# Opening the Home. A Web Service Approach to Domotics

Aiello, Marco; Marchese, Maurizio; Busetta, Paolo; Calabrese, Gaetano

UNIVERSITY
OF TRENTO

**DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY**

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

# OPENING THE HOME A WEB SERVICE APPROACH TO DOMOTICS

Marco Aiello, Maurizio Marchese, Paolo busetta, Gaetano Calabrese

December 2004

Technical Report # DIT-04-109

# OPENING THE HOME
# A WEB SERVICE APPROACH TO DOMOTICS

Marco Aiello and Maurizio Marchese
DIT - Univ. of Trento
Via Sommarive, 18
38100 Trento, Italy
{aiellom,marchese}@dit.unitn.it

Paolo Busetta and Gaetano Calabrese
ITC-irst
Via Sommarive, 14
38100 Trento, Italy
{busetta,calabrese}@itc.it

**Abstract**

The pervasiveness of smart appliances in the home, the increase of computational power of traditional home appliances, and the widespread adoption of wireless networking are creating a unique opportunity in domotics. The challenge lies in moving from a set of isolated devices towards a network of home appliances interoperating and scaling seamlessly to provide services to the home user. The major barriers to achieve this is the abundance of non-interoperable domotics standards and the lack of a generally agreed architecture. We propose a service-oriented architecture (SOA) for the interaction of home appliances and the creation of value added services based on standard web services. In particular, we propose the use of WS-Notification as the basic layer for device interoperation. We have experimented the proposed architecture in the context of a project for the assistance of the elder citizen. We provide preliminary results for the case of fall-detection. Addressing the operability of different sensors yields, for the first time, to an integrated and scalable system that detects falls by fusing data transported via a truly vendor-independent publish/subscribe mechanism.

## 1 Introduction

One of the driving factors of the current electronic revolution is the vision that computing will pervade our environments. Any meaningful device or object of our everyday life will be capable of performing calculations and at the same time, communicating transparently with any other computing device, may that be in its physical environment or remotely connected.

Each component can be seen at the same time as a service provider and a service client. For instance, a washing machine will serve information on its status to a home controller and, concurrently, will be a client when connecting via the Internet to an electrical power supplier to negotiate quantity, price and timeframe for power usage. Many standards have been proposed to enable home (local) communication. The shift has gone from that of providing basic communication infrastructure without laying new wires (e.g., X10) towards richer middleware infrastructures providing means for spontaneous networking (e.g., Jini). Proposals such as the Open Services Gateway Initiative (OSGi) allow for the definition of uniform interfaces between home appliances on heterogeneous networks. Other standards include European Home System (EHS), Home Audio Video Interoperability (HAVi), Universal Plug and Play (UPnP), Konnex-KNX, LonWorks, No New Wires, to name a few.

What all these standards lack is a general way of interfacing devices. Spontaneous networking should be achieved not only at the level of connectivity between devices (such as in Jini and UPnP) but at the service level. New devices entering the home should connect to any existing network, communicate, discover services, publish own services and start interacting in the home transparently. This is hard, if not impossible, to achieve with current standards. Open Services Oriented Architectures (SOA) [PG03] are one of the most promising architectural models. A SOA provides the framework to support communication between software modules independently designed, developed and deployed. The visionary promise of SOA is a world of cooperating services where application components are assembled with little effort into a network of services that can be loosely coupled to create agile applications that span organizations and computing platforms.

The key aspects to achieve home pervasive computing at the middleware-application level consist of a stack of interrelated research activities, including: action and service modeling, service composition, request specification and appropriate user interfaces. We propose to use the SOA architecture, and in particular web service technology, as an open middleware for the home of the future. In this architecture, every device is modeled as a service provider and consumer that can publish and discover other devices thus creating a spontaneous network of home appliances. Each device must implement the web service stack [CKM+03], but no assumption is made on the hardware, operating system software or even transport layer. Web services are based on asynchronous XML messaging, which in turn needs no assumption on the home networking infrastructure, e.g., wireless connections (WiFi, Bluetooth, etc.), coaxial cables, or even using the power line as carrier. The coordination among the devices is guaranteed by a publish/subscribe mechanism based on the recently proposed standard WS-Notification [WN04]. Devices can register to one or more event servers for the publication or the notification of given events [Int02].

