

# Graph-based Approach for Interference Free Integration of Pervasive Applications

Christophe Soares<sup>\*†</sup>, Rui S. Moreira<sup>†\*</sup>, Ricardo Morla<sup>\*</sup>, José Torres<sup>†</sup> and Pedro Sobral<sup>†</sup>

*\*INESC TEC, FEUP. University of Porto, Porto, Portugal.*

*Email: {cpsouares, rmorla}@inescporto.pt*

*†ISUS Unit, FCT. University Fernando Pessoa, Porto, Portugal.*

*Email: {csoares, rmoreira, jtorres, pmsobral}@ufp.edu.pt*

**Abstract**—Smart-space modeling is a fundamental requirement towards the achievement of safe integration of pervasive systems and devices. Traditional behavior models lack explicit representation of rich semantic for ubicomp systems. We propose a graph-based model supporting ubicomp’s computational, physical, user and temporal contextual information. The model is part of a solution to tackle the interference problem, which results from unexpected interactions involving users and entertainment, communication, and health-related devices. This is a major concern when, for instance, daily routines are affected by different types of applications. The proposed graph-based interference detection is integrated in the Safe Home Care reflective platform, and allows reifying system’s state and simulating home care scenarios. A set of home care scenarios are used to assess the applicability and correctness of our approach.

**Keywords**-Feature interaction, interference-free, graph-based interference pruning, Safe Home Care, reflective middleware.

## I. INTRODUCTION

Numerous technologies exist and have been proposed that can improve home care. These are mostly health monitoring and drug-related devices that can be acquired independently of most supermarket’s shelves to enrich the growing panoply of smart devices used at home, e.g., in communications, entertainment and support for daily tasks. These devices are developed independently, bought off-the-shelf (OTS), and likely deployed directly by homeowners without any coordination. This leads to a potential set of unplanned interactions and unwanted behavior, which is referred to as feature interaction (FI) and interference, i.e., unwanted behavior resulting from unplanned interactions. This issue is critical especially concerning home care and health-related devices.

In our previous papers, we present a 3D simulation framework [1], which will be used to generate system state data; and we propose a first approach to represent and analyze in realtime the behavior of an OTS system using graphs [2]. In this paper, we extend this approach for detecting unwanted behavior in home care introducing relations between independent systems during their execution through a media (i.e., akin to physical, user, time, history and location ubicomp’s context). This is based on a graph-traversal algorithm that uses observed and expected system state. System-specific

state information is introspected through physical sensors such as temperature, noise, and brightness sensors and application APIs (e.g., ongoing VoIP call, empty drug dispenser, etc.). In this setting, the premisses/conditions are that system’s behavior in the smart space (SS) were previously known and that the sensors collecting the observed data are not subject to any noise. We test our interference detection algorithms with different interaction scenarios between the OTS systems present in the SS. We describe the integration of our approach in the Safe Home Care (SHC) reflective framework, which is being developed to simulate, analyze, manage and deploy interference-free home settings.

The paper is organized as follows: Section II, reviews related work on FI in pervasive environments and cross-reality systems. Section III describes the project SHC and its architecture. Section IV explains the graph-based approach followed in SHC for tackling FI. Section V presents increasingly complex use cases applied on evaluating and comparing intra-system FI detection with different approach. Finally, in Section VI, we present final remarks and discuss future work.

## II. RELATED WORK

Jakkula et al. [3] propose a methodology to avoid interference in smart environments. They record normal user interaction patterns with the SS. Those patterns are then compared with the observed user behavior. The interference detection is based on a probability model that matches the expected and observed user actions. For example, if the user does not open the refrigerator all day, as usual, is considered an abnormal behavior and should be reported. Kolberg et al. [4] propose a solution to ease the integration of independent systems in an intelligent space. They detect resources (e.g., environmental sensors or actuators) that can be shared between systems. The intelligent space manages concurrent access to resources using system priorities and protocol interworking techniques to avoid interference. This approach focuses on interactions between different systems, the user is not considered as a possible source of interference. Prendinger et al. [5] present a bi-directional interface between a physical Sensor-Based System and a 3D virtual space (Second Life). This middleware layer (cf.

