Prognostic of Feature Interactions between Independently Developed Pervasive Systems

Christophe Soares*†, Rui S. Moreira†*, Ricardo Morla*, José Torres† and Pedro Sobral†
*INESC TEC, FEUP. University of Porto, Porto, Portugal.
Email: {cpsoares, rmorla}@inescporto.pt
†ISUS Unit, FCT. University Fernando Pessoa, Porto, Portugal.
Email: {csoares, rmoreira, jtorres, pmsobral}@ufp.edu.pt

***Abstract*—Statistics show an aging trend in the world population, which will progressively overload existing health systems. Therefore, we believe that ubiquitous computing will play an important role in domicile settings, coping with the growing need for automated home healthcare support, especially for the sick and elderly. The integration of independently developed off-the-shelf systems (e.g., health-monitoring, entertainment, communications, home automation, etc.) may cause unplanned interactions between them (cf. feature interactions). This is a major concern since the correct/expected behavior of an isolated system may not be the same when deployed in conjunction with other systems, causing interferences, i.e., unexpected outcomes or misbehaviors. The Safe Home Care project tackles this problem to pursuit the safe deployment and reconfiguration of home healthcare smart-spaces. We propose the use of state graphs to represent off-the-shelf systems and predict the occurrence of intra-system's feature interactions. We use pre-deployment simulations to forecast feature interactions before deployment. We assess the applicability and correctness of this approach through a set of simulated home assisted living scenarios.**

***Index Terms*—Feature Interaction, Interference-free, Graph-based Interference Pruning, Safe Home Care, Reflective Middleware, Ubiquitous Computing.**

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, 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 [1]. Unwanted behavior resulting of unplanned interactions is especially concerning when home care and health-related devices are considered. FI was first applied in telecommunication, but as [2] reveal this could also affect others domains, e.g., Ubicomp.

In this paper, we present an approach for detecting unwanted behavior in home care. This approach is based on a graph-traversal algorithm that uses observed and expected system state. State information is introspected through physical sensors such as temperature, noise, and brightness sensors and application APIs that reify system-specific information (e.g., ongoing VoIP call, empty drug dispenser, etc.). We consider different home care scenarios with off-the-shelf (OTS) systems. For each scenario we analyze how our approach is able to detect or not detect interference. We generalize on the applicability of our approach using a set of features of the scenarios. We also describe the integration of our approach in the Safe Home Care (SHC) reflective framework, which is being developed for simulating, analyzing, managing and deploying interference-free home settings.

The paper is organized as follows: Section II describes the graph-based approach followed in the Safe Home Care (SHC) project for tackling feature interaction. Section III describes a simple home care scenario to illustrate our intra-system feature interaction detection. Section IV broadens our approach to several scenarios and analyses the feature interaction detection results. Section V integrates our graph-based approach in the context of SHC framework. Related work on home care, feature interaction in pervasive environments and cross-reality systems is presented in Section VI. Section VII presents the final remarks and future work.

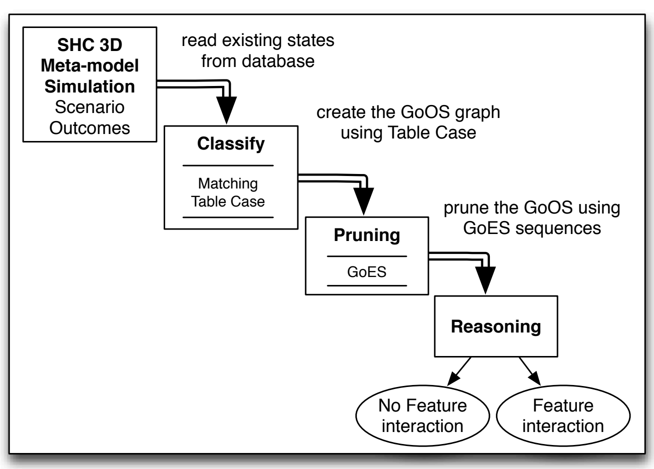

Figure 1. Workflow of the SHC Feature Interaction Engine

This work was supported by ERDF through the COMPETE Programme and by FCT within project SHHC/PTDC/EIA-EIA/108352/2008
Christophe Soares thanks FCT for a PhD Grant SFRH/BD/64210/2009

II. Feature Interaction Detection

Graph representations are well understood and provide a flexible representation for state sequence transitions. We use a graph representation of the state of independently deployed OTS systems. Our approach has three modules, as described in Figure 1: i) classify – creates a graph with observed OTS system states; ii) pruning – based on the knowledge of expected state, identifies unexpected states; and iii) reasoning – identifies unexpected behavior or malfunctions using the pruning results.

