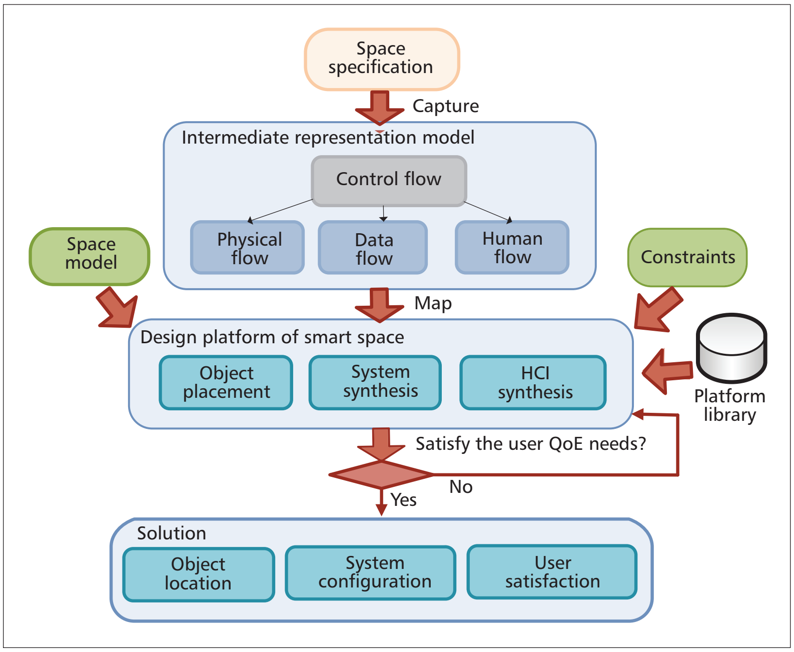


Figure 1. The design framework for smart space. Initially, the space specification is defined according to the application needs. Subsequently, an IRM captures the specification, exhibited by the control flow that corresponds with data flow, physical flow, and human flow. Furthermore, the specification is mapped into the design platform of smart space leveraging the IRM, jointly with space model and constraints. Eventually, the design solution exploration process is iterated until it satisfies the QoE needs of users.

explores the computation and communication resources, meanwhile ensuring maximum energy and QoS satisfaction. The HCI synthesis involves maximizing human satisfaction at the HCI interface design. Eventually, we can attain a set of design solutions by solving the optimization problems. However, the solution set probably cannot satisfy the users' QoE needs, but the solution process allows continued iterations to yield the solution. The final output includes smart space object locations or layout, system configuration incorporating selected computing platforms, and network solution. Also, user satisfaction values are also illustrated for these generated solutions.

Space Specification

Space specification is a high-level language that gives a unified description of humans, computers, and physical objects within a smart space. A specification language usually should incorporate the description of structure and behavior.

Regarding the structure, the objects in a smart space are divided into human actor, cyber actor, and physical object, respectively. The human actor involves human objects in a smart space. Typically, the human actor is able to send and receive messages leveraging the HCI interface. For example, individuals can interact with the environment via a text, haptic, acoustic, or visual manner. Each human actor is characterized by an identity and an HCI interface. The cyber actor is partitioned into sensor, actuator, and computation node. Each cyber actor owns an input interface and an output interface. The physical objects illustrate the objects in the physical environment, where they are sensed or controlled by cyber actors. Accordingly, each defined physical actor associates with a cyber actor. The above basically defines the language's structure. Furthermore, we describe the behavior of the language. We utilize the predefined state and event to specify the behavior within a smart space. The behavior interaction among humans, computers, and physical objects can be abstracted as the state transition and event trigger.

Specifically, we extend the language in [7], which is a compiler for smart space. However, it merely gives the definition of cyber part. A smart space should holistically consider humans, computers, and physical objects. Therefore, we take the three classes of objects as a whole; the event trigger in a smart space is closely related to the state transition of the three types of objects.

The Intermediate Representation Model

The space specification is merely a coarse-grained description of smart space needs. Furthermore, we need to transform it into an underlying architecture. To model smart space, there are three critical issues that need to be focused. The first is how to effectively specify the state transition and event trigger among the three classes of objects in the space specification. The second is how to capture their relations with a unified and consistent model. The third is that the model should be expressive enough and be able to adapt to the mapping needs of the bottom platform. To address these tricky problems, as Fig. 1 demonstrates, we present a flow-based model that allows the formal capture of the relations among human actor, cyber actor, and physical object. Also, it consists of control flow, physical flow, data flow, and human flow.