To show the validity and feasibility of the approach, we consider the case of the support of the elder population. The physical inabilities which inevitably come with aging, together with the need of health monitoring, demand special attention which used to be possible only in health care structures. With appropriate domotic technology, it is possible to give the same level of comfort and safety letting the elder citizen live in her/his own home [Sta02]. The goal is not that of offering a technological home to the end-user, which is unlikely to go through a learning curve to use the new potential of her/his home, but rather to transparently pervade the home of intelligence aiding the user, possibly without her/his awareness. Of course, awareness and control can always be returned to the user, if desired.

The specific application we use as our case-study is that of detecting the fall of the home inhabitant, which has an incidence of at least 30% amongst persons aged over 65 [SJ04]. This type of accident is attracting the attention of domotic research: in [NCM04] the monitoring of the user is achieved via a camera which detects posture, in [SJ04] the monitoring is achieved using an array of infrared sensors. Other work on fall detection has been reported in [DJRW03, NBV$^+$03, DVD$^+$02]. We propose a new approach for the home middleware based on open web service standards, and show how it can be used to integrate distinct sensor information. The fall detection prototype we built is novel as it integrates different sensors and lets users register to be notified of various home events seamlessly. We use both a set of wireless cameras to detect the posture and an electronic accelerometer to detect possible falls. The use of distinct sensors is possible thank to the web service notification infrastructure. All the sensors register in order to publish events. The advantages of this architecture are twofold: on the one hand, it is possible to integrate several information sources simply by having them register their information on the WS-Notification server; on the other hand, various devices can subscribe to the generated events. For instance, a relative can register his PDA to be notified of falling events, a medical doctor could be notified of specific physiological circumstances, a neighbor could register his PC for falling events. Privacy and security concerns are of paramount importance in the actual deployment of such an architecture, but are beyond the scope of the current treatment.

The remainder of the paper is organized as follows. In Section 2 we review the Service-Oriented Architectural model. In Section 3 we propose a domotic middleware based on the SOA. Section 4 is concerned with a concrete case study for the monitoring of the elder. Concluding remarks are presented in Section 5.

## 2 Service Oriented Architectures (SOA)

Web-Services are a set of protocols to enable communication between indipendent software modules that offer their funcionalities in the form of services. Current Web-Services are based on Services Oriented Architectures (SOA). Such sofware architectures utilize services as the lightweight constructs to support the development of rapid, low-cost and easy composition of distributed applications. In a Service Oriented Architecture, services are self-contained, modular applications - deployed over standard middleware platforms, e.g., J2EE - that can be described, published, located, and invoked over a network.

The service-oriented computing paradigm promises to revolutionise the process of developing and deploying distributed software applications. Benefits of applying the service-oriented computing paradigm include reduced complexity and costs, exposing and reusing core business functionality, increased flexibility, resilience to technology shifts and improving operational efficiency.

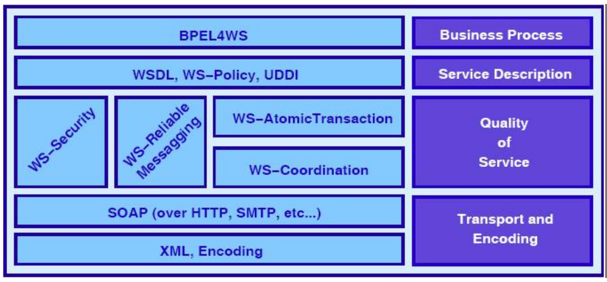


Figure 1: The web service protocol stack.