Twin-World Mediator) is used to reflect and reify changes and is able to identify problems and interferences related with spatial requirements of devices in the SS. Similarly to [3] and [4], we manage to capture the state of the system and its applications. However, contrary to these systems, we use a graph representation that implicitly represents state sequences, i.e., system’s behavior (cf. computing and user ubicomp’s context). Moreover, we additionally gather information about the environment and the user perception, also implicitly represented in the graphs. Furthermore, akin to [5], our architecture also proposes the reification of base-level SS; yet, our meta-level combines the use of a 3D simulation framework with graph-based state representations; this combination allows the analysis and detection of FI between independent built OTS systems.

Majuntke et al. [6] propose a framework for managing FI between multiple pervasive applications that share the same physical environment. They use a coordinator not only for monitoring a shared medium, applying locks and timeouts, but also for managing the concurrent access to that medium. Therefore the coordinator must interact with existing OTS systems, introducing synchronization delays and possibly becoming a source of interference. Initially, SHC will only observe in real-time the behavior of all OTS systems and will not interfere directly with the systems, e.g., using direct notifications to the user.

### III. THE SHC SYSTEM

In this section, we first present the major requirements behind the Safe Home Care (SHC) reflective middleware system and then describe the implemented architecture.

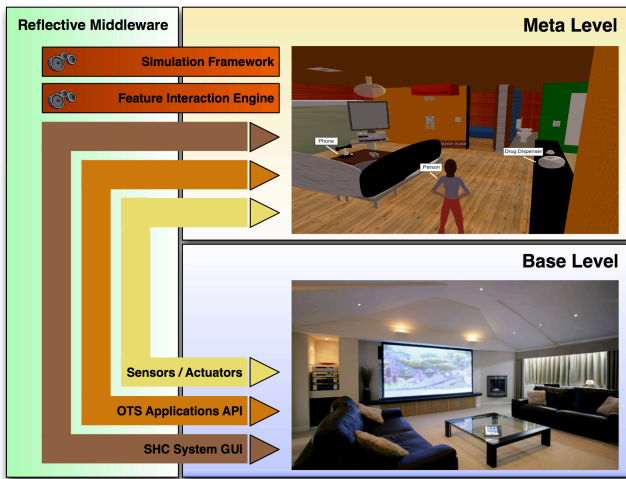


Figure 1. Safe Home Care System - Architecture

#### A. Requirements and Goals

The presented graph-based approach (Section IV) is a core component of the SHC reflective framework (cf. FI Inference

Engine depicted in Figure 1). This middleware tool focuses on building interference-free home care environments, in two possible ways: i) before deployment: use a 3D virtual meta-model to simulate an home environment [1]; then, exploit the simulation outcomes to detect a priori FI between independently developed appliances; ii) after deployment: use sensing mechanisms at run-time to reify the state of installed OTS systems (cf. introspection); then, employ the meta-model representations to detect a posteriori the cause of feature interactions and react by interacting with users or directly adapting the base-level applications (cf. reflection) to help keeping the home environment safe. Next, we present the overall SHC architecture.

#### B. System Architecture

The SHC architecture is organized in two levels: base-level and meta-level, which are causally connected through reflective components (see Figure 1). These components interact with the physical environment through: sensors/actuators, available OTS systems APIs (e.g., Asterisk control API) and SHC user interface (UI) facilities. Basically, these components provide the current status for OTS system state reification into 3D Meta-Level (e.g., medicine dispenser alarms, VoIP calls, TV state, human activity, etc.) and reflection back to Base-Level for interacting with user or control the environment and OTS systems. The Meta-Level uses a 3D virtual environment to represent the home care setting and a data model for storing information about the user, environment and appliances. For example, we have programmed the behavior of numerous primitives to represent several OTS systems deployed at home: drug dispenser (DD), VoIP system (Phone), TV system, Home Automation systems, environment and the person as well (User). For each system we store its location, state changes and also the source element responsible for triggering each change (e.g., when the DD element triggers its alarm it causes a state change in the Ambient sound, which in turn, might trigger the attention of a nearby User). The 3D home setting was built on top of OpenSim, an open source platform for virtual worlds. This 3D representation allows non-intrusive real-time monitoring of the home environment; more importantly, it allows us to prepare a semi-random simulation agenda and generate several outcome scenarios (as described on Table I and II). All state changes are time stamped and stored to be analyzed by the FI engine. State sequences are represented using a graph format (cf. GraphML) detailed in Section IV-A. The FI engine was developed in Java and uses system state sequences to detect FI detailed in Section IV-B.