A. Graph Representation

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; and Figure 2c reveal the output of the State Pruning Algorithm (see Section II-C). 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. Given a set of state transition graphs, it should be possible to assemble a single graph with common start and finish nodes; this is the Graph of Expected States (GoES) (see Figure 2a). This approach is extensible and facilitates the addition of new element graphs or state sequences. During system runtime or simulation a second graph is built by capturing the actual state history of all elements. This Graph of Observed States (GoOS) reifies the state of selected elements (see Figure 2b). Then we use a SPA to extract all expected state sequences from the observed behavior and detect faulty states.

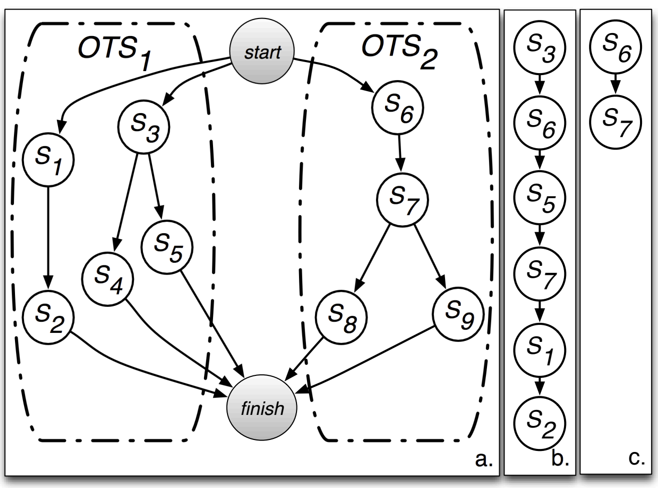

Figure 2. Toy example of the Feature Interaction detection approach: a) Generic Graph of Expected States, b) State Pruning Algorithm Input: Generic Graph of Observed States and c) State Pruning Algorithm Output: Result Graph of Observed States processed by the State Pruning Algorithm

This approach does an independent and per system analysis for detecting intra-system feature interactions. We are currently extending it to consider also interconnectivities between systems, i.e., inter-system feature interactions.

B. Extracting well-known state sequence from data-model

The GoES represents how systems should behave (under proper user utilization, i.e., without interferences) and the GoOS represents the current/observed system behavior. For example, assume that the current GoOS sequence is $<S_3,S_6,S_5,S_7,S_1,S_2>$ (see Figure 2b), all possible path sequences identified in the GoES are: *Off-The-Shelf System*$_1$ - $<S_1,S_2>$,$<S_3,S_4>$,$<S_3,S_5>$ and *Off-The-Shelf System*$_2$ - $<S_6,S_7,S_8>$, $<S_6,S_7,S_9>$.

Based on the two graphs (see Figure 2a and 2b) the pruning algorithm removes the following complete GoES path sequences from the current GoOS: $<S_3,S_5>$ and $<S_1,S_2>$. The SPA ends up without being able to prune every state sequences in the GoOS (cf. $<S_6,S_7>$) thus assuming one of two things: i) there are feature interactions that should be solved; ii) there are state sequences or malfunctions not captured in the existing GoES, which should be re-drawn.

C. The State Pruning Algorithm

The objective of the SPA is to identify and eliminate sequences of GoES sub-paths in the GoOS, until there are no more possible sub-paths to prune (see Figure 3). 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 allowing us to extract some knowledge of our sensing infrastructure (cf. knowledge discovery process phases) [3]. If we use the GoES (Figure 2a) and the GoOS (Figure 2b), the SPA will return the sequence $<S_6,S_7>$ (see Figure 2c). All the other sequence have been pruned, because they happen as expected by the GoES.

III. Example Scenario

To illustrate our feature interaction detection approach, we have applied it to a simple home care use case deployed in a 3D virtual world. 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.

