A home service deployment platform with support for detection and resolution of physical resource conflicts

Marios Sioutis, Junsoo Kim, Azman Osman Lim, Yasuo Tan
School of Information Science, Japan Advanced Institute of Science and Technology
Asahidai 1-1, Nomi, Ishikawa
Email: {smarios, junsoo, aolim, ytan}@jaist.ac.jp

***Abstract*—Services in the home environment compete for computational and physical resources. In the event of a resource conflict, the service deployment platform usually resolves it by suspending a number of services to allow the remaining services execute successfully. For the case of physical resource conflicts, we argue that a compromising solution which allows parallel execution of conflicting services successfully may exist. We propose a system that uses novel compromising algorithms and the notion of "Area of Effect" to detect and resolve such conflicts, without the need to suspend any running services. Based on the experimental results we conclude that the proposed system and algorithms are a promising solution for the problem of physical resource management.**

I. Introduction

Advances in interoperability between devices and standardized network protocols have made it possible to envision a new role for the home environment as a deployment platform for home services provided by external service providers[1]. Such a platform would allow services to access the resources of the home environment for their purposes.

In such a scenario, services conflict over the use of resources. However, unlike traditional operating systems that manage computational resources, a system of this nature has to also manage physical resources as well: temperature, illumination, humidity, sound and noise levels.

In a case of resource conflict, previous algorithms[2] suggest the suspension or the termination of a process based on priority (e.g. user priority, interface priority, service priority, e.t.c.) to at least allow one of the conflicting services to continue successfully. Furthermore, due to the nature of the physical resources, traditional resource management algorithms such as time sharing cannot be adapted to handle such conflicts in a meaningful way.

To overcome this limitation, we propose the notion of "Area of Effect" (AoE) for the detection and the use of compromising algorithms for the resolution of physical resource conflicts in the home environment. The notion of physical resource conflicts first appeared in [5]. A model checking approach for the detection of conflicts was conducted in [3], [4].

We argue that in the case of physical resource conflicts between services, usually a meaningful compromise can be reached that will allow the services to continue parallel execution with relative success (i.e. without a perceivable sharp degradation of the user experience quality).

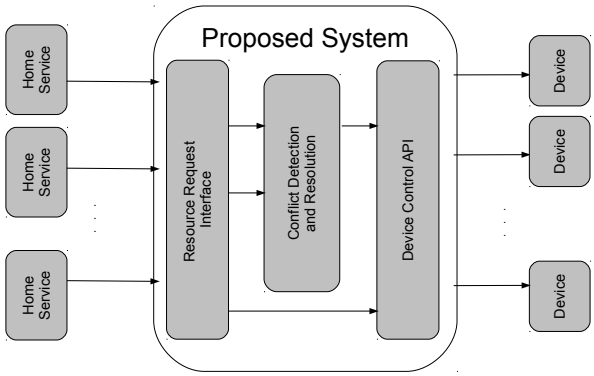

Fig. 1. Proposed system architecture

The proposed system is composed by three parts: a resource request interface to allow services to make requests, a conflict detection and resolution part that utilizes area of effect and compromising algorithms and finally a device control API, used to control devices in the home environment. Furthermore, the system is assumed to have absolute control over the devices and maintains real-time spatial information(e.g. user/object positions). The overall architecture can be seen in figure 1.

II. System Characteristics

A. Resource Request Interface

A major component of the proposed system is the resource request interface. Through this interface, any service will be able to make requests for resources, physical or computational. Part of the ongoing research involves the development of a sufficiently expressive and general programming interface for any resource type.

The prototype system offers high-level functionality such as the ability to set the intensity of a physical property over a given physical area (e.g. setting room temperature, adjusting the illumination around a user). The task to find appropriate device settings to fulfil such a resource request is left to the system.

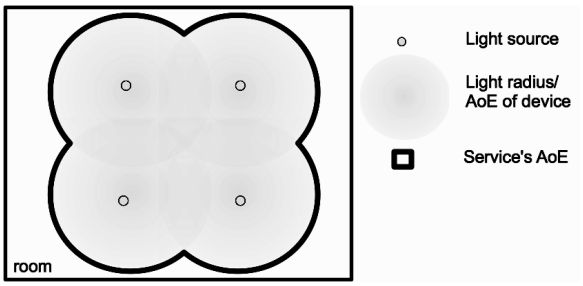

Fig. 2. Area of Effect

This approach has two major advantages. First, a service that targets this platform is shielded from the details of the home environment, thus it can be employed as-is in other houses that use the same platform. Secondly, such a high-level programming interface simplifies and reduces the time necessary for the development of home services.