The entire SHC system runs embedded on a media center connected to the television, since, we believe, this was a ubiquitous/omnipresent device, in the future of our household. We currently use a linux media distribution running on a traditional computer.

#### IV. FEATURE INTERACTION DETECTION

Graph representations are well understood and provide a flexible representation for state sequence transitions. We use a graph representation to model behavior of independently deployed OTS systems. We propose two approaches based on graphs: i) intra-system: modeling per-system correct state sequences; ii) inter-systems: capturing possible interactions between OTS systems through shared medium (e.g., environment light or temperature influenced by independent systems, window controlled by autonomous systems, etc.). This paper describe the intra-system FI detection, evaluated throughout several scenarios. We also introduce the inter-system FI approach, which will analyze the dependencies between independent OTS systems (cf. Figure 2a) and will allow us to identify FI causes, i.e., searching for all the online interactions occurrences from the OTS system on a set of medium. We explicitly model shared medium using "links" dependencies between state transitions of different OTS systems. We expect later to be able to identify the cause of FI.

##### A. Graph Representation

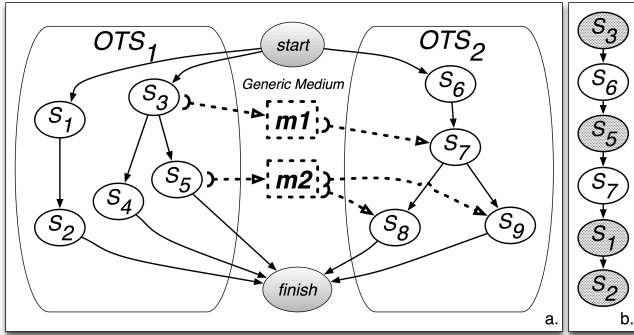


Figure 2. Generic example of FI detection approach: a) Graph of Expected States - model behavior & interactions - and b) Graph of Observed States - online behavior observation

Directed graphs can be used to capture both the expected behavior of isolated systems and the observed behavior of combined systems. Figure 2a shows the expected state sequences for a toy example; Figure 2b depicts the state transitions perceived through sensors and available system APIs.

Assuming that each system has a set of well-known state sequences, it should be possible to represent this sequence through a directed graph. The state of an element is characterized by its feature values. Each graph node represents a unique element state, i.e., a value change on any feature. The expected behavior of each deployed system can be captured into a state transition graph that we name Graph of Expected States (GoES) (see Figure 2a). This approach is extensible to several OTS systems and facilitates the addition of new element graphs or state sequences.

We then, for manipulation purposes, assemble a single graph with common start and finish nodes from all given OTS system graphs. During system runtime or OpenSim simulation we capture the history of state transitions of all OTS systems, i.e., their Graphs of Observed States (GoOS) (see Figure 2b). The GoES represents how systems should behave without interferences and the GoOS represents the current/observed system behavior. In Figure 2a, all possible state path sequences are:  $\langle S_1, S_2 \rangle$ ,  $\langle S_3, S_4 \rangle$ ,  $\langle S_3, S_5 \rangle$ ,  $\langle S_6, S_7, S_8 \rangle$  and  $\langle S_6, S_7, S_9 \rangle$ . We consider also that states  $\langle S_3 \rangle$  and  $\langle S_5 \rangle$  will affect, respectively, medium  $m_1$  and  $m_2$ . States  $\langle S_7 \rangle$  and  $\langle S_8 \rangle / \langle S_9 \rangle$  will be affected, respectively, by medium  $m_1$  and  $m_2$ . Each GoES could be provided by the OTS system manufacturer or installers, or even built from statistical in-site observations. The User's GoES represents only important activities (e.g., exercising, sleeping, take pill, etc.) that can be perceived through sensors, activity agendas, etc.