A. Scenarios and Use Case Description

Recent Portuguese statistics [4] drove the construction of a provisional persona that served as a user model in our settings. Mary is 70, lives alone in a small house, and the phone and a TV set are her always-on daily companions. She is autonomous but has some health impairments, which can be mitigated by technology and the help of family and friends: she has breathing and heart problems, has to follow a rigorous therapeutic program (inspire oxygen, perform specific breathing and physical exercises, and take an average of six pills every day). Mary uses the phone to talk to family and

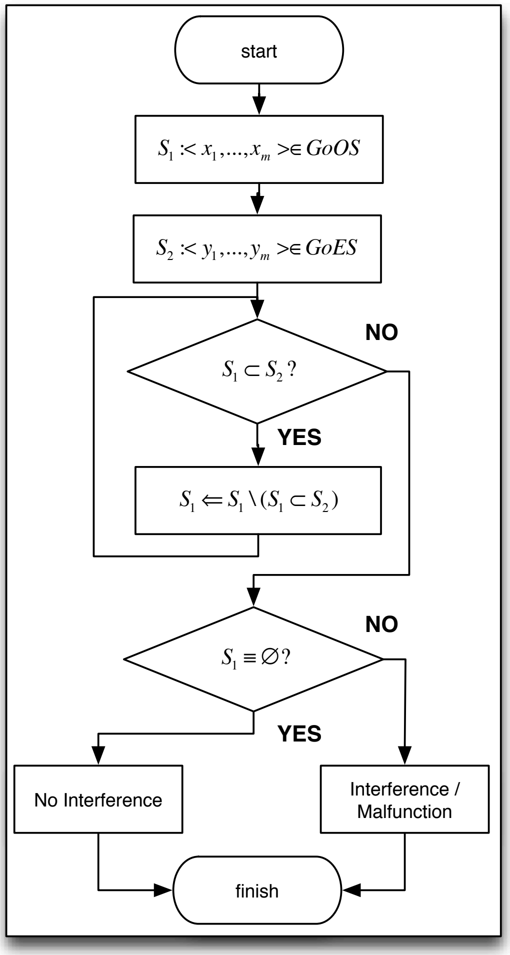

Figure 3. Intra-system Approach for Feature Interaction Detection

friends. Two or three times a week she also receives the visit of her grandson, which helps her organizing things. She has a drug dispenser to help her manage pill intake and a VoIP service to diminish her fixed telephone costs and improve friends presence through video calls. We will consider scenarios using other devices that Mary may have, including: a Vital Jacket [5] to monitor vital signs; an ambulatory system to monitor activity and fall detection [6]; an health promoting system to regulate and stimulate physical activity, like the Xbox 360 Kinect; and also home automation and intrusion detection systems.

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 the DD and misses her pill (with FI - outcome 2).

The SHC reflective system allows to monitor or simulate the home setting and reify the state of OTS systems (see Table I and II). Table III classifies the information that may be introspected through different kinds of inputs (cf. sensors and APIs). We present a GoES (see Figure 4c) for this scenario (used by the pruning algorithm); and two GoOS (see Figure 4a and 4b).

B. *Feature Interaction Detection walk-through*

The feature interaction detection has three steps: i) create the GoOS, i.e., read state values from the SHC database and assemble a GoOS based on the known expected states; ii) prune the GoOS, i.e., use the GoES to remove correct state sequences from the GoOS; iii) identify intra-system FI, i.e., from the state sequences left in the graph recognize missed behavior; iv) identify inter-system FI, i.e., from explicit medium dependencies between OTS system graphs, explore possible interference sources. This last phase is currently under work; for now our focus is on interference detection without identifying the causality. Table I and II represents a subset of the *state* table obtained by simulating the home setting from 10:30 AM to 11:10 AM.

In order to look for interference a GoOS is built from the database records (see Figure 4a without interference and Figure 4b with interference). This is achieved through a matching table (Table III) where each case corresponds to an expected state. A state belongs to a given system (Element), results from its activity (Feature, Value and Type) and is introspected by a particular source (APP, API or both).

1) Outcome 1: The SPA performs four iterations, identifying and removing all GoES sub-sets in GoOS: *<A,B,C,E,F>*, *<Q,S>*, *<K,L,H,P,N>* and *<R,T>*. All paths have been removed, leaving an empty GoOS, which means that no FI were detected.

2) Outcome 2: The SPA uses the GoES (see Figure 4c) to remove correct sequence states from the GoOS. In this example, the first three SPA iterations identify and remove three sub-paths *<A,B,D,H,E,F>*, *<K,L,H,P,N>* and *<R,T>*. At the fourth iteration there is no other path subset of GoES in GoOS. Hence, the SPA returns the *<Q>* state since state *<S>* was not observed. Our approach successfully identifies the interference, i.e., Mary does not take her medicine as expected.

