

# Semantic Knowledge Management and Integration Services for AAL

Gianfranco E. Modoni, Mario Veniero and Marco Sacco

**Abstract** The integration of a set of heterogeneous data streams coming from different source into a coherent scheme is still one of the key challenges in designing the new generation of AAL system enabling the Smart Home. This paper introduces a service-oriented platform that aims to enhance data integration and synchronization between physical and virtual components of an AAL system. The idea behind this research work goes in the direction to find scalable technological solution in order to answer the continued growth of objects (Things) connected to the network within the domestic environment. Thus, the Smart Objects can operate synergistically on the basis of a shared semantic model, supporting various tailored services that assist elderly users or users with disabilities for a better and healthier life in their preferred living environment. Moreover, a prototype of the platform has been implemented and validated in order to prove the correctness of the approach and conduct a preliminary performance evaluation.

**Keywords** AAL · Integration services · Internet of things · Semantic web · Knowledge management

---

G.E. Modoni (✉)  
Institute of Industrial Technologies and Automation, National Research  
Council, Bari, Italy  
e-mail: gianfranco.modoni@itia.cnr.it

M. Veniero  
ABMEDICA SPA, Cerro Maggiore, MI, Italy  
e-mail: veniero.mario@abmedica.it

M. Sacco  
Institute of Industrial Technologies and Automation, National Research  
Council, Milan, Italy  
e-mail: marco.sacco@itia.cnr.it

## 1 Introduction

Today's homes are environments where various devices perform separate and isolated tasks. Instead, future homes can become systems of distributed and interconnected smart objects (SO) working together in a reliable and predictable manner [1]. In fact, on the basis of the increasingly widespread protocol of the Internet of Things (IoT), the SO can acquire, handle and share the knowledge about the home inhabitants in order to meet the goal of achieving their comfort and well-being [2], thus enabling the model of Smart Home (SH) [3].

One of the main application of the SH is in the field of the Ambient Assisted Living (AAL), where the SH can provide tailored services that support users with disabilities for a better, healthier and safer life in their everyday living environment; e.g. such services can be used to drive the behaviors of users through the processing of the context information. However, the implementation of the SH entails a wide range of technological and scientific problems to be tackled. One of the most relevant is represented by the lack of interoperability between the various involved smart objects which often isolates significant data sets and emphasizes the "existing problem of too much data and not enough knowledge" [4]. This issue is mainly due to the adoption of different communication interfaces, since devices often are produced from various vendors, which use different programming languages, operating systems, and hardware. Moreover, even many standards have been defined also in the AAL field, this issue still remains to be faced.

Under these conditions, it is necessary to adopt a new model of collaboration among the various involved sources, regardless the information representation formats. In order to address this problem, various generic models of interoperability have been proposed by researchers. Sacco et al. have proposed the Virtual Home Framework (VHF) [5], that represents a possible pattern based on Semantic Web technologies to be applied to solve such issue of integration. On the basis of this reference model, the research introduced in this paper analyzes and designs a service-oriented platform [6] for ambient assisted living (AAL) systems, in which information coming from different types of SO are enriched with semantic metadata, thus contributing to seamlessly integrate, aggregate and synchronize the various SO. The platform, which is one of the main outcomes of the ongoing Italian research project "Design for All" (D4A) [5], allows to handle and maintain near real-time shared semantic representation of the available knowledge, while enabling new, implicit knowledge inferencing, based on semantic derivation rules and ontology entailments. A relevant role here is played by the *Integration Services* and specifically by a publish/subscribe middleware enabling messages exchange among all the loosely coupled SO and applications included in the AAL system. A prototype has been also implemented and validated in order to prove the correctness of the approach and to perform a preliminary performance evaluation.

The remainder of the paper is structured as follows. Section 2 examines the motivation for this research work, whereas Sect. 3 illustrates the main characteristics on the basis of the service-oriented platform. Section 4 presents the conducted experiments. Finally, Sect. 5 draws the conclusions, summarizing the main outcomes.