The flow-based model, in essence, is based on graph theory and Petri net [8, 9]. The former is used to model the cyber-physical, cyber-cyber, and cyber-human relationships, typically corresponding to physical flow, data flow, and human flow, respectively. The latter, associated with control flow, is applied to capture the state and event defined by the space specification.

We define that physical flow, data flow, and human flow are directed acyclic graphs. The physical flow shows that each physical object in a smart space associates with a cyber actor, indicating that the physical object is able to be sensed or actuated by cyber actors (sensors, actuators, or computation nodes). The data flow serves to model the messages sent between cyber actors. Here, they can be seen as processing elements, which cope with input messages and yield output messages. According to the data flow, we can automatically allocate the computing and network resources to satisfy the application needs of a smart space combined with specific design constraints. The human flow depicts the interaction between human and cyber actors. Furthermore, the HCI interface is able to be determined leveraging the human flow.

To model the state and event of the specification, we employ Petri net [8] as the representation of control flow that allows the concurrent nature of smart space applications to be effectively captured. In actual applications, Petri net typically uses place to denote the state and transition to indicate the event, and the directed edge to express their relations [9]. Also, Petri net provides a mathematical approach to analyze the system behavior and performance, thus paving the way for our design optimization. To relate the three classes of flows, each place in Petri net associates with an arc at the physical flow, data flow, or human flow. In other words, the interaction between the cyber actor, human actor, and physical object is controlled via Petri net or control flow.

Design Platform of Smart Space

The platform concept has been emerging for some years. Nonetheless, the definition of a platform is usually determined or interpreted depending on the application domain (e.g., software design, embedded system design). However, few existing works adopt the platform-based approach to conduct the design of smart space. We define the design platform of smart space as an extensible, resilient, and reliable infrastructure that allows the software-defined space specification to be defined into underlying architecture. The platform consists of a platform library, design constraints, and a space model. The design process is the exploration of the platform library according to given constraints. Concretely, the mapping from the specification to the platform is to deposit the function in the device platform while meeting the given performance criteria, such as cost, energy consumption, and user preference. Otherwise, the space model serves as the object placement model within smart space, which largely influences the design performance (e.g., user location preference, communication energy consumption, and device installation cost).

Platform Library: The platform library encompasses various types of libraries. Here, we define the computation library and communication library. The computation library includes multiple computation platforms in particular, the performance requirements of the application determine whether the function defined in the specification is implemented via software or hardware. For example, the software implementation characterizes with generality but high energy consumption, while hardware, such as field programmagle gate arrays (FPGAs) and application-specific integrated circuits (ASICs) is more energy-efficient but has higher development cost. The communication library is made up of the network technology and communication devices (Bluetooth, Zigbee, WiFi, etc.). All of them are described by the price, energy consumption parameter, or other user defined parameters.

Constraints: We use a variety of design constraints to assist the exploration of the design space. In a real application scenario, more complex applications need stronger constraints. For our smart space design, there are multiple aspects of constraints. For instance, a computation task requires memory, longer execution time, network energy consumption, and delay and QoS needs. Besides, the space's geometry constrains the device and object placement. Users themselves can also define constraints according to design needs.

Space Model: Not only is designing smart space related to computational devices and communication networks, but the space model also needs to be investigated. The space model [10] can be classified into two classes: physical-location-based and symbolic-location-based. The former uses the geometry coordinates to denote the object location, represented as longitude, latitude, and height. Specifically, the distance of two objects can be attained by calculating their Euclidean distance. The symbolic location model depicts the object model with abstract symbols. The symbolic-based space model can be divided into graph-based, set-based, hierarchy-based, and so on. The symbolic model is appropriate for humans to read, but cannot calculate the distance directly. In our design methodology, we combine them to solve the space location allocation problem. In high-level software defined specification, we use the symbolic model to express object location together with the graph-based model. The object location will be further refined into specific space coordinates at the design platform. We take the object's physical location as a set of discrete location spaces. However, even though the real space is successive, in the actual application scenario there is only limited location space that can be adopted due to the physical geometry constraints or user interaction comfort needs. Therefore, the object placement problem in smart space can be abstracted as an exploration problem in the limited location space via calculating their Euclidean distance. Technically, the exploited evaluation object can be user locality satisfaction or user defined function.