Table I
SHC STATE TABLE OBSERVED - OUTCOME 1

Element	Feature	Value	Timestamp	Type
DD	Alarm	ON	10:30 AM	Out
DD	Ringing	ON	10:30 AM	In
Person	Needs Pill	ON	10:30 AM	In
Phone	Call In	ON	10:31 AM	In
Phone	Ringing	ON	10:31 AM	In
Person	Receive Call	ON	10:31 AM	In
Phone	Call	ON	10:32 AM	In
Person	Answers Call	ON	10:32 AM	In
Phone	Ringing	OFF	10:32 AM	In
Phone	Call	OFF	10:45 AM	In
DD	Take Pill	ON	10:46 AM	In
Person	Take Pill	ON	10:46 AM	In
DD	Alarm	OFF	10:46 AM	Out
DD	Ringing	OFF	10:46 AM	In

Table II
SHC State Table Observed - Outcome 2

Element	Feature	Value	Timestamp	Type
DD	Alarm	ON	10:30 AM	Out
DD	Ringing	ON	10:30 AM	In
Person	Needs Pill	ON	10:30 AM	In
Phone	Call In	ON	10:31 AM	In
Phone	Ringing	ON	10:31 AM	In
Person	Receive Call	ON	10:31 AM	In
Phone	Call	ON	10:32 AM	In
Person	Answers Call	ON	10:32 AM	In
Phone	Ringing	OFF	10:32 AM	In
DD	Take Pill	OFF	11:00 AM	In
DD	Notify	ON	11:00 AM	Out
DD	Alarm	OFF	11:00 AM	Out
DD	Ringing	OFF	11:00 AM	In
Phone	Call	OFF	11:10 AM	In

Table III
Introspected Environment Information

Case	Element	Feature	Value	Type	Kind
A	DD	Alarm	ON	OUT	APP
B	DD	Ringing	ON	IN	APP
C	DD	Take Pill	ON	IN	APP
D	DD	Take Pill	OFF	IN	API
E	DD	Alarm	OFF	OUT	APP
F	DD	Ringing	OFF	IN	APP
G	DD	Low Drug	ON	IN	API
H	DD	Notify	ON	OUT	API/APP
I	DD	Low Battery	ON	IN	API
J	DD	Upside Down	ON	IN	API
K	Phone	Call In	ON	OUT	API/APP
L	Phone	Ringing	ON	IN	APP
M	Phone	Call	ON	IN	API
N	Phone	Call	OFF	IN	API
O	Phone	Call In	OFF	OUT	API/APP
P	Phone	Ringing	OFF	IN	APP
Q	User	Needs Pill	ON	IN	APP
R	User	Receive Call	ON	IN	APP
S	User	Take Pill	ON	IN	APP
T	User	Take Call	ON	IN	APP

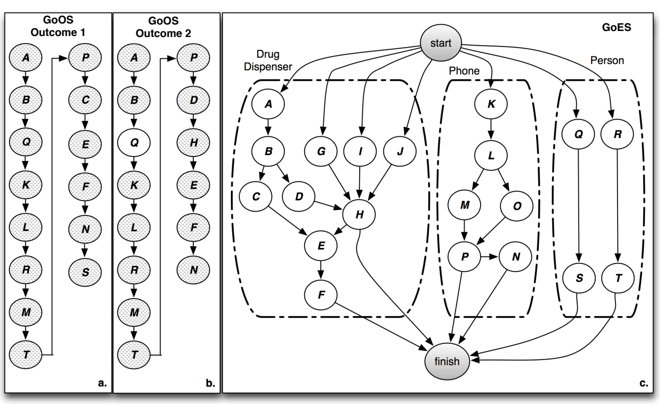

Figure 4. Behavior Model Representation: a) Graph of Observed State - GoOS - outcome 1, b) Graph of Observed State - GoOS - outcome 2 and c) Graph of Expected States

IV. Application to other Scenarios

We apply our intra-system feature interaction detection approach to different scenarios [7] equipped with diverse systems and appliances. For each scenario, we first provide a short description and then identify possible FI outcomes.