## 2 The Vision

The proposed approach mainly addresses the data integration issue, which is particularly relevant when talking about complex domain models often expressed as huge, intensive, and multisource data [7]. The idea behind this work is to provide AAL applications and tools with a common and high-level interfacing means, the *Integration Services* (Fig. 1), allowing them to access a shared domain knowledge managed by the *Semantic Repository*, contributing to simplify collaboration among applications.

This approach represents a common best-practice when dealing with Enterprise Information Systems [8, 9]. Applications can easily interoperate while the handled data are integrated, aggregated and shared or dispatched through mechanisms that are transparent to their clients. In the context of the D4A project, *Integration Services* foster the semantic integration among several different domestic devices and tools, contributing to enhance their near real-time synchronization capabilities and making them smarter. In fact, the *Integration Services* allow to abstract from implementation details of each given device, hiding the complexity of their different interfaces. In this sense, *Integration Services* enable a proper SO interaction and enhance their capability to exchange information providing a seamless view on high-quality data extracted through a common and generalized interface providing the following main functionalities:

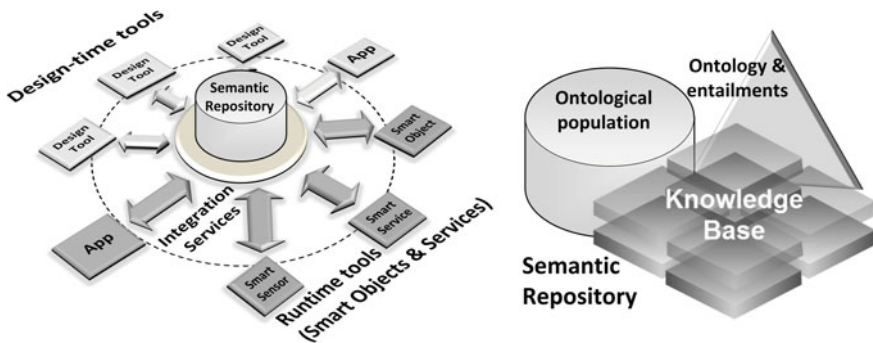


Fig. 1 General vision behind the integration services

1. acquisition of information from any SO;
2. storing, interpreting and properly managing the information received;
3. sharing or proactively dispatching information when a SO asks for them or a needed information becomes available.

*Integration Services* comprise a specific service (the *Information Dispatching Service*) which makes available the changed information to a whatever application through a multi agent server enabling the near real time semantic signaling. Under these conditions, the *Information Dispatching Service* plays the role of an IoT middleware that supplies the central point which join heterogeneous devices communicating through heterogeneous interfaces. It also implements the major functionalities that Sacco et al. [5] conceived for the Virtual Home Manager, the component of the reference framework that enables the bidirectional connection between the two worlds of the real home and the digital home.

The proposed service-oriented platform has been implemented and tested in the context of D4A project activities to integrate both AAL design tools and runtime smart services and objects included in an AAL system (Fig. 1). In order to test and validate the proposed platform, some demonstration scenarios have been identified, which are thought to represent habits and activities occurring on a regular basis in a domestic environment. A significant one focuses on the user during the grocery shopping; in this situation the idea is to monitor food inventory through a smart interaction between a refrigerator and a smart phone. Performances has been evaluated considering not only the pure execution time but also the feasibility and ergonomics w.r.t. developers and systems integrators.

### 3 The Integration Services

The *Integration Services* provides the following capabilities:

- *Authoring*, which allow each application to manipulate shareable owned knowledge through the operations of insert, update and delete;
- *Retrieval*, which allow the extraction (retrieval) of both asserted and inferred knowledge from the shared domain;
- *Signaling*, allowing socially connected applications to register themselves and receive alert related to the changing of the state of one or more interesting elements.

Through these capabilities, the *Integration Services* allow each application or smart object distributed in the AAL environment to translate information expressed according to its own applicative ontology into application expressed according to the shared domain ontology and vice versa, thus contributing to create a shared common understanding of the relevant knowledge of the domestic environment.