To support the realization of the SOC paradigm, Web service need to be based on standardized definitions of an interoperability communication protocol, mechanisms for service description, discovery, and composition as well as a basic set of quality of service (QoS) protocols. The initial trio of Web service specifications, SOAP[1], WSDL[2], and UDDI[3], provided open XML-based mechanisms for application interoperability (SOAP), service description (WSDL), and service discovery (UDDI). SOAP is now a W3C standard, and WSDL and UDDI are being considered by standard bodies. In order to implement this basic framework in real applications, mechanisms for service composition and quality of service protocols are required. Several specifications have been proposed in these areas, most notably the Business Process Execution Language for Web Service (BPEL4WS)[4] for service composition, Web service coordination (WS-Coordination)[5] and Web service transactions (WS-Transaction)[6] to support robust service interactions, Web service security (WS-Security)[7], and Web service reliable messaging (WS-ReliableMessaging). The descriptive capabilities of WSDL can be enhanced by the Web Service Policy Framework (WS-Policy), which extends WSDL to allow the encoding and attachment of QoS information to services in the form of reusable service "policies." All these aspects are critical elements for meaningful services interactions.

The described Web Service protocol stack is shown in Figure 1, from [CKM$^+$03].

[1] `http://www.w3.org/TR/soap/`

[2] `http://www.w3.org/TR/wsdl`

[3] `http://www.uddi.org`

[4] `http://www-106.ibm.com/developerworks/library/ws-bpel/`

[5] `http://www-106.ibm.com/developerworks/library/ws-coor/`

[6] `http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/`

[7] `http://www-106.ibm.com/developerworks/webservices/library/ws-secure/`

On the lower level of the stack one finds transport and encoding layers, in the middle level protocols for service description, security, transaction and coordination are located, and, finally, on the top level the protocol stack has the business process composition layer. In [PG03], more details of the service-enabling protocol stack are presented.

## 3 Proposed framework

We propose to take full advantage of the SOA approach in the context of domotics by 'opening the home' in the sense that any device, sensor, appliance which implements the standard web service stack is welcome to join the home computing environment. To achieve this goal we consider a specific portion of the web service stack. Referring to Figure 2, the transport layer is based on SOAP messaging over any kind of transport protocol. Above the most relevant issue of Quality of Service is security. Service offered and searched by home appliances are described via WSDL, usage policies are defined via WS-Policy and services are found and published via UDDI registries. Finally, at the top of the stack we have WS-Notification to coordinate the interaction of all home devices.

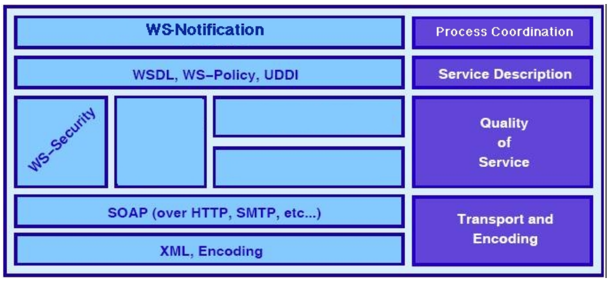


Figure 2: The web service protocol stack for domotics.

Only recently (march 2004), a WS-Notification[8] specifications has been proposed by a consortium of software vendors lead by IBM. WS-Notification is a family of related specifications that define a standard Web service approach to notification using a topic-based publish/subscribe pattern. In particular, it includes three normative specifications: (1) WS-BaseNotification defines the Web service interfaces for NotificationProducers and NotificationConsumers. It includes standard message exchanges to be implemented by service providers that wish to act in these roles, along with operational requirements expected of them. This is the base specification on which the other WS-Notification specification documents depend; (2) WS-BrokeredNotification defines the Web service interface for the NotificationBroker. A NotificationBroker is an intermediary which, among other

[8]http://www-106.ibm.com/developerworks/library/specification/ws-notification/

things, allows publication of messages from entities that are not themselves service providers. It includes standard message exchanges to be implemented by NotificationBroker service providers along with operational requirements expected of service providers and requestors that participate in brokered notifications; (3) WS-Topics defines a mechanism to organize and categorize items of interest for subscription known as ”topics.” These are used in conjunction with the notification mechanisms defined in WS-BaseNotification. WS-Topics defines three topic expression dialects that can be used as subscription expressions in subscribe request messages and other parts of the WS-Notification system. It further specifies an XML model for describing metadata associated with topics.