A. Scenario Description

- *Scenario A*: This is the example scenario presented above. We identify one FI outcome: *A1* - Mary answers the Phone, talks to her friend over the DD alarm timeout and she forgets to take her pills (i.e., memory failure problem).
- *Scenario B*: Mary is watching the news on TV and, at the same time, the DD triggers its alarm. She needs to take her pills and still watch the news. Here, we identify three different FI outcomes: *B1* - she does not hear the DD because the TV sound overlaps the DD Alarm (i.e., user perception/ awareness); *B2* - she hears the DD but does not take her medicines because she does not want to move away from the TV (i.e., user will); and *B3* - she hears the DD ringing and moves toward it to take her pills thus missing the news (i.e., user will). All these outcomes focus on user senses and behavior: hearing, willingness and awareness.
- *Scenario C*: Mary is using her kinect application to practice the prescribed morning exercises. Meanwhile, the phone starts ringing but to answer the phone she needs to move toward the phone and stop her training. We identify three different FI outcomes: *C1* - Mary does not answer the phone because she does not hear it (i.e., sound overlapping and user perception problems) and she keeps doing her physical practice; *C2* - Mary does not answer the call because she does not want to pause and resume her physical practice (i.e., user will); *C3* - Mary stops her physical practice and answers the call, but after hang-up she forgets to resume her exercises (i.e., memory failures/cognitive memory problem). These outcomes focus on the environment sound and user senses and behavior.
- *Scenario D*: Mary's home is equipped with an intrusion system and also an HVAC (cf. heating, ventilation and air conditioning) to maintain home temperature. The day is sunny and Mary goes to the elderly services center. Before leaving home she turns on the alarm. We identify two possible FI outcomes: *D1* the inner temperature is getting higher, hence the HVAC system opens the windows instead of turning air conditioning (to reduce energy consumption); the intrusion system detects the window is opening and triggers an alarm (i.e., window state/control is shared by two systems); and *D2* the inner temperature is getting higher and the HVAC system keeps opening the windows causing an opportunity for intrusion since Mary forget to turn on the alarm before leaving (i.e., intrusion problem caused by memory/cognitive problems and HVAC system). Systems can lead to unwanted situations as showed in the second outcome.
- *Scenario E*: Mary sits on her couch to watch a movie while the lights are reduced by the entertainment system to create a better ambiance. The home automation system (cf. Home Automation) uses presence detection sensors to turn lights on/off. We identify two possible FI outcomes: *E1* Mary stands still in front of her movie but the home

Table IV
EVALUATION THROUGH SEVERAL SCENARIOS

SCENARIO		IDENTIFIED PROBLEM		DETECTION APPROACH	
Use-Case	Outcome	Description	Category	Kolberg [8]	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	✗	✓
	B3	user looses his TV show	user will	✗	✗
SCENARIO C	*C1*	user does not hear, sound overlapping	user awareness, perception	✗	✓
	C2	user needs to move to another place, interrupt physical practice	user will	✗	✓
	C3	user forgets to resume physical practice	user memory failure, cognition	✗	✓
SCENARIO D	*D1*	alarm behavior influenced by HVAC	shared environment between two systems (i.e., window controller)	✓	✗
	D2	alarm is turned (causing insecurity)	user memory failure	✓	✓
SCENARIO E	*E1*	light intensity change (systems conflict)	shared environment between two systems (i.e., light controller)	✓	✗
	E2	user turns lights on but the media-center dims them down	shared environment between two systems (i.e., light controller)	✓	✗
SCENARIO F	*F1*	false alarm generated by fire control system	shared environment between two systems (i.e., window controller)	✓	✗

automation system detects any small movements and turns lights on (i.e., two systems counter-sharing the lights control); and *E2* Mary deactivates lights automation and manually turns lights to be able to read a magazine while seeing her movie. However, the entertainment system reduces lights on the living room when starting the movie (both the user and the entertainment system control the lights).

- *Scenario F*: Mary leaves the house and turns on the security system. The house possesses also an active fire control system. We identify one possible FI outcome: *F1* the fire control system detects the presence of gas in the kitchen and opens the window for ventilation. At the same time, the security system triggers the intrusion alarm associated with that window opening.

We have used the 3D simulation features of our SHC reflective framework for generating all the presented scenarios and possible FI outcomes. The proposed graph-based algorithm was then applied to each of the scenarios to analyze all particular outcomes. Hence, we will predict the occurrence of an FI through several scenarios/outcomes. In the next Section, we present and discuss the results of our approach.