Figure 2 reports the main steps of this knowledge sharing process in a processing pipeline from authoring to retrieval. Sharable legacy data, whose definition

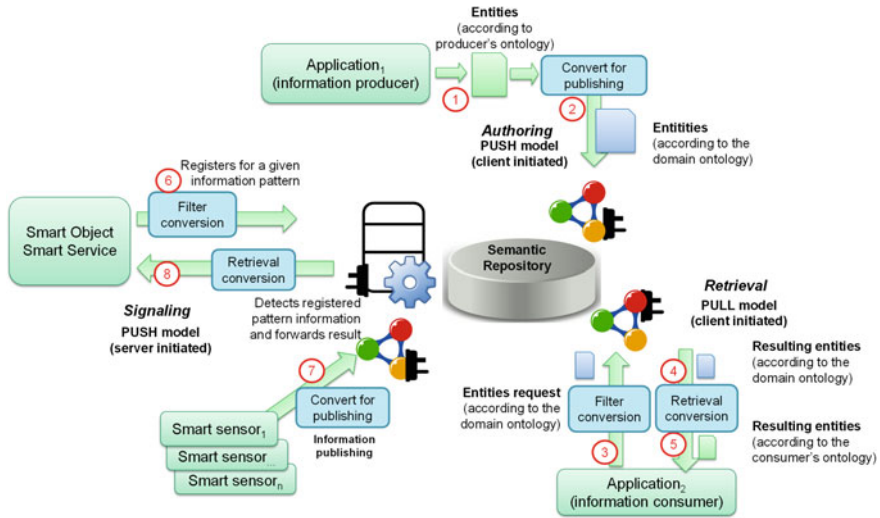


Fig. 2 The processing pipeline concerning the *Integration Services*

is given with respect to the specific application’s ontology, will be translated according to the shared D4A domain ontology (Step 1). Once translated, this common representation is forwarded to the SR (Step 2) in order to be stored by means of semantic data manipulation languages and tools. On the other side, by means of search criteria conveniently converted in a SR understandable form (Step 3), domain ontology represented entities are retrieved (Step 4) by semantically querying shared knowledge, and successively converted in the requestor understandable form (Step 5).

In order to support the aforementioned capabilities, the *Integration Services* comprises the three following components (Fig. 2):

- *Data mash-up services.* It implements the typical data access layer in a multi-tier application architecture, allowing to manage application sharable data. Designed specifically for each application, it is in charge of translating forth and back data entities from the application’s specific format to the shared domain ontology representation.
- *Semantic data access service.* In the form of a shared common application library, is in charge of translating received SPARQL requests into corresponding HTTP request, according to SPARQL 1.1 Protocol for RDF, which is the adopted standard way to compose the semantic query. In this way, the translated request will be served through the cloud integration services component.
- *Information dispatching service,* allowing to real-time distribute emerging information to interested subsystems (services, applications and so on).

### 3.1 *Supporting the Signaling Capabilities Through the Information Dispatching Service*

The *Integration Services* provide near real-time signaling capabilities to all the networked Smart Objects and Services involved in the AAL environment thanks to the *Information Dispatching Service*, which is a multi agent and semantic based signaling server. In particular, the *Information Dispatching Service* plays the role of an effective IoT middleware, allowing applications to register for interesting information and receive back alerts about the changing state or the emergence of interesting knowledge. Moreover, it abstracts and hides the complexities of the hardware or software components, involved within the system.

Under these conditions, this architecture resembles the one of Enterprise 2.0 Social Software (E2.0) [10–12] since it supports social and networked applications concurrently accessing and modifying a shared knowledge domain. The evaluation of the emersion of the asked information is proactively activated by the agent upon recognizing an update of the knowledge-base. Each agent can act as a publisher to send changed information and as a subscriber to receive all the updates related to subscribed information. Figure 2 reports the processing pipeline concerning the *Information Dispatching Service* (Step 6, Step 7, Step 8). These services has the potential to significantly reduce the bandwidth cost of busy spin semantic queries, as well as the required workload both at the client application and knowledge base sides. Moreover, it enables an effective and performant semantic event-driven model to support design and development in the field of AAL; thus reducing the development cost, while also increasing interoperability, quality and portability.

#### 3.1.1 The Architectural Model

The implementation of the *Information Dispatching Service* leverages the specifications of the Foundation for Intelligent Physical Agents (FIPA) [13], which include a full set of computer software standards for specifying how agents should communicate and interoperate within a system. In particular, it has been take into account the FIPA Subscribe-like interaction protocol (IP) specification [14] that defines messages to be exchanged, as well as their sequencing according to a Request-Reply Enterprise Integration Pattern [15].