Design Flow

The design flow can also be called the mapping process from the software defined specification to our design platform. Basically, it refers to multiple design decision problems. Specifically, leveraging the IRM, we can accomplish the design performance estimation of smart space, including design cost, energy consumption, QoS, and so on. Subsequently, their user satisfaction value can be attained according to our defined satisfaction function. Therefore, we join the design decisions into combinatorial optimization problems. The design decisions of smart space need to focus on three aspects: object placement, system synthesis, and HCI synthesis. Object placement is to optimize object locality satisfaction; system synthesis is to optimize the energy and QoS satisfaction in computation and network devices according to the design needs of application; and HCI synthesis is to optimize the QoE in HCI interface.

Object Placement: Object placement aims to deposit objects defined in the specification in the physical space. As a matter of fact, the objects are physical objects and cyber actors. In the actual application, our candidate location set is the discrete location space, which is represented by the

The flow-based model, in essence, is based on graph theory and Petri net [8, 9]. The former is used to model the cyber-physical, cyber-cyber, and cyber-human relationships, typically corresponding to physical flow, data flow, and human flow, respectively. The latter, associated with control flow, is applied to capture the state and event defined by the space specification.

Due to the distributed nature of smart space, we need suitable network technology to support the communication between tasks. To satisfy the experience in energy and QoS requirements, similarly, we also utilize the IRM to capture the communication needs of smart space.

coordinates (longitude, latitude, height). According to the approach in [11], we define the locality satisfaction, which is a function of user comfortable interaction zone Z_c, location of user L_u, and location of object location L_o, as follows:

$$\phi(Z_c, L_o, L_u) = \begin{cases} 1 & \text{if} \quad L_o \in Z_c \\ 0 & \text{if} \quad L_u \in Z_c \\ (0,1) & \text{otherwise} \end{cases} \tag{1}$$

Hence, for the object set O in the specification, the object placement is eventually abstracted as the combinatorial optimization problem. Locality satisfaction L_S is defined as

$$L_S = \sum_{o \in O} \phi(Z_c, L_o, L_u) \quad \forall o \in O \tag{2}$$

We solve the maximum Eq. 2, which can be solved by an available optimization tool such as SAT4J [12]. Hence, we can get each object location in physical space.

System Synthesis: Depending on the software-defined specification, system synthesis conducts the computation and communication resource allocation to enhance user experience on QoS and energy consumption. Technically, system synthesis consists of computation synthesis and communication synthesis. To address the architecture design, the computation synthesis chooses appropriate software or hardware platform from our design platform of smart space to enable the task defined in the specification. The communication synthesis automatically yields a network to satisfy the task distributed communication at a given QoS.

Here, we proceed with the computation synthesis leveraging the IRM, which allows us to perform system-level estimation of the cost and energy consumption for space specification. Hence, the evaluation object function is user energy satisfaction with the decision selection of computation synthesis, which can be computed via our proposed graph-based model. Each actor in the specification is associated with the computation platform in the platform library. For smart space application, there are a wide range of system platforms that can be used to develop the function component. However, they differ in cost and energy consumption or other performance parameters. Essentially, computation synthesis seeks the trade-off between them. For example, for signal processing or complicated mathematical operation, a digital signal processor (DSP) or FPGA is more energy-efficient. However, for complex algorithm, the general-purpose processors or software implementations are more cost-effective.

Due to the distributed nature of smart space, we need suitable network technology to support the communication between tasks. To satisfy the experience in energy and QoS requirements, similarly, we also utilize the IRM to capture the communication needs of smart space. Accordingly, the communication synthesis can be a MOCO problem. Besides, the protocol, topology, and routing are the decision variables determined by our communication synthesis. Multiple wireless (Bluetooth, Zigbee, WiFi, etc.) or wired (Ethernet, ARCnet, etc.) network protocol stacks can be used to synthesize the network. The topology can be star-based, ring-based, tree-based, or mesh-based, depending on the specific application. Also, the objects (sensors, actuators, computation nodes) in smart space usually do not have routing capacities. Hence, routers are allocated into the space to assist with data forwarding. The numbers, locations, and types of routers are decision variables determined by the communication synthesis results.