B. Analysis of Intra-system approach

Table V summarizes the accuracy of our approach, i.e., if we succeed to detect the feature interaction problems.

The analysis of these results allows us to conclude that we successfully manage to identify feature interactions in Scenarios <A,C>, partially detect feature interactions in Scenarios <B,D> and fail to detect feature interactions in Scenarios <E,F>.

In Scenarios <A,C>, we are able to detect feature interactions because these resulted in behavior that was specified in the graph of expected behavior. For example in the C3 outcome, we identify that Mary stops is physical practice to take a call and after forgets to keep on doing is exercise. We detect the FI because the graphs lack one or more sensed states in the user or on the 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 feature interaction. 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. We are currently exploiting additional graph-based representations between inter-systems that might enable us, in the future, to detect and solve feature interaction problems due to unexpected interactions between different OTS systems.

Finally, in Scenarios <B,D> we did not have enough information for detecting the feature interactions, 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 feature interaction detection. In these scenarios, we assume a reasonable information reification level. For example, we are able to access the user agenda to know the medicine prescriptions but we cannot capture if the user is willing to watch a specific TV show (see B3 outcomes in Table V). But if we increase the introspection level so that we may infer the user will (e.g., based on a user profile, combining activity and presence sensors, etc.) then we will also be able to detect all feature interactions identified in these scenarios.

Contrary to the approach proposed by Kolberg [8], our graph-based solution considers the user, i.e., represents the user as another element in the home setting, which might generate or suffer feature interactions. The Kolberg approach considers only FI between systems but is able to detect FI in some of the outcomes that we are not. These outcomes, however, deal with issues that do not affect the expected functioning of the systems; rather, these feature interaction outcomes result from system interactions that occur through a shared medium or variable, i.e., indirect intra-system relationships. We are currently extending our graph-based approach for supporting intra-system relationships.

Table V
EVALUATION OF INTRA-SYSTEM APPROACH THROUGH SEVERAL SCENARIOS

	FI Outcomes	Intra-System FI Detection
Scen. A	1	100% (1/1)
Scen. B	3	66% (2/3)
Scen. C	3	100% (3/3)
Scen. D	2	50% (1/2)
Scen. E	2	0% (0/2)
Scen. F	1	0% (0/1)

V. INTEGRATION IN THE SHC SYSTEM

The presented graph-based approach is a core component of the SHC reflective framework (cf. FI Inference Engine depicted in Figure 5). 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 [9]; then, exploit the simulation outcomes to detect a priori feature interactions between independently developed appliances (as presented in this paper); 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.

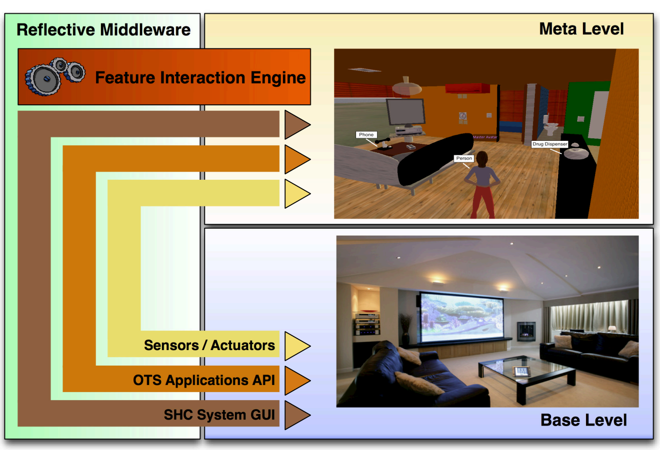

Figure 5. Safe Home Care System - Architecture

A. System Architecture

The SHC architecture is organized in two levels, the base-level and the meta-level, connected through reflective components (see Figure 5). These components interact with the physical environment through sensors/actuators, available OTS APIs (e.g. Asterisk control API) and SHC user interaction facilities. Basically, these components provide the means for reification (e.g., detect human activity and collect OTS states into the meta-level) and reflection (i.e., interact with user and control the environment and OTS appliances).