The *Information Dispatching Service* is enabled by a messaging system, supporting both the publish/subscribe pattern and message queue models. The first allows the specific receivers (subscribers) to express interest in one or more information and only receive messages that are of interest, while the second enables an asynchronous inter-process communications protocol. In this implementation, the selected messaging system is the Apache ActiveMQ™ [16], an open source messaging platform through which clients can make a subscription specifying the SPARQL query of interest, the content type of the expected response and the authorization credentials for accessing the repository (username and password). The

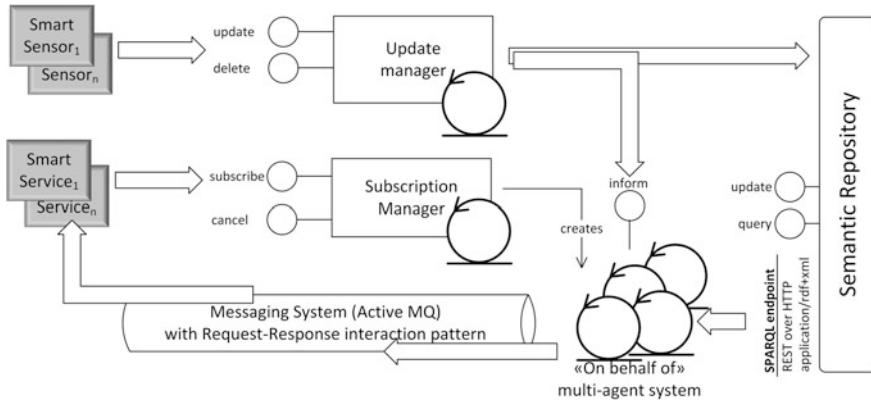


Fig. 3 The architectural model

presented infrastructure is paired with a web server based on the Jersey framework [17] which allows to expose a notification service of the state of the knowledge base changes.

Through the Apache ActiveMQ™, the *Information Dispatching Service* is always listening on the queue that manages the new subscriptions (*SubscriptionManager*) (Fig. 3). Whenever it receives a subscription request from a network client, it activates server-side an agent (the *ClientAgent*) which takes care of the client interests.

The subscription requires that the client sends the following information:

1. the *client-ref*, a unique identifier of the client;
2. the *req-id*, a unique identifier of the query of interest;
3. the SPARQL query of interest, the credentials (username and password) for the authentication, and a boolean flag, *reasoning*, which tells the system whether to activate the reasoning during the query execution (in order to extract implicit knowledge).

After the transmission of the subscription, the client waits for subsequent notifications provided by the *SubscriptionManager* whenever the knowledge base is changed and the corresponding query produces a result different from that previously transmitted. The changes applied to the knowledge base are notified to *SubscriptionManager* through appropriate communication interface (e.g. a REST, etc.). Notifications are automatically (transparently) carried by the operator SPARQL endpoint at each interaction of type SPARQL Update. In summary, the main operations performed server-side by the *Information Dispatching Service* are:

1. the activation of a queue listening on new subscriptions;
2. the activation of a queue listening on new notifications of the knowledge base changes;
3. the execution of the subscribed queries, reports any change to the listening client;

while the main operations performed client-side are:

1. the creation of a temporary queue of messages specific for the client;
2. the preparation of the message for the subscription and listening of notifications;
3. the sending by the client of the request for subscription to one or more of the queries;
4. the cancellation of the registration, by closing the connection.

### 3.1.2 The Subscription Protocol

The client starts the interaction with the server through a subscription message containing the description of the information of interest (specified in a query) with the minimum refresh rate in milliseconds, together with a unique identifier of the request (*req-id*) and a reference (pointer, address, etc.) of the client (*client-ref.*) to which forward the discovered information.

The query transmitted from the client is expressed through the SPARQL 1.1 syntax and may be a SELECT, ASK or CONSTRUCT that refers to semantic model contained in the repository. The server processes the subscription request and decides whether to accept it. If it is rejected, the repository sends to the client the rejection condition ending the interaction. If it is accepted, at each interval of minimum refresh rate, the server updates the evaluation of the subscribed queries and transmits an information message (*inform-result* to the client) containing the result of the executed query (*query-result*) if the result is not empty (SELECT or CONSTRUCT query type) or positive (ASK query type), according to the chosen response format.