##### B. Intra-system Approach

Our intra-system approach has two main steps: i) classify – creates a graph with observed OTS system states; and ii) pruning – based on the knowledge of expected state, identifies unexpected states. For example, assume that the current GoOS sequence is  $\langle S_3, S_6, S_5, S_7, S_1, S_2 \rangle$  (see Figure 2b). Based on these two graphs (Figure 2a and 2b) the pruning algorithm removes the complete GoES path sequences from the current GoOS:  $\langle S_3, S_5 \rangle$  and  $\langle S_1, S_2 \rangle$ . The result set is:  $\langle S_6, S_7 \rangle$  because as expected the  $\langle S_8 \rangle$  or  $\langle S_9 \rangle$  were not observed. Our State Pruning Algorithm (SPA) extracts all expected state sequences from the observed behavior; when the result of the SPA algorithm is not an empty sequence of states then it assumes: i) FI occurs and should be solved; ii) state sequences or malfunctions not captured in the existing GoES, which should be re-drawn. The main goal of the State Pruning Algorithm (SPA) is to identify and eliminate sequences of GoES sub-paths from the GoOS, until there are no more possible sub-paths to prune. Hence, the algorithm filters all expected actions/states that have been properly executed by deployed OTS systems. The SPA acts as a pre-processing tool and allows to detect the existence of per system FI (cf. knowledge discovery process phases) [7].

##### C. Inter-system Approach

We extend the previous approach to support medium-based, we named it the inter-system FI. This should enable us to identifies unexpected behavior or malfunction causes using the pruned results - act as a reasoning tool. This approach consists on analyzing (online or offline) every possible interactions between OTS systems through a specific set of media. These interactions could be possible FI causes. For this particular case (Figure 2a) reusing the previous example at IV-B, since  $\langle S_8 \rangle$  or  $\langle S_9 \rangle$  did not occur we

may follow their common dependencies on  $m_2$ . A reverse-path analysis, based on GoES, allow us to identify that both  $\langle S_8 \rangle$  or  $\langle S_9 \rangle$  are affected by  $\langle S_5 \rangle$ . Hence, we are able to identify that  $OTS_1$  was affecting  $OTS_2$  through the medium  $m_2$ .

## V. INTRA-SYSTEM APPROACH EVALUATION

To illustrate our FI detection intra-system approach, we first have applied it to a simple home care use case deployed in a 3D virtual world. Later, we will describe the performance of this approach on other scenarios. The inter-system approach will not be subject of evaluation on this article since it is currently under development.

### A. Example Scenario

The particular scenario in which we have applied our approach is the following. Like every morning, Mary watches her favorite TV show and at 10 AM the drug dispenser triggers the alarm light and buzzer reminding her it is time to take her medicines. Unlike every morning however, the phone rings just after the dispenser alarm is triggered. She answers the phone and spends some time talking to her friend. In this scenario, we explore two possible outcomes based on the duration of the call and on what happens after Mary hangs up: one where Mary hears the DD and takes her pill (without FI - outcome 1); another where Mary does not hear (user perception/awareness) the DD and misses her pill (with FI - outcome 2), e.g., the DD buzzer timeout period has expired during the phone call. In this assisted-living setting, equipped with a drug dispenser (DD) and a VoIP system (Phone), we simulated information reification based on sensors (e.g., capturing environment raw data) and specific APIs provided by installed OTS systems (see Table I and II for collected information). The SHC reflective system allows to monitor or simulate the home setting and reify the state of OTS systems. We present a GoES with the information that may be introspected through different kinds of inputs, i.e., sensors and APIs (see Figure 3a) for this scenario (used by the pruning algorithm); and two GoOS (see Figure 3b and 3c). We consider that a FI occurs each time a system (user or system) prevents an other system to execute has it have been expected by the manufacturer.

1) *Intra-System FI Detection walk-through:* Table I and II represent a subset of the *state* observation in the 3D simulation framework from the home setting from 10:30 AM to 11:10 AM for both outcomes. GoOS is built from these database records (see Figure 3b without interference and Figure 3c with interference). This step is achieved through a matching; each state corresponds to an expected state. A state has some specific and unique characteristic: belongs to a given system (Element), results from a specific activity (Feature, Value and Type) and is introspected by a particular source (Sensors, API or both). *Outcome 1:* the SPA uses the GoES (see Figure 3a) to