A typical resource request regarding a physical property codifies information about the intensity of that property in terms of *thresholds*. Three types of thresholds exist: *lowest*, *highest* and *exact* which correspond to a minimum desired intensity, a maximum desired intensity and an exact intensity. Furthermore, to codify the area over which the desired intensity is to be applied a combination of an *anchor point* (a point of interest) and a distance can be used. Alternatively, whole rooms may also be used.

B. Area of Effect

The concept of AoE of a service can be intuitively understood as the physical space over which the service wants to impose a preferably consistent intensity regarding a physical property (figure 2). In the prototype system, AoE can be expressed using a combination of a point of interest (an "anchor point") and a distance from this point. The anchor point may be a moving target.

Such information is passed to the system whenever a request for a physical resource is made. The system, using this information, will attempt to find an optimal solution that satisfies the request. Information about the area of effect of a service is also used during the conflict discovery step performed by the system. A solution is a set of devices and device settings that will best fulfil the request made by the service.

The area of effect is closely related to the devices and settings used in a solution. From this information, an estimation of overlapping AoEs can be made. In its simplest form, if the set of devices used to fulfil a request overlap with the set of devices used to fulfil another request and the overlapping devices have conflicting settings, it is a good indicator that a conflict has occurred. In more sophisticated scenarios where no overlapping use of devices occurs, the system will first search for any service whose request cannot be fulfilled and then search for devices whose settings hinder this by estimating their effect at the anchor point of that service.

C. Compromising Algorithms

To solve physical resource conflicts, two types of algorithms are proposed: space-based conflict resolution and intensity-based conflict resolution algorithms.

Space-based resolution algorithms try to reduce the area that is under conflict. Such algorithms are usually effective in illumination scenarios, where even in the same room the physical property can be partitioned (as shown in the demonstration section).

A basic assumption used in both of the proposed space-based algorithms (explained in section III-A) is that, during the evaluation of a prospective solution, points that are closer to the anchor point are considered to be of higher importance compared to points that are further away. From this assumption, devices that are further away from the anchor point of a service usually end up being less important for the fulfilment of a request. This kind of heuristics can be used to determine the sequence with which the solution space will be explored.

The space-based algorithms usually are iterative in nature. In each iteration a single change is made to the settings of one of the conflicting devices. If this change improves over the previous solution (or even the initial state) it is considered a success and the algorithm continues to the next iteration. If the new solution is actually worse, the algorithm may choose to backtrack and explore other options.

Intensity-based resolution algorithms try to reach a compromise regarding the intensity of a given physical property. Such algorithms are effective when the physical property tends to retain a homogeneous intensity over a relatively large area. Temperature and humidity are perceived to be such physical properties. For example, it may be possible to partially fulfil two conflicting requests for temperature settings by settling for an intermediate temperature intensity.

To evaluate a prospective solution, various strategies exist (two of which are explained in section III-A). These strategies may prioritize different characteristics of a solution thus, depending on the strategy used different solutions will be obtained.

D. System Decision Making Flow

Events such as the arrival and expiration of a resource request or a significant change of an anchor point position will trigger the conflict detection and possibly the conflict resolution steps that the system performs.

For each new resource request arrival, the system will attempt to find an ideal solution that fulfils it without taking into consideration any other possibly conflicting requests that may be active at the time. The devices that will be used as well as their settings, are marked as part of the solution. For requests that have a highest or lowest threshold, finding an optimal solution is straightforward: set the surrounding devices to their minimum or maximum setting accordingly. For requests that have exact thresholds finding an optimal solution is more difficult.

During the conflict detection step, the system searches for requests that have overlapping AoEs as described in section

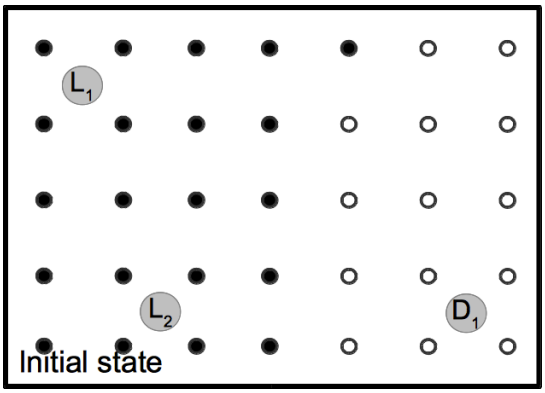

Fig. 3. Initial device assignment

II-B. The system in its final form will make estimations of the AoEs that are based on accurate simulation of physical laws and sensor input if available.