Hence, energy satisfaction needs to take computation and communication energy jointly into account, which can be calculated according to the proposed IRM. We define the energy satisfaction as follows:

$$\delta(u, s) = \begin{cases} 0 & \text{if} \quad E_s > E_u \\ 1 & \text{if} \quad E_s - E_{avg} < 2\sigma \\ 0.5 - \dfrac{E_s - E_{avg}}{4 \cdot \sigma} & \text{otherwise} \end{cases} \tag{3}$$

where E_u denotes the user expectation to the generated solution, E_s denotes the energy consumption of the generated solution (computation and communication), E_{avg} is the mean value of multiple users' expectation energy, and σ is the standard deviation of energy consumption. Furthermore, various QoS parameters can be considered as constraints added in computation and communication synthesis. Assume that QoS has N dimensions. Each dimension could be price, availability, latency, and so on. The QoS satisfaction of solution s is defined as

$$\kappa(s) = \frac{1}{N} \sum_{i=1}^{N} \omega_i \cdot \phi(i), \qquad \sum_{i=1}^{N} \omega_i = 1 \tag{4}$$

where ϕ_i is the user satisfaction of the ith dimension of QoS, and ω_i is the weight coefficient.

HCI Synthesis: HCI synthesis exhibits the human preference for interaction manner. Diverse users differ in their interaction preference for HCI. For example, the disabled possibly favor voice or visual interaction; youngsters pursue a haptic manner; and conventional offices are more inclined to use texts. HCI synthesis is to ensure the maximum user satisfaction; that is, a user preference value as an evaluation object enables the design decisions, which can be the HCI manner, device manufacturer, and so on. To qualify the user preference, we use the model in [13]. The HCI preference value is grouped into eight important levels. Each level denotes the user satisfaction for the decision, and its importance level is a value ranging from –1 to 1. In real application, we collect user preferences through a GUI, and subsequently the answers of users can be automatically mapped into a software-defined specification.

Evaluation Object: Based on design flow, we take locality satisfaction, energy satisfaction, QoS satisfaction, and HCI satisfaction as the evaluation object of user QoE. For simplicity, locality satisfaction can be considered as an independent step to be optimized. Moreover, energy satisfaction E_s, QoS satisfaction Q_s, and HCI satisfaction H_s are defined as a triple-objective combinatorial optimization problem. In actual application, QoS parameters can include substantial performance criteria. To simplify the complexity of optimization, they can be seen as constraints added in the optimization procedure. The triple-objective optimization problem can be denoted as $\max\{E_s, Q_s, H_s\}$, where the value of energy satisfaction and QoS satisfaction can be calculated according to Eqs. 3 and 4, while HCI satisfaction can be calculated leveraging the approach above.

Case Study

To demonstrate the feasibility and effectiveness of the proposed methodology, we employ home health care as a case to examine. Home health care is a promising application in smart space, which allows assisting in the daily life of the elderly. A typical application of home health care is fall detection [14], which provides automatic fall detection for the elderly by cameras and computation nodes, and then sends alarms via actuators. We assume that an aging person lives in a house as shown in Fig. 2, where the house covers 9 m × 14 m. The house comprises a kitchen, a bedroom, a living room, and a corridor. To accomplish home health care, a variety of cameras, alerters, actuators, and distributed computation nodes should be allocated in the smart space.

Typically, the computing tasks of the fall detection application defined in the specification have five aspects:

1. Segmentation (PE1): This task preprocesses the image data sampled from the synchronous cameras, and to attain the foreground region.
2. Tracking (PE2): This task tracks the objects in the scene, and marks them as human and non-human.
3. Feature extraction and head tracking (PE3): For each object that is not marked as non-human, its direction features and variance ratio are captured according to the views of a video sensor. Subsequently, the head position can be tracked based on skin color.
4. Fall detection (PE4): Relying on the feature extraction from step 3, a fall detection algorithm is capable of enabling judgement of the fall.
5. Control task (PE5): The control node sends the affirmation to the user, and if the user does not respond or times out, a control command is sent to the actuators to send an alarm or call an emergency number.