Table I  
OUTCOME 1 - STATE TABLE  
ON SHC DATABASE

Element:Feature	Timestamp
DD:alarm	10:30 AM
DD:buzzer	10:30 AM
U:needs_pill	10:30 AM
PH:call_in	10:31 AM
PH:ringing	10:31 AM
U:receives_call	10:31 AM
PH:call	10:32 AM
U:take_call	10:32 AM
PH:¬ringing	10:32 AM
PH:¬call	10:45 AM
DD:take_pill	10:46 AM
U:medicated	10:46 AM
DD:¬alarm	10:46 AM
DD:¬buzzer	10:46 AM

Table II  
OUTCOME 2 - STATE TABLE  
ON SHC DATABASE

Element:Feature	Timestamp
DD:alarm	10:30 AM
DD:buzzer	10:30 AM
U:needs_pill	10:30 AM
PH:call_in	10:31 AM
PH:ringing	10:31 AM
U:receives_call	10:31 AM
PH:call	10:32 AM
U:take_call	10:32 AM
PH:¬ringing	10:32 AM
DD:¬take_pill	11:00 AM
DD:notify	11:00 AM
DD:¬alarm	11:00 AM
DD:¬buzzer	11:00 AM
PH:¬call	11:10 AM

remove correct state sequences from the GoOS. In this example, the SPA performs four iterations, identifying and removing all GoES sub-sets in GoOS:  $\langle DD:alarm, DD:buzzer, DD:take_pill, DD:¬alarm, DD:¬buzzer \rangle$ ,  $\langle U:needs_pill, U:medicated \rangle$ ,  $\langle PH:call_in, PH:ringing, PH:call, PH:¬ringing, PH:¬call \rangle$  and  $\langle U:receivr_call, U:take_call \rangle$ . All paths have been removed, leaving an empty GoOS, which means that no FI were detected. *Outcome 2:* the first three SPA iterations identify and remove three sub-paths  $\langle DD:alarm, DD:buzzer, DD:¬take_pill, DD:notify, DD:¬alarm, DD:¬buzzer \rangle$ ,  $\langle PH:call_in, PH:ringing, PH:call, PH:¬ringing, PH:¬call \rangle$  and  $\langle U:receivr_call, U:take_call \rangle$ . At the fourth iteration there is no other path subset of GoES in GoOS. Hence, the SPA returns the  $\langle U:needs_pill \rangle$  state since state  $\langle U:medicated \rangle$  was not observed. Our approach successfully identifies an interference, i.e., Mary - the affected "system" - does not take her medicine as expected.

### B. Use Case Scenarios

We have applied our intra-system FI detection approach to different scenarios [8] equipped with diverse systems / appliances (see Table III). For each scenario, we identify possible FI outcomes (see reference [8] for full description of the scenarios).

1) *Scenarios Descriptions and Outcomes:* We have used the 3D simulation features of our SHC reflective framework for generating all the presented scenarios and possible FI outcomes using [1]. The proposed graph-based algorithm was then applied to each of the scenarios to analyze all particular outcomes. In the next Section, we present and discuss the results of our approach.

2) *Analysis of Intra-system approach:* Table III allows us to conclude that we successfully manage to identify FI in Scenarios  $\langle A,C \rangle$ , partially detect FI in Scenarios  $\langle B,D \rangle$  and fail to detect FI in Scenarios  $\langle E,F \rangle$ . In Scenarios  $\langle A,C \rangle$  we are able to detect FI because these resulted in behavior