The server continues to broadcast type messages (*inform-result*) as long as one of the following conditions happen:

1. the client deletes the subscription request by the cancellation request (see next section);
2. an error occurs for which the server is no longer able to communicate with the client or to process queries.

All interactions are identified by a unique identifier other than zero (*req-id*) assigned by the initiator of the protocol and valid for it (*client-ref*). This allows stakeholders to manage their communication strategies and activities. Moreover, since it can be important to preserve the sequence of the messages, the transport layer has to preserve the order of the messages (reliable transport layer). Thanks to the oneness of the *req-id*, each client can participate in multiple signaling at the same time. Figure 4 reports the overall workflow of the subscription process.



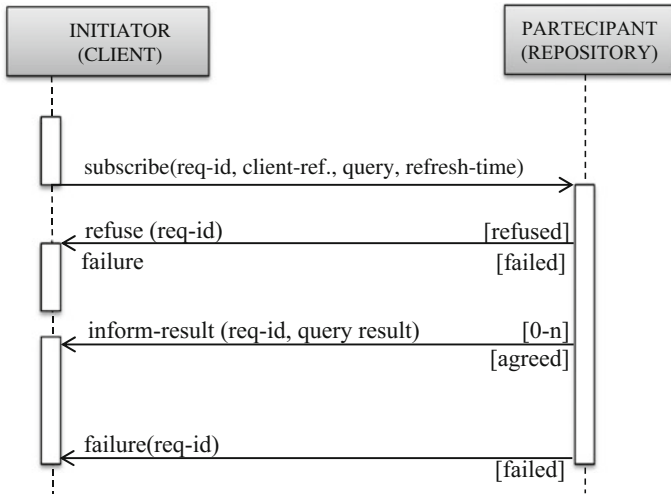


Fig. 4 Subscription and notification workflow

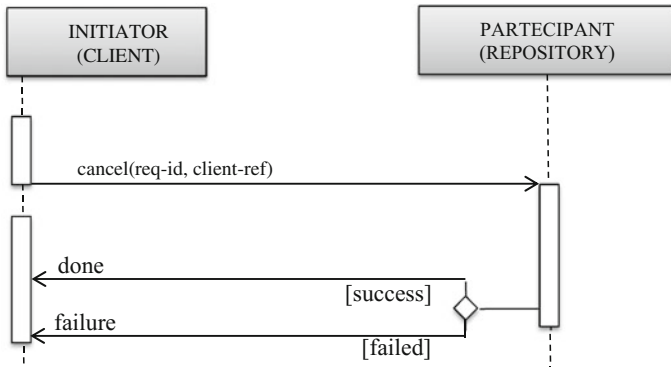


Fig. 5 Workflow of the cancellation of the subscription

### 3.1.3 Cancelling Subscriptions

At any time, the client may cancel a subscription request by transmitting a cancel request to the server. In such a request, the parameters *req-id-ref* and *client* identify the interaction to be stopped (Fig. 5). The server inform then the client if the interruption succeeded (*done*) or that it was not possible to break the interaction due to an error (*failure*).

### 3.2 *The Data Persistence Through the Semantic Repository*

A significant study during this research has regarded the identification of a valid Semantic Repository (SR) capable to handle large amount of Semantic data, also in the form of Big Data [18]. Managed data comprise:

- the domain ontology, as representation of the knowledge about home environment;
- the ontological population, compliant to the domain ontology;
- the derivation rules needed to properly entail the implicit knowledge.

This component must also provide reasoning capability in order to automatically infer new knowledge about the concepts and their relationships, starting from the explicitly asserted facts [19].