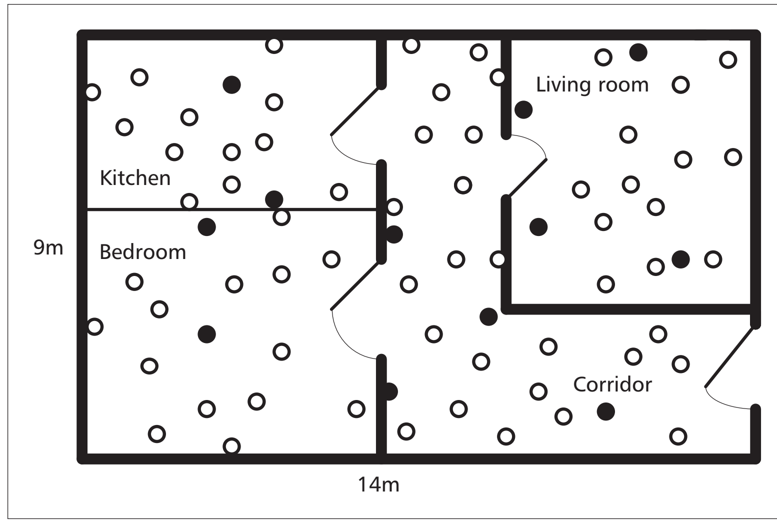
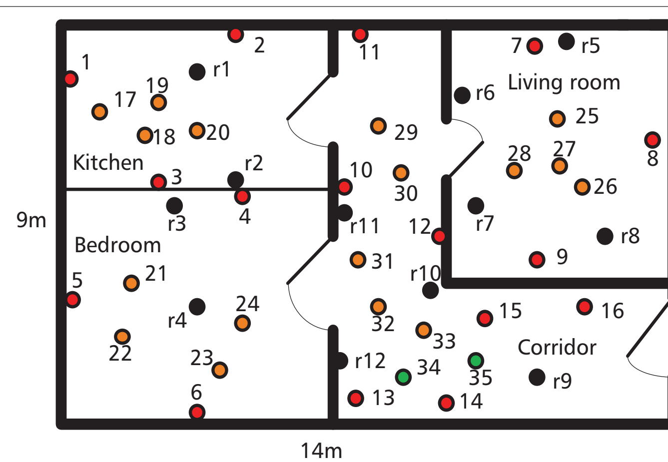


Figure 2. Floor plan of a smart home. Black dots denote the candidate position set of routers, while white dots represent the candidate position set of physical objects and cyber actors.

According to our design approach, the candidate location set is illustrated in Fig. 2, which is given by the design platform of smart space. First, the object placement is performed. We solve the location preference optimization problem based on SAT4J [12] and get the maximum locality satisfaction, 8.0. Cameras, alerters, and computation nodes are deposited into the candidate location set, as depicted in Fig. 3b. Each node is indexed with a number, denoting the actor index in Fig. 3a.

Index	Index
1–16	Camera
18, 21, 27, 31	PE1
17, 22, 26, 30	PE2
19, 23, 25, 29	PE3
20, 24, 28, 32	PE4
33	PE5
34, 35	Alerter

(a)



(b)

Figure 3. The result of object placement: a) the corresponding index for actors; b) the result of object placement from Fig. 2. The red dots are cameras, the green dots are alerters, the orange dots are computation nodes, and the black dots are wireless routers.

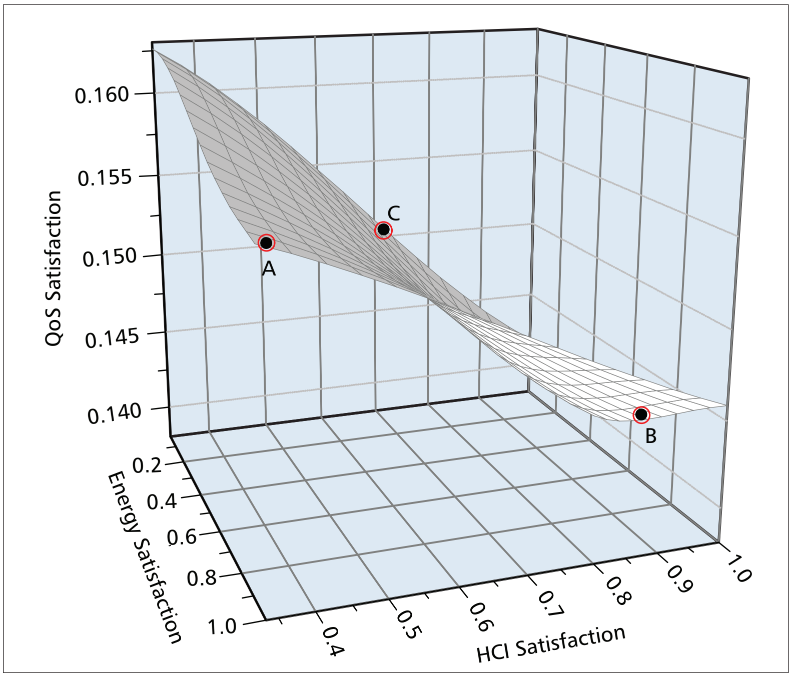


Figure 4. The final result of the design space exploration.The curved surface denotes the Pareto front with Pareto optimal points. The points at the upper side of the curved surface are dominated by the Pareto point on the curved surface. Points A, B, and C are selected as the result solution for comparison.