Any computational element taking part into the domotic environment must thus be able to communicate with the others and implement the SOAP messaging scheme. It describes itself in WSDL and publishes its own services in the UDDI registries. WS-Security guarantees access control and confidentiality of the messages, while WS-Policy can be used to expose services operational behaviors. Interaction among devices is achieved either by direct WSDL invocations, or via event notification and registration at the level of process coordination via WS-Notification.

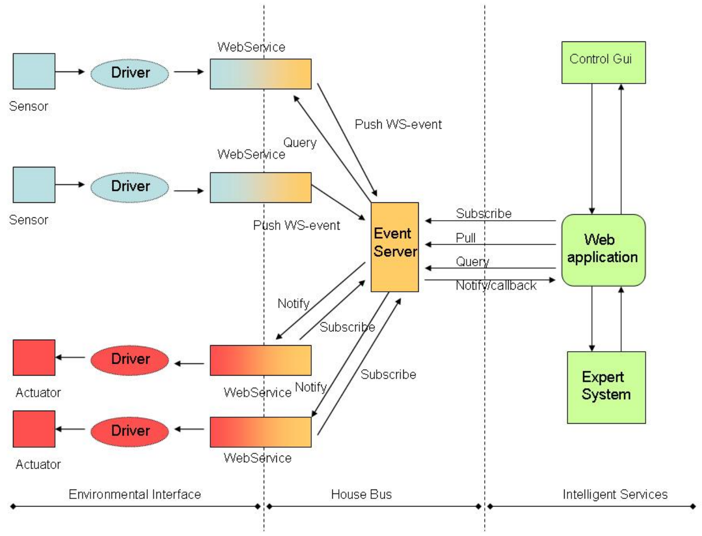


Figure 3: Notification in domotics: a typical configuration.

As an example, we consider the case of a set of heterogeneous sensors which pervade a domestic environment. These sensor differ in what they measure, but also in the operating system and network interfaces. These have drivers that can read and compose SOAP message. The drivers also implement WSDL interfaces. The WS communication is used to publish events and offer sensor readings to an *event server* implementing the WS-Notification protocol. Similar is the case of a set of home actuators. Sensors and actuators are depicted in Figure 3 on the left. Central in the figure is the WS-Notification server which collects sensor information, queries the sensors and operates the actuators. At the same time, the WS-Notification serves as the middleware for intelligent services (right in the figure). For instance, a web application could be monitoring and acting in the home environment. The web application uses an expert system to reach high level conclusion on the status of the home and which intervention is necessary at any given moment. With an appropriate graphical user interface GUI this is the entry point for the human remotely supervising the home.

The architecture of the event server comprises several interacting components. First, events generated by devices, sensors in particular, may be fused in order to generate higher-level events (for instance, the fact the someone is laying down should not trigger a fall event, unless there is also an acceleration event). To fuse sensor data, a rule engine is used. Notice that a sensor can directly generate an event by simply having a rule which does not require any additional condition for event propagation. The rule engine is depicted on the left of Figure 4. The

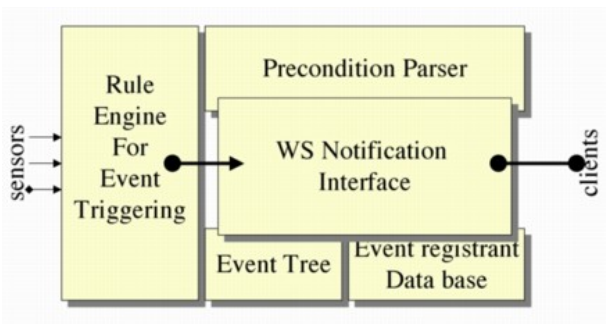


Figure 4: The event server.

rule engine is connected to the actual WS-Notification interface. The hierarchy of events (event tree) that can be generated and that one can be registered is managed in a separated component. Notice that the event tree is a dynamic structure that can be modified at run time by the event server, but also by clients via appropriate WSDL calls. Client registration to events are stored in an appropriate database in the event server. Registrations can be made at any instant to atomic events or to complex events. Registration may expire or be cancelled.