The Meta-Level uses a 3D virtual environment to represent the home care setting and a data model for storing information about the user, the environment and appliances. For example, we have programmed the behavior of numerous prims to represent several OTS systems deployed at home: drug dispenser (DD), VoIP system (Phone), TV system, light system, environment and the person as well (User). For each of these systems we store their location, state changes and also the source element that triggered the state change (e.g., when the DD element triggers its alarm, it causes a state change in sound property of the Ambient which might then trigger the awareness 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 in this paper, it allows us to prepare a semi-random simulation agenda and generate several outcome scenarios. All state changes are time stamped and stored for further analysis by the FI engine. The FI engine was developed in Java and uses the state sequences to implement the feature interaction detection workflow described earlier in Section II. The entire SHC system runs on a LinuxMCE software platform.

VI. RELATED WORK

In [10] the authors propose a methodology to avoid interference in smart environments. They record normal user interaction patterns with the smart-space. 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 days, as usual, is considered an abnormal behavior and should be reported. In [8], the authors 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.

In [11], the authors present a bi-directional interface between a physical smart space (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 also able to identify problems and interferences related with spatial requirements of devices in the smart space. Similarly to [10] and [8], 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. Moreover, we additionally gather information about the environment and the user perception, also implicitly represented in the graphs. Furthermore, akin to [11], our architecture also proposes the reification of base-level smart spaces; 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 feature interaction between independent built OTS systems.

The purpose of [12] also consists to tackle out interference in Ubicomp domain using a coordinator to monitor the interaction with the physical environment. They combine lock and timeout principles to assure that two different systems will not interact on the same medium at the same time. SHC framework will: i) previously generate all the possible interactions using a virtual-world, ii) identify them using our approach, iii) set a resolution/adaption scenario for possible interactions and iv) apply this set of adaptations to the real-world smart-space. Since [12], they produce reconfiguration on real-time so they introduce a tradeoff on natural behavior execution. Our approach will try to minimize the reconfiguration delay and provide a pre-deployment tool to discover interference. We have previously defined all the reconfigurations for the identified problems. Hence, SHC will only act in real-time using a previously defined strategy to tackle the interference.

VII. Conclusions and Future Work

Current health systems will be unable to cope with the problems raised by the significant increase 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., by 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 smart spaces, 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; ii) provide non-intrusive pervasive interface mechanisms for home assisted-living actors. In this paper, we focus specifically the first goal and explore state-graph representations to perceive feature interactions. We demonstrate the pertinence and efficacy of the proposed intra-system approach on different home care use cases. We use the intra-system approach in simulation [9] scenarios to make a prognostic of possible FI. Moreover, we are able to detect ongoing FI in real deployed systems. We explore the representation and simulation facilities of 3D virtual worlds for generating OTS systems state-graphs used to study more complex feature interaction scenarios. We are currently extending our approach for detecting and resolving inter-system FI between different OTS systems. 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] R. Morla and N. Davies, "A framework for describing interference in ubiquitous computing environments," in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, march 2006, pp. 4 pp. –635.

[2] L. Blair, G. Blair, J. Pang, and C. Efstratiou, "Feature' interactions outside a telecom domain," in *Workshop on Feature Interaction in Composed Systems*, 2001.

[3] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data Mining: A Knowledge Discovery Approach*, 1st ed. Springer, Feb. 2007.

[4] "A dependência: o apoio informal, a rede de serviços e equipamentos e os cuidados continuados integrados," vol. Ministério do Trabalho e da Solidariedade Social, 2009, http://www.cartasocial.pt/pdf/estudo_dependencia.pdf, Gabinete de Estratégia e Planeamento.

[5] J. Cunha, B. Cunha, A. Pereira, W. Xavier, N. Ferreira, and L. Meireles, "Vital-jacket: A wearable wireless vital signs monitor for patients' mobility in cardiology and sports," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, march 2010, pp. 1 –2.

[6] P. Gonçalves, J. M. Torres, P. Sobral, and R. S. Moreira, "Remote Patient Monitoring in Home Environments," in *The First International Workshop on Mobilizing Health Information to Support Healthcare - MobiHealthInf 2009 (in conjunction with BIOSTEC 2009)*, Porto, Portugal, 2009. [Online]. Available: http://isus.ufp.pt/wp-content/uploads/2010/03/mobihealthinf2009.pdf

[7] SHHC, "Safehomehealthcare scenarios outcomes: Interference-free home health-care smart spaces," http://http://homepage.ufp.pt/~csoares/shhc/, Last accessed July 2011.

[8] 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.

[9] 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.