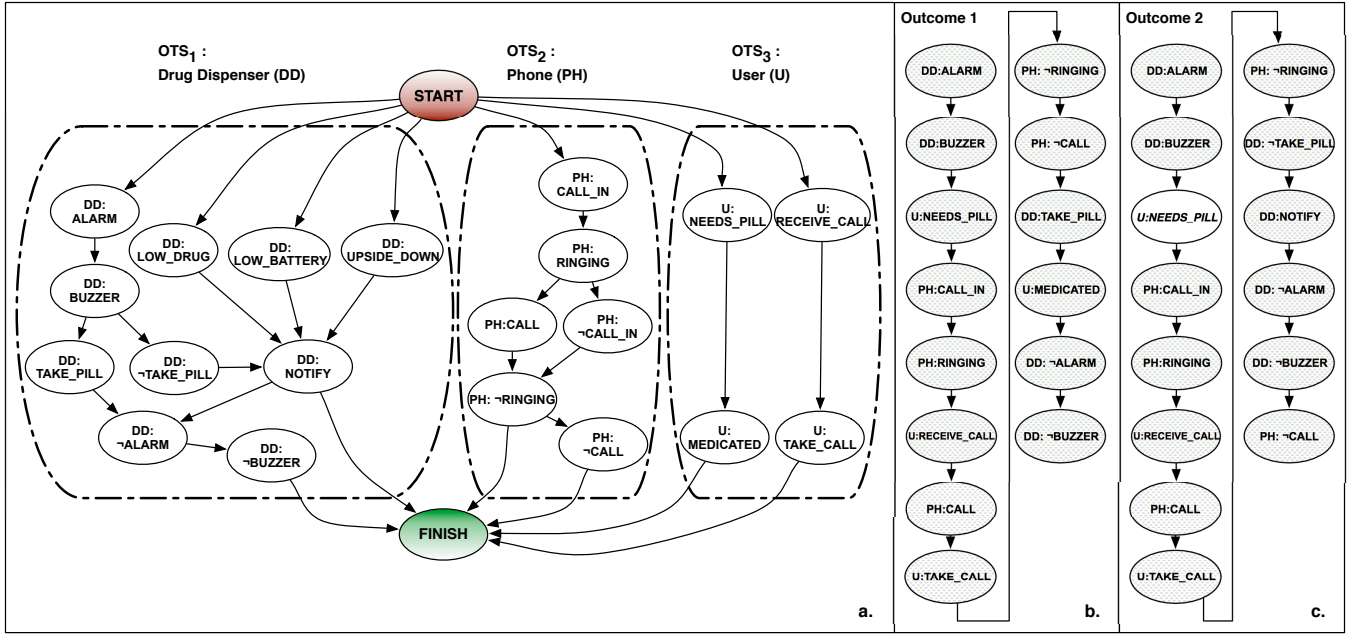


Figure 3. Behavior and Interaction Model for Pervasive Applications: a) Graph of Expected States - GoES and Graph of Observed States - GoOS - b) for outcome 1 without Feature Interaction and c) for outcome 2 with Feature Interaction.

Table III  
EVALUATION THROUGH SEVERAL SCENARIOS

SCENARIO		IDENTIFIED PROBLEM		DETECTION APPROACH	
Use-Case	Outcome	Description	Category	Kolberg [4]	Safe Home Care
SCENARIO A	A1	user does not taker her medicine	user awareness, perception	×	✓
SCENARIO B	B1	user does not hear	user awareness, perception	×	✓
	B2	user does not want to loose TV Show / move to another place	user will	×	✓
SCENARIO C	B3	user looses his TV show	user will	×	×
	C1	user does not hear, sound overlapping	user awareness, perception	×	✓
	C2	user needs to move to another place, interrupt physical practice	user will	×	✓
SCENARIO D	C3	user forgets to resume physical practice	user memory failure, cognition	×	✓
	D1	alarm behavior influenced by HVAC	shared environment between two systems (i.e., window controller)	✓	×
SCENARIO E	D2	alarm is turned (causing insecurity)	user memory failure	✓	✓
	E1	light intensity change (systems conflict)	shared environment between two systems (i.e., light controller)	✓	×
SCENARIO F	E2	user turns lights on but the media-center dims them down	shared environment between two systems (i.e., light controller)	✓	×
	F1	false alarm generated by fire control system	shared environment between two systems (i.e., window controller)	✓	×

that was specified in the graph of expected behavior. For example in the C3 outcome, we identify that Mary stops her physical practice to take a call and after forgets to keep on doing her exercise. We detect the FI because the graphs lack one or more sensed states in the user or on the Microsoft Kinect System. So we assume something unexpected happens. For Scenarios <E,F>, the observed

behavior matches what is specified in the graph of expected behavior even in the outcomes where there is FI. In these particular cases, even when the system is affected by an FI, it continues to behave as expected, preventing us from using this approach to detect this type of problem. The inter-systems approach might enable us, in the future, to detect, reason and solve FI problems due to unexpected interactions

between different OTS systems. Finally, in Scenarios <B,D> we did not have enough information for detecting the FI, since the introspection level used was insufficient. The more information we collect from the real world, the richer will be the state graph representations, hence, the better will be the FI detection. In these scenarios we assume realistic/feasible level information reification. For example, it permits us to access the user agenda to know his medicine prescriptions but does not capture the user will for watching a TV show (see B3 outcomes). But if we are able to sense this kind of data and our introspection level, we may infer the user will (e.g., based on a user profile) then we will also be able to detect FI in these scenarios. Our graph-based solution considers the user, which might be the cause or effect of FI.