The WS-Notification protocol contemplates the possibility of having *preconditions* and *selectors* connected with the registration of an event and controlling its generation and delivery. For instance, a client could register to the event of the home user fall, only under the condition that it is night.

# 4 The ITEA case study

The Trentino Institute for Public Housing (ITEA) has started a domotic program to aid the citizen with physical disabilities. Particular attention is given to the elder population, as the average age is increasing. In Trentino Italy,[9] 17,8% of the total population is above 65, 8,7% is above 75, and 2,4% above 85. The goal is to let the elder citizens live in their home as long as possible to avoid the discomforts of hospitalization, but providing the same level of safeness and health monitoring.

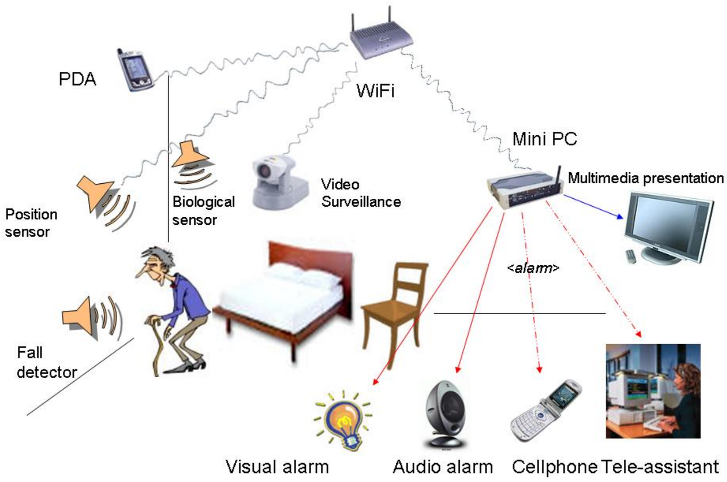


Figure 5: ITEA domotic scenario.

The reference scenario within the ITEA program is that of an autonomous elder home user whose environment is pervaded with sensors and computing devices as in Figure 5. Example of sensors to be installed include position sensors, biometrical sensors, cameras, accelerometers. The sensors are all connected, possibly wirelessly, to an event server which is also responsible for the initial analysis of the raw data (running in the mini PC in the figure). A number of devices subscribe to the events and react to them, e.g., a PDA, a cellphone, a remote human assistant, or a visual alarm.

In the first phase of the project, we have decided to focus on the detection of the fall of the home inhabitant. This is one of the most common home accident for elder people. To detect a fall we use two types of sensor: (1) a custom wearable MEMS accelerometer with WiFi radio transmitter, being developed by ITC-irst and DIT-Univ. of Trento; (2) a set of WiFi cameras together with a sophisticated image processing module computing mass center and posture in real-time, developed by

[9]Data from the Italian population census, ISTAT 2001.