Second, for the system synthesis, we employ Xilinx Spartan 6, Actel IGLOO2, Altera Cyclone III, ARM Cortex-A5, LPC4350, and MPC5125 as the candidate computation platform, where their performance parameters on energy consumption and price can be gained from their product data sheet. For communication synthesis, to keep simplicity, we take IEEE 802.11a as an example to investigate the communication synthesis. The topology is based on a cluster tree, and we take the design price as the main QoS evaluation, and the rest of the QoS parameters (latency, bandwidth, deadline, etc.) are used as constraints added into the optimization. Furthermore, a candidate router location as demonstrated in Fig. 3b is marked with symbol r. The numbers of routers and location selection are determined by the optimization results. Also, for HCI synthesis, we consider four types of HCI interfaces: textual, acoustic, visual, and hybrid interaction.

To proceed system synthesis and HCI synthesis, we solve the MOCO problem with the NSGA-II algorithm and the available MOEA framework [15]. Configuration parameters are set as follows:

- Population size = 2000
- Mutation probability = 0.000001
- Bit crossover probability = 0.9

In Fig. 4, the curved surface is the Pareto front generated by our design algorithm, which is plotted by 300 non-dominated solutions. In particular, energy satisfaction, QoS satisfaction, and HCI satisfaction are computed leveraging the proposed IRM to perform the system-level estimation. We indicate the energy consumption as the power dissipation at the least common cycle of the home health application. Cost is the total price of enabling the home health care application. User preference value is qualified by the method in [13]. Theoretically, we transform the maximum triple-objective optimization into a dual problem to be solved.

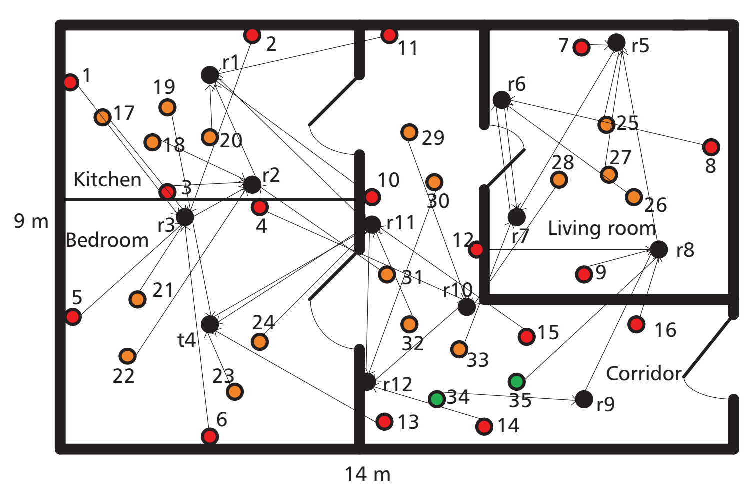
To investigate our generated solution, we select points A, B, and C in Fig. 4. The results are indicated with a computation solution, a network solution, and an HCI interface solution. For point A, each computation actor in Fig. 3b corresponds to a computation platform, demonstrated in Fig. 5a. Figure 5b illustrates the link state of the wireless network, which consists of the connections of end devices and routers. The direction of the arrows indicate that the nodes point to their parent nodes. The communication solution is automatically generated by exploration in design space. Similarly, points B and C can also be denoted in the same manner. Clearly, point A favors much higher QoS satisfaction and energy

Computation synthesis result

HCI interface manner
Textual

Actor index	Platform index
17, 18, 21, 22, 25, 26, 29, 30	Xilinx Spartan 6
19, 27	Actel IGLOO2
23, 31	Altera Cyclone III
20, 24, 28	ARM Cortex-A5
32	LPC4350
33	MPC5125

(a)



(b)

Figure 5. The generated design solution of point A by our proposed method. a) denotes the result of computation synthesis; b) denotes the result of communication synthesis.

satisfaction, but it has lower HCI satisfaction embodied with the user HCI interface preference value, while point B can enable higher energy satisfaction and HCI satisfaction than A. However, it is lower in QoS satisfaction. Point C has good QoS satisfaction and HCI satisfaction, but has relatively poor performance in energy satisfaction. Actually, in the real scenario, users can further choose the appropriate solution according to their expectation range of these criteria.