The last step is the conflict resolution step. Finding an optimal compromising solution can be treated as an optimisation problem and the search strategy of the solution space depends on the algorithm itself.

III. Experiments and results

A. Experiment set up

The following scenario was simulated as a part of the experiments conducted to evaluate the prototype system. In a room 5 meters wide and 7 meters long, 35 spot lights are affixed to the ceiling in a grid pattern with an equal distance from each other. Each spot light provides an on/off control interface. They shed light in a conical fashion and their individual effect at a given point in space is modelled with a simple physics engine. The illumination measured from a distance of 1 meter from a spot light is 1000 lumen, further decreasing as the distance from the source increases according to the inverse square law.

Three services with anchor points L_1, L_2 and D_1, as seen in figure 3, make requests regarding the illumination of the room. Services with anchor point L_1 and L_2 make a request for a bright setting with a *lowest* threshold of 1000 lumen whereas service with anchor point D_1 make a request for a dark setting with *highest* threshold of only 10 lumen. The requested area of effect for all three requests is the whole room.

The system receives the requests for resources and decides on an optimal solution for all three requests. For the two bright setting requests the system decides that the best solution is to switch on all the lights in the room, whereas to fulfil the request for a dark setting the best solution is to turn off all the lights in the room.

The conflict detection step takes place. The system detects that for each illumination device three settings (two of which are conflicting) have been proposed. The system deduces that the three services are conflicting and enters the conflict resolution phase.

In the conflict resolution phase, the system performs an initial assignment of devices to the services, as seen in figure 3. Each device is assigned to the service with the closest anchor point. The four different cases that where examined all use this initial device assignment as a potential solution that will be improved upon.

Two variations of space-based algorithms where used. In the ”greedy” algorithm, in each iteration the service with the lower score gains control of a device that is closest to its anchor point. In the ”forfeit rights” algorithm, the service with the highest score voluntarily yields control of its least important device in hopes that it will improve significantly the score of some other service. Both algorithms have a one-level ”roll back”: if the change of the settings of an illumination device is to deteriorate the quality of solution, the setting for that device will not change and a different device will be selected. The algorithms stop when no other alternative devices that could be examined exist.

To calculate the score for each service formula 1 was used. I_{Est} is the estimated intensity for the service and I_{Req} is the actual requested illumination intensity. This formula takes into consideration the logarithmic nature of human perception. Furthermore, if the threshold for a service request was met, then the score will be positive, else it will be negative. Moreover, I_{Est} is the average of the illumination intensity taken from 8 points that form a circle with radius of 75 centimetres around the anchor point.

$$score = \begin{cases} 10\log_{10}\left(\frac{I_{Est}}{I_{Req}}\right) & \text{, if threshold is met} \\ -10\log_{10}\left(\frac{I_{Est}}{I_{Req}}\right) & \text{, if threshold is not met} \end{cases} \quad (1)$$

To evaluate the quality of the solutions proposed, two different evaluation strategies were used. The simple ”highest score” strategy favours solutions that produce the highest total score (i.e. the summation of all the scores of each service is maximal). The second evaluation strategy is the ”constraint programming” strategy. This strategy favours solutions with minimal ”badness”. The badness of the solution is the sum of all violated constraints i.e. requests that could not be fulfilled. Violated constraints always have a negative evaluation score, thus the ”badness” of a solution is always a negative number (or zero).

B. Obtained results and discussion

The linear combination of the above algorithms and evaluation strategies yields a total of four cases. The ”greedy” algorithm was used in cases 1 and 2. The ”forfeit rights” algorithm was used in cases 3 and 4. Furthermore, the ”highest score” evaluation strategy was used in cases 1 and 3, whereas in cases 2 and 4 the ”constraint programming” evaluation strategy was used.

The final solutions (device states) of the simulated scenario can be seen in figure 4. Depending on the evaluation strategy and the compromising algorithms, different end states where proposed as solutions (cases 1 to 4) which have different