A study of the state of the art of a set of existing semantic repositories has been carried out, considering the reasoning capability as main criteria of evaluation, since it represents an essential requirement for the platform currently under development. The result of the survey is that majority of reasoners are still far away from conformance to the full specification of the OWL (Ontology Web Language) [20], the standard language used to represent semantic data. One of the causes is the great expressivity allowed by OWL that results in difficult (if not impossible) reasoner implementations. OWL, in its full specification, is undecidable. (i.e. query answering for an OWL ontology needing reasoning could require infinite time). Since decidability is an important property in real world scenarios, a decidable syntactic subset of OWL has been defined. This leads to an enormous amount of possible implementations, and appears to disrupt the standardization effort made by product developers. In the context of this project, we have evaluated and finally adopted Stardog [21], since it supports the largest subset of OWL compared to other solutions, thus, allowing an higher level of expressivity to represent the derivation rules.

Moreover, the here presented platform attempts to deal with the horizontal scalability. This can be also addressed through the enabling technologies of the cloud computing [22]. The Infrastructure as a Service (IaaS) paradigm, characterized in that the providers offer computers (physical or virtual machines) and also other resources, has been used for the tests of the platform, since it is the only offering valid solution compliant with the Semantic Web technologies.

## 4 **The Conducted Experiments**

One out of the defined experimental settings is a context aware *Situation Identification System (SIS)* designed and prototyped by ab medica, one of the D4A project partners. The SIS is aimed at identifying ongoing situations that are relevant

from a cognitive point of view in the reference scenario and selecting services to be activated based on the contextual and situational profile.

The main features of this system are:

- Gathering, analyzing and semantically annotating brain signals, as well as several other bioelectrical and medical information. All these information are to be merged with contextual information acquired by ubiquitous home-automation sensors;
- Extraction of all those features useful to identify situations that are relevant from a cognitive perspective, in the reference scenario;
- Contextual profiling of the observed user and identification of status and behavioral patterns;
- Activation of needed services, identified according to the current contextual and situational user profile.

As can be seen in Fig. 6, the general architecture of *SIS* recalls the Endsley model for Situation Awareness [23] and is based on an specific knowledge base feeding and sustaining situational models for identification and elicitation. These model and their underlying knowledge are, in their turn, modeled according to an application ontology focused on modeling and reasoning on medical and behavioral features of a observer subject. *SIS* exploits D4A Integration Services by getting contextual information coming from smart sensors and objects deployed in the environment such aa internal and external temperature, light levels, devices activation, receipt execution, etc. At the same time *SIS* is able to feed back the repository with information related to the current state of the user and his current health profile.

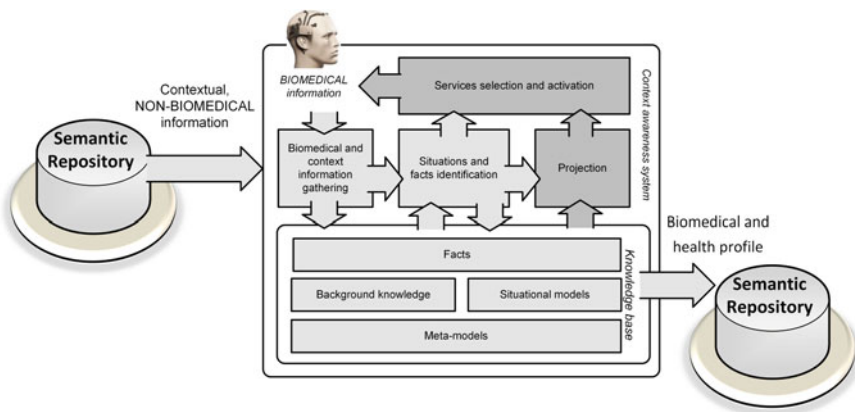


Fig. 6 Situation identification system’s architecture and its interaction in the AAL platform

## 5 Conclusion

This paper has introduced a service-oriented platform for ambient assisted living which contributes to enable a smarter interaction of the various devices distributed within the domestic environment. Devices can cooperate through the integration of the knowledge about home environment, thanks to a strategy leveraging the central role of the Semantic Web Technologies. The proposed platform is particularly relevant in the context of AAL, in the perspective of moving away from more traditional assistive technologies towards an approach that takes into account the full range of human diversity. Nevertheless some issues continues to limit the potential usage of the platform, especially when dealing with legacy systems whose knowledge models are huge or too complex to make the integration process simply economically unaffordable [24]. This problem represents the most relevant technological gap to be addressed in the future developments of the platform [25]. Future work also concerns a quantitative analysis of performance, also exploiting new benchmarks that are currently under study.