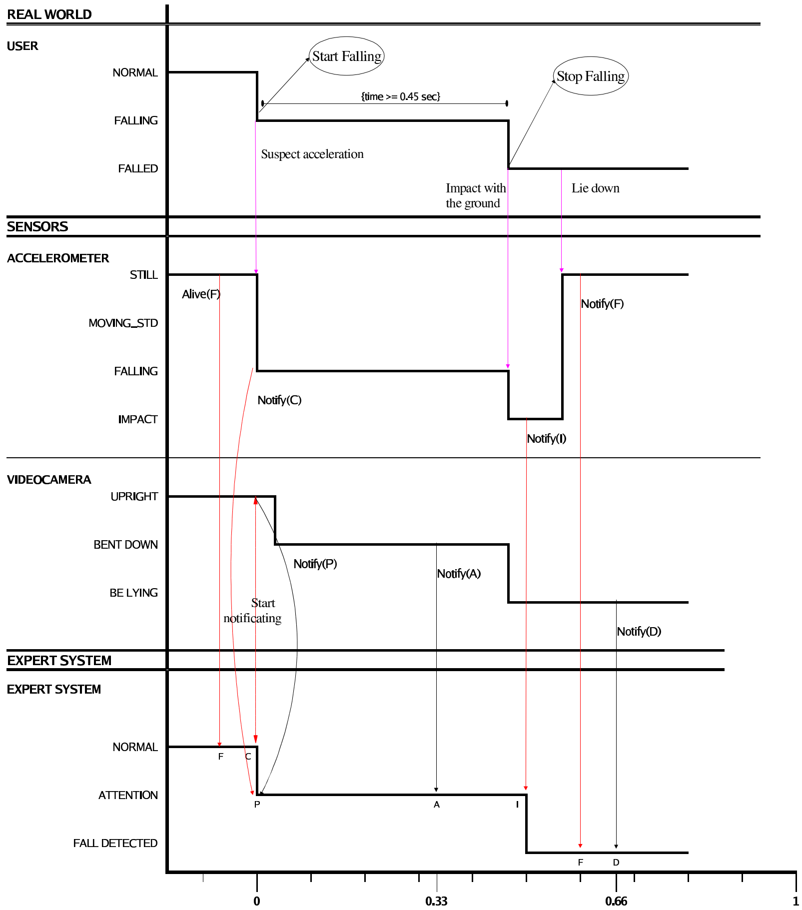


Figure 6: Event generation in case of fall.

CNR-IMM of Lecce. The devices publish the events on the event server; one of their subscribers is an expert system that provides istantaneous reaction to alarm situations and longer-term mining of data to detect patterns of behaviour. The expert system is based on the JESS rule engine.[10]

An example of the generation of events by sensors and by the expert system with respect to the actual status of the world is presented in Figure 6. In that example, at time 0 the user begins to fall and at time 0,5 seconds he reaches the ground and lies down (top of the figure). A possible behaviour of the accelerometer is represented below in the figure: at regular intervals the accelerometer sends a liveliness message to inform of his correct functioning, then it notifies when it detects an acceleration due to the fall and an even stronger, negative acceleration due to the impact with ground. Finally, no acceleration is detected when the user is lying on the floor. The images coming from the camera are analyzed and the

[10] http://herzberg.ca.sandia.gov/jess/

position of the user is categorized in {upright, bent, lying}. The software is very accurate as long as there are no other people in the range of the camera, otherwise tracking of the user may result in mistakes. Finally, the expert system aggregates the notifications from the accelerometer and from the camera and generates an “attention” event followed by a “fall detected” event.

The implemented architecture is completed by a wireless PDA (Palm Tungsten C running PalmOS) subscribing to the “fall detected” event, and generating an alarm whenever notified of the fall event.

In the fall detection scenario just presented it is crucial that the response time to a falling event is as fast as possible. Hard real-time requirements are not possible to enforce with a distributed asynchronous architecture based on web services such as the one introduced here, but a high quality of service in the generation of the high-level events is essential for feasibility of the deployment in real homes. Thus, latency time in event generation is an important parameter to consider.

We implemented a prototype based on the architecture presented in Section 3 to test the feasibility of the approach and to be deployed in the home laboratory of ITEA. The system is an implementation of the WS-Notification protocol [WN04]. It consists of an event server implemented using the Java J2SE, JDK 1.4.1. The server implements a custom XML parser to handle SOAP messages, WSDL and WS-Notification messages. We also implemented in Java three hosts that subscribe and publish events with the server simulating the sensors and rule based system.

We tested the implementation to verify the latency time occurring between the generation of an event and of the delivery to a subscriber. This to check whether the system is suitable to be used in an environment populated by real people who may fall. The event server and the devices have been tested on a standard PC with the following hardware: Pentium 4 at 2,61 GHz, 512 MB RAM and operating system Windows XP Prefessional service Pack 1. The test consisted of the generation of 500 events by the publishers, the forwarding of the events by the server to the appropriate subscribers, and by the receipt of the the event notification by the subscriber. The results are summarized in Figure 7.

| | pub-delivery | serv handling | serv. delivery |
|---|---|---|---|
| average | 162 | 13 | 149 |
| median | 110 | 15 | 109 |
| stdDev | 150 | 17 | 141 |
| max | 672 | 110 | 657 |
| min | 15 | 0 | 15 |

Figure 7: Latency times in milliseconds.