Conclusion

The smart space design refers to multiple technology challenges for improving QoE. In this article, we use a systematic approach to solve the design problem of smart space. An automatic design decision can be fulfilled leveraging our proposed design approach. It also enables the near optimization of various QoE criteria by top-down refinement.

We use a systematic approach to solve the design problem of smart space. An automatic design decision can be fulfilled leveraging our proposed design approach. It also enables the near optimization of various QoE criteria by top-down refinement.

References

[1] J. Ma *et al.*, "Towards a Smart World and Ubiquitous Intelligence: A Walkthrough from Smart Things to Smart Hyperspaces and UbicKids," *Int'l J. Pervasive Computing and Commun.*, vol. 1, no. 1, 2005, pp. 53–68.

[2] N. Zhong *et al.*, "Research Challenges and Perspectives on Wisdom Web of Things (W2T)," *J. Supercomputing*, vol. 64, no. 3, 2013, pp. 862–82.

[3] M. Garcia *et al.*, "A QoE Management System for Ubiquitous IPTV Devices," *Proc. 3rd Int'l. Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies*, Oct. 2009, pp. 147–52.

[4] H. Zhu *et al.*, "SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks," *IEEE Trans. Vehic. Tech.*, vol. 58, no. 8, Oct. 2009, pp. 4628–39.

[5] H. Zhu *et al.*, "A Probabilistic Misbehavior Detection Scheme toward Efficient Trust Establishment in Delay-Tolerant Networks," *IEEE Trans. Parallel Distrib. Sys.*, vol. 25, no. 1, Jan 2014, pp. 22–32.

[6] A. Sangiovanni-Vincentelli, "Quo Vadis, SLD? Reasoning about the Trends and Challenges of System Level Design," *Proc. IEEE*, vol. 95, no. 3, 2007, pp. 467–506.

[7] U. Bischoff and G. Kortuem, "A Compiler for the Smart Space," *Proc. Ambient Intelligence Euro. Conf.*, vol. 4794, 2007, pp. 230–47.

[8] J. Zeng *et al.*, "HyperspaceFlow: A System-Level Design Methodology for Smart Space," *IEEE Trans. Emerging Topics Computing*, 2015.

[9] L. Popova-Zeugmann, *Time Petri Nets*, Springer, 2013.

[10] L. Tang *et al.*, "Situation-Based Design: A Rapid Approach for Pervasive Application Development," *Proc. 9th Int'l. Conf. Ubiquitous Intelligence Computing* and *9th Int'l. Conf. Autonomic Trusted Computing*, Sept 2012, pp. 128–35.

[11] Y. Xu *et al.*, "An Intensive Location-Aware Framework for Device-Involved Human Tasks," *Proc. IEEE 10th Int'l. Conf. High Performance Computing and Commun.*, Nov 2013, pp. 2135–42.

[12] D. Le Berre *et al.*, "The Sat4j Library, Release 2.2, A System Description," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 7, 2010, pp. 59–64.

[13] H. Mukhtar, D. Belaïd, and G. Bernard, "Dynamic User Task Composition Based on User Preferences," *ACM Trans. Autonomous Adaptive Sys.*, vol. 6, no. 1, p. 4, 2011.

[14] M. Mubashir, L. Shao, and L. Seed, "A Survey on Fall Detection: Principles and Approaches," *Neurocomputing*, vol. 100, no. 0, 2013, pp. 144–52.

[15] Moea framework: http://www.moeaframework.org

Biographies

Jing Zeng is currently a Ph.D. student in the School of Computer Science and Technology, Huazhong University of Science and Technology, China. His research interests include ubiquitous computing, system-level design methodology, and smart space.

Laurence T.Yang (ltyang@gmail.com) received a Ph.D. degree in computer science from the University of Victoria, Canada. He is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology, and in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include smart space, embedded and ubiquitous computing, and parallel and distributed computing.

Huansheng Ning is a professor in the School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include the Internet of Things, and electromagnetic sensing and identification.

Jianhua Ma is a professor and head of the Department of Digital Media in the Faculty of Computer and Information Sciences at Hosei University, Japan. His research interests include multimedia, ubiquitous computing, social computing, and cyber intelligence.