[10] V. Jakkula, D. Cook, and A. Crandall, "Temporal pattern discovery for anomaly detection in a smart home," *IET Conference Publications*, vol. 2007, no. CP531, pp. 339–345, 2007. [Online]. Available: http://link.aip.org/link/abstract/IEECPS/v2007/iCP531/p339/s1

[11] 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. [Online]. Available: http://dx.doi.org/10.1162/pres.18.6.468

[12] 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.

Christophe Soares is a PhD student in Computer Science and a Trainee Assistant Professor in the Sciences and Technology Faculty of University Fernando Pessoa (UFP), Portugal. He is a researcher at the Institute for Systems and Computer Engineering (INESC TEC), Porto, Portugal; and he is also a researcher at Intelligent Sensing and Ubiquitous Systems (ISUS), University Fernando Pessoa, Porto, Portugal. His research interests include system and deployment issues in ubiquitous computing, with a focus on test, simulation and interference. He has an MSc in Mobile Computing from University Fernando Pessoa (UFP). Contact him at INESC TEC Porto <inescporto.pt>, R. Roberto Frias, 378, 4200-465 Porto, Portugal; cpsoares@inescporto.pt and csoares@ufp.edu.pt

José Torres graduated in 1993 and completed his MSc in 1997, both in electrical and computers engineering, major in telecommunications, from Faculty of Engineering from University of Porto (FEUP), Portugal; he obtained his PhD in Computer Science from Faculty of Applied Sciences, Lancaster University, UK, in 2005. He is Associate Professor in the Faculty of Science and Technology from University Fernando Pessoa (UFP), Portugal. He is also a researcher at the Intelligent Sensing and Ubiquitous Systems (ISUS), FCT, UFP; and a research collaborator at INESC TEC, Porto, Portugal, since 1997. His research interests include multimedia description analysis and retrieval, intelligent and ubiquitous systems, computer vision and virtual and mixed reality. Contact him at ISUS <isus.ufp.pt>, Praça 9 de Abril, 349, 4249-004 Porto, Portugal; jtorres@ufp.edu.pt

Rui S. Moreira is Associate Professor at Universidade Fernando Pessoa (UFP), where he is responsible for the Programing Languages, Mobile Applications, Distributed Systems and Ubiquitous Computing courses; He has a PhD in Computer Science from Faculty of Applied Sciences, Lancaster University, UK, in 2003. He completed his MSc in 1995 (major telecommunications) and graduated in 1992 (major in systems and computers) both in Electrical and Computers Engineering from Faculdade Engenharia da Universidade Porto (FEUP), Portugal. Prof. Moreira is also a researcher at Instituto de Engenharia de Sistemas e Computadores (INESC TEC), Porto, Portugal, since 1996; He is also a researcher at the ISUS unit, FCT, UFP, since 2007. His main research interests are reflective middleware for distributed, mobile and ubiquitous systems, applied to mixing cross-reality and multimedia environments, smart spaces and safe home care systems. Contact him at ISUS Unit <isus.ufp.pt>, FCT, UFP: Praça 9 de Abril, 349, 4249-004 Porto; rmoreira@ufp.edu.pt and rjm@inescporto.pt

Ricardo Morla is an Assistant Professor with the Electrical and Computer Engineering Department, and a principal investigator with INESC Porto, at the Faculty of Engineering of the University of Porto. His research interests are in the field of modeling and managing IT systems with an emphasis on probabilistic and machine learning approaches applied to networks and ambient intelligence. Ricardo graduated from U.Porto in Electrical and Computer Engineering and holds a PhD in Computing from Lancaster University. He was a lecturer and post-doc at UC Irvine in 2007, and a visiting faculty at Carnegie Mellon University in 2010 under the CMU-Portugal program. Contact him at INESC TEC Porto <inescporto.pt>, R. Roberto Frias, 378, 4200-465 Porto, Portugal; rmorla@inescporto.pt

Pedro Sobral is an Assistant Professor in the Sciences and Technology Faculty of University Fernando Pessoa (UFP), Portugal, and a researcher at the Intelligent Sensing and Ubiquitous Systems (ISUS), FCT, UFP. His research interests include wireless sensor networks and ubiquitous computing, with a focus on embedded systems and mesh networks. He has a PhD on Telecommunications from Universidade Nova de Lisboa. Contact him at ISUS Unit <isus.ufp.pt>, FCT, UFP: Praça 9 de Abril, 349, 4249-004 Porto; pmsobral@ufp.edu.pt