On the first column of the table, the latency time occurring between the generation of an event and the delivery to the correct subscriber, time is measured as the average, median, maximum and minimum time expressed in milliseconds. The second and third columns represent the sub-times for the handling of the event by the server and for the delivery of the event to the appropriate subscriber by the server. The values indicate average and median latency times of about 1/10 of a second and peaks of at most about half a second. These values are fully satisfactory and meet the soft-real time constraints set of the fall detection scenario. The experimentation has been carried out on the same machine to avoid the impact of network latency on the measurement.

# 5 Concluding Remarks

We proposed a SOA architecture for domotics. The novelty of the approach lies in the openness of the resulting home network where any device is seamlessly added and removed independently of its hardware, operating system and network interface. The approach is based on WS-Notification. To the best of our knowledge, this is the first implementation of the standard in a prototype deployed for a concrete application.

The WS-Notification server has been implemented and tests are underway in a real home environment where two type of sensors have been deployed: a wearable custom accelerometer and a videocamera. Furthermore, registration of events and modification of the event tree is performed by a PDA. The test case is the fall detection of the elderly at home. Previous approaches to fall detection have used one sensor only. The open architecture we propose allows the use of distinct sensors for fall detection.

Future work includes checking the feasibility of the approach in the case of the fall detection in a home with actual elder inhabitants in Trento, Italy. Feasibility issues include notification latencies, generation of false positives, device power consumption, overall robustness. Finally, implementation of other components of the domotic stack, such as WS-Security and WS-Policy will be investigated and possibly integrated into the current implementation.


## Acknowledgments

We acknowledge the programming of the WS-Notification event server by Manuel Zannoni, of the Palm Tungsten C by Alessandro Zolet, of the precondition processor by Alida Palmisano, of the rule engine by Linda Tambosi [Tam04], and the preliminary study of domotic standards by Paolo Leoni [Leo04]. We thank ITEA for making available the experimental home in Piazza Garzetti, 17, Trento, Italy.


# References


[CKM+03] F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana. The next step in web services. In *Commun. ACM* [PG03], pages 29–34.

[DJRW03] T. Degen, H. Jaeckel, M. Rufer, and S. Wyss. SPEEDY: a fall detector in a wrist watch. In *Proceedings. Seventh IEEE International Symposium on Wearable Computing*, 2003.

[DVD+02] J. Demongeot, G. Virone, F. Duchêne, G. Benchetrit, T. Hervé, N. Noury, and V. Rialle. Multi-sensors acquisition, data fusion, knowledge mining and alarm triggering in health smart homes for elderly people. *Editions scientifiques et médicales*, 325:673–682, 2002.

[Int02] S. S. Intille. Designing a home of the future. *IEEE Pervasive Computing*, 1(2):76–82, 2002.

[Leo04] P. Leoni. A web service based domotic architecture, 2004. Univ. of Trento, `http://dit.unitn.it/~aiellom/tesi/leoni.doc` (in Italian). Winner of the ANIA-Confindustria prize.

[NBV+03] N. Noury, P. Barralon, G. Virone, P. Boissy, and M. Hamel nad P. Rumeau. A smart sensor based on rules and its evaluation in daily routines. In *IEEE Engineering in Medicine and Biology Society*, pages 3286–3289, 2003.

[NCM04] H. Nait Charif and S. J. McKenna. Activity summarisation and fall detection in a supportive home environment. In *Int. Conf. on Pattern Recognition (ICPR)*, 2004.

[PG03] M. P. Papazoglou and D. Georgakopoulos. Service-oriented computing. *Commun. ACM*, 46(10):24–28, 2003.

[SJ04] A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing*, 3(2):42–47, 2004.

[Sta02] V. Stanford. Using pervasive computing to deliver elder care. *IEEE Pervasive Computing*, 1(1):10–13, 2002.

[Tam04] Linda Tambosi. Monitoraggio di incidenti domestici: un sistema guidato da regole per l'integrazione di dati multi-sensoriali, 2004. Univ. of Trento (in Italian).

[WN04] WS-Notification. Publish-subscribe notification for web services, 2004. `http://www-106.ibm.com/developerworks/library/specification/ws-notifica%tion/`.