In contrast, the Kolberg approach [4] does not consider the user in the FI analysis and does not model the medium of interference. Some outcomes, however, deal with issues that do not affect the expected behavior of the systems; rather, these FI outcomes result from system interactions that occur through a shared medium or variable, i.e., indirect intra-system relationships. The main goal of our inter-system approach consists to tackle out these particular cases and define a modelization, based on graph, to depict OTS system: i) behavior and iii) media interaction - related with physical, time/history, user and location context.

## VI. CONCLUSIONS AND FUTURE WORK

Current health systems will be unable to cope with the significant raise of life expectancy. Hence, new ways must be found to answer the care needs of the population and of elderly people in particular. We argue that home assisted-living settings may relieve the pressure on health systems, e.g., providing non intrusive remote monitoring capabilities, assisting on daily activities, allowing automatic collection of health parameters, endorsing the use of health promoting systems, etc; however, creating these future SS, involves equipping homes with autonomous OTS systems that may not be easily integrated and managed. The proposed SHC reflective middleware was conceived with two goals in mind: i) manage the safe integration of OTS systems (cf. interference-free) by exploiting reflection and 3D virtual world simulation [1]; ii) provide non-intrusive pervasive interface mechanisms for home assisted-living actors. In this paper, we issue specifically the first goal and explore state-graph representations to perceive FI. We demonstrate the pertinence and efficacy of the proposed intra-system approach on different home care use cases. We explore the representation and simulation facilities of 3D virtual worlds for generating OTS systems state-graphs used to study more complex FI scenarios. We are extending our approach for supporting inter-system FI detection over different ubicomp context (location, time, history, user and physical), resolution and providing a better accuracy on the FI detection and FI detection/reasoning. Moreover, our future work will also

focuses on usability tests performed with elderly and also caregivers.

## ACKNOWLEDGMENT

Christophe Soares thanks the FCT - Foundation for Science and Technology, Portugal, for PhD Grant SFRH/BD/64210/2009.

This work is financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project Safe Home Care Project (Interference-free Home Health-Care Smart Spaces using Search Algorithms and Meta-Reality Reflection)/EIA-EIA/108352/2008.

## REFERENCES

- [1] C. L. Velasquez, C. Soares, R. Morla, R. S. Moreira, J. Torres, and P. Sobral, "A 3D Simulation Framework for Safe Ambient-Assisted Home Care," in *The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies - UBICOMM 2011*, Lisboa, Portugal, 2011, pp. 61 – 66.
- [2] C. Soares, R. S. Moreira, R. Morla, J. Torres, and P. Sobral, "Prognostic of Feature Interactions between Independently Developed Pervasive Systems," in *Prognostics and Health Management, 2012. PHM 2012. International Conference on*, Denver, CO, USA, jun. 2012.
- [3] V. Jakkula, D. Cook, and A. Crandall, "Temporal pattern discovery for anomaly detection in a smart home," in *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, sept. 2007, pp. 339 –345.
- [4] M. Kolberg, E. Magill, and M. Wilson, "Compatibility issues between services supporting networked appliances," *Communications Magazine, IEEE*, vol. 41, no. 11, pp. 136 – 147, nov. 2003.
- [5] H. Prendinger, B. Brandherm, and S. Ullrich, "A simulation framework for sensor-based systems in second life," *Presence: Teleoper. Virtual Environ.*, vol. 18, pp. 468–477, December 2009.
- [6] V. Majuntke, G. Schiele, K. Spohrer, M. Handte, and C. Becker, "A coordination framework for pervasive applications in multi-user environments," in *Intelligent Environments (IE), 2010 Sixth International Conference on*, july 2010, pp. 178 –184.
- [7] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data Mining: A Knowledge Discovery Approach*, 1st ed. Springer, Feb. 2007.
- [8] SHHC, "Safehomehealthcare scenarios outcomes: Interference-free home health-care smart spaces," <http://http://homepage.ufp.pt/~csoares/shhc/>, Last accessed July 2011.