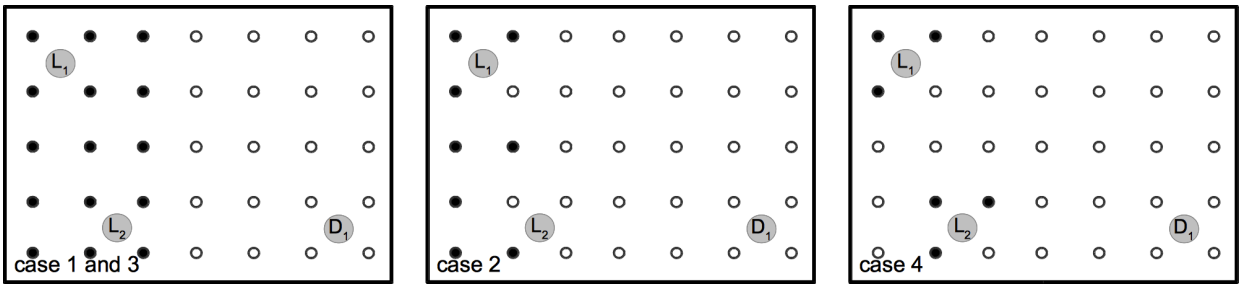

Fig. 4. End states - four possible solutions

characteristics. Any of the four solutions is better than the alternative of suspending one of the three services.

In cases 1 and 3 the lights that are located on the left side of the room will be switched on. The solution for these cases was dominated by the "highest score" evaluation strategy, with the two algorithms ending up proposing the exact same solution. Although the total score of this solution is the highest, the solution greatly punishes the request of service D_1 whose requested threshold was not honoured. The user experience for the user at D_1 would be dissatisfying.

Cases 2 and 4 are more interesting, as the two algorithms proposed quite different solutions. For these scenarios the "constraint programming" evaluation strategy was used. In case 2 the "greedy" algorithm keeps on "seizing" control of lights one by one on behalf of service D_1. However, quite a few critical lights around L_2 where switched off, something that reduced the score for that service. The greedy algorithm was not able to improve any further on this solution.

Case 4 is of particular interest as it managed to fulfil the requests of the services L_1 and L_2 whereas at the same time it achieved the best result for service D_1 compared to all other scenarios. The "forfeit rights" algorithm makes sure that the very important lights close to L_1 and L_2 would be the last to be switched off, if at all. This way, the algorithm is able to aggressively improve the solution and switch off all unneeded lights. An unintended side effect is that in this scenario a minimal amount of devices was used, thus achieving energy efficiency.

The scores for each service, the total score and the "badness" of each solution can be seen in table I. Case 4 fares better in terms of "badness", thus encompassing the spirit of compromising solution. Although in cases 1 and 3 the total score is higher, this solution severely hampers service D_1, whose constraint was violated by a wide margin. Solution 2 fares less favourably to the other cases, as it does not excel to any of the metrics used in this test.

Test cases	1, 3	2	4
L_1	2.58	0.78	0.06
L_2	3.35	-0.57	0.07
D_1	-8.40	-4.40	-4.19
$Badness$	-8.41	-4.96	-4.19
$TotalScore$	-2.48	-4.19	-4.05

TABLE I
SUMMARY OF RESULTS

IV. CONCLUSION AND FUTURE WORK

In this paper we introduced the basic concepts of a home service deployment platform that acts as a resource management system with support for detection and resolution of physical resource conflicts. The system introduces the notion of "Area of Effect" and the novel idea of compromising algorithms to allow the parallel execution of conflicting services, which is a preferred alternative to algorithms proposed so far. Finally, a space-based algorithm scenario is used to demonstrate the concept of compromising solutions, yielding promising results.

As future work, more research must be conducted, especially to deal with *exact* threshold requests (i.e. thresholds that specify that the intensity should be as close as possible to the one requested). This problem is similar to a discrete knapsack problem and sufficient care must be taken to ensure good solutions. Furthermore, the scenarios presented in this paper were simulated. In future revisions of the system, different scenarios should be tested in a real experimentation environment, that takes into consideration data available from sensors for better accuracy and validation of solutions. Finally, a systematic user evaluation of the solutions produced by the system must be pursued to ensure the validity and the applicability of such solutions in real life scenarios.

REFERENCES

[1] Next Generation IP Network Promotion Forum, Home Network WG, *Next generation home networks and the challenge towards a new age of value* (original text in Japanese), pp. 58-60, July 2009

[2] Pattara Leelaprute, *Resolution of Feature Interactions in Integrated Services of Home Network System*, Proceedings of Asia-Pacific Conference on Communications, 2007

[3] L. du Bousquet, *Feature Interaction Detection using Testing and Model-checking - Experience Report*, World Congress in Formal Methods, France, 1999

[4] M. Calder, A. Miller, *Using SPIN for Feature Interaction Analysis - A Case Study*, Model Checking Software 8th International SPIN Workshop, Canada, 2001

[5] A. Metzger, C. Webel, *Feature interaction detection in building control systems by means of a formal product model*, Feature Interactions in Telecommunications and Software Systems VII, pp.105-122, 2003