**Acknowledgements** This work has been co-funded by the Ministry of University and Research of Italy within the cluster “Tecnologie per gli ambienti di vita—Technologies for Ambient Assisted Living” initiatives, with the overall objective of increasing the quality of life in the domestic environments through the use of the modern technologies.

## References

1. Perumal T, Ramli AR, Leong CY, Mansor S, Samsudin K (2008) Interoperability for smart home environment using web services. *IJSH* 2:1–16
2. Koskela T, Vaananen-Vaino-Mattila K (2004) Evolution towards smart home environments: empirical evaluation of three user interfaces. *Pers Ubiquit Comput* 8(3–4):234–240
3. Harper R (2003) *Inside the smart home*. Springer Science & Business Media
4. Sheth AP, Sahoo SS (2008) Semantic sensor web. *IEEE Internet Comput* 12(4):78–83
5. Sacco M, Caldarella EG, Modoni G, Terkaj W (2014) Supporting the design of AAL through a SW integration framework: the D4All project. In: *Universal access in human-computer interaction. Design and development methods for universal access*. Springer, pp 75–84
6. Jammes F, Smit H, Martinez Lastra JL, Delamer IM (2005) Service-oriented paradigms in industrial automation. *IEEE Trans Ind Inf* 1(1):62–70
7. Modoni GE, Sacco M, Terkaj W (2016) A telemetry-driven approach to simulate data-intensive manufacturing processes. 49th *procedia CIRP-CMS* 2016
8. Hinchcliffe D (2007) *The state of enterprise 2.0*. ZDNET.com, London
9. Cook N (2008) *Enterprise 2.0: how social software will change the future of work*. Gower Publishing Ltd.
10. Bellifemine F, Caire G, Greenwood D (2007) *Developing multi-agent systems with JADE*. Wiley. ISBN 978-0-470-05747-6
11. McAfee AP (2006) Enterprise 2.0: the dawn of emergent collaboration. *MIT Sloan Manage Rev* 47(3):21
12. McAfee A (2009) *Enterprise 2.0: new collaborative tools for your organization’s toughest challenges*. Harvard Business Press

13. FIPA Communicative Act Library Specification. Foundation for intelligent physical agents, 2000. <http://www.fipa.org/specs/fipa00037/>
14. IEEE Computer Society, IEEE foundation for intelligent physical agents (FIPA) standards committee [Online]. Available <http://www.fipa.org/>. Retrieved Oct 2016
15. Hohpe G, Woolf B (2003) Enterprise integration patterns. Addison-Wesley, 650 pages. ISBN 0321200683
16. Apache ActiveMQ™ [Online]. Available <http://activemq.apache.org>. Retrieved Oct 2016
17. Project Jersey [Online]. <https://jersey.java.net/>. Retrieved Oct 2016
18. Modoni G, Sacco M, Terkaj W (2014) A survey of RDF store solutions. In: Proceedings of the 20th international conference on engineering, technology and innovation, Bergamo
19. Modoni GE, Sacco M, Terkaj W (2014) A semantic framework for graph-based enterprise search. *Appl Comput Sci* 10(4):66–74
20. [Online]. Available from <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>. Retrieved Oct 2016
21. Stardog [Online]. Available <http://stardog.com/>. Retrieved Oct 2016
22. Sriram I, Khajeh-Hosseini A (2010) Research agenda in cloud technologies. arXiv preprint [arXiv:1001.3259](https://arxiv.org/abs/1001.3259)
23. Endsley MR (2000) Theoretical underpinnings of situation awareness: a critical review. In: Endsley MR, Garland DJ (eds) *Situation awareness analysis and measurement*. LEA, Mahwah, NJ
24. Canfora G, Fasolino AR, Frattolillo G, Tramontana P (2006) Migrating interactive legacy systems to web services. In: *Conference on software maintenance and reengineering (CSMR'06)*. IEEE, 10 pp
25. Modoni GE, Doukas M, Terkaj W, Sacco M, Mourtzis D (2016) Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models. *Int J Comput Integr Manuf* 1–17