Research Article

A Collaborative Semantic Annotation System in Health: Towards a SOA Design for Knowledge Sharing in Ambient Intelligence

Gabriel Guerrero-Contreras, José L. Navarro-Galindo, José Samos, and José Luis Garrido

Software Engineering Department, E.T.S.I.I.T., University of Granada, C/Periodista Daniel Saucedo Aranda s/n, Granada, Spain

Correspondence should be addressed to Gabriel Guerrero-Contreras; gjguerrero@ugr.es

People nowadays spend more and more time performing collaborative tasks at anywhere and anytime. Specifically, professionals want to collaborate with each other by using advanced technologies for sharing knowledge in order to improve/automatize business processes. Semantic web technologies offer multiple benefits such as data integration across sources and automation enablers. The conversion of the widespread Content Management Systems into its semantic equivalent is a relevant step, as this enables the benefits of the semantic web to be extended. The FLERSA annotation tool makes it possible. In particular, it converts the Joomla! CMS into its semantic equivalent. However, this tool is highly coupled with that specific Joomla! platform. Furthermore, ambient intelligent (AmI) environments can be seen as a natural way to address complex interactions between users and their environment, which could be transparently supported through distributed information systems. However, to build distributed information systems for AmI environments it is necessary to make important design decisions and apply techniques at system/software architecture level. In this paper, a SOA-based design solution consisting of two services and an underlying middleware is combined with the FLERSA tool. It allows end-users to collaborate independently of technical details and specific context conditions and in a distributed, decentralized way.

1. Introduction

Context-aware systems are defined as those which "*use context to provide relevant information and/or services to the user*" [1]; this kind of system adapts to the user and his/her environment, being also possible, among other features, the optimisation of their functioning [2]. intelligence (AmI) environments make use of additional techniques and methods of those adopted by context-aware systems to provide a natural, intelligent, and unconscious interaction with other users and the computational system itself. Thus, interconnected smart computer-based systems (sensor networks, platforms, services, and applications) can allow people to carry out their everyday life tasks by exchanging information, a.k.a. Internet of Things (IoT) [3]. However, the Internet of Things involves an increasing volume of heterogeneous information that makes difficult for people and smart things to manage [4]. In this context, Semantic web is presented as an appropriated approach to facilitate the management of this information, for both people and smart things.

One of the main issues to be resolved in order to progress towards Semantic web is how to convert existing and new information that can be understood by humans into semantically enriched contents that can be understood by machines. Semantic enrichment is made possible by tagging documents with metadata, which enables entities that are found in the contents, and relations between them, to be described. The provision of the information elements that currently make up the web with a well-defined meaning [5] would improve its contextual search capabilities, increase interoperability between systems for collaboration, and allow the automatic composition of published web services [6] to be used by applications.

Nowadays, a wide range of annotation tools for producing semantic tags is available, such as Amaya [7], Epiphany [8], DOSE [9], or Melita [10]. However, such tools often take a platform-centred approach rather than a user-entered one. They usually require complex installation procedures and some are becoming obsolete. In response, a semantic annotation tool called FLERSA [11] has been developed. It was

built to transform a CMS (Content Management System) into its semantic equivalent, in order to partially mitigate the lack of semantic content of the current web and take advantage of the multiple benefits offered by semantic web technologies, likewise the facilities and advantages provided by CMS platforms. FLERSA provides several advantages in comparison to existing annotation tools, such as user-centred interface, lightweight, manual, and automated annotation, avoids the "Deep Web" problem (i.e., search engines indexers can access semantic information stored in documents annotated with the tool, as annotations are embedded within documents in RDFa format), and offers multiontology annotation. However, the FLERSA tool also presents a main architectural limitation, it is highly coupled with Joomla [12], as the CMS is used as the underlying web infrastructure. That is, server-side and client-side implementations cannot be easily adapted to another system. Thus, the possibility of making multiple semantic annotations simultaneously by several users specifically depends on the Joomla! CMS, and it is not provided.

The Service Oriented Architecture 2.0 (SOA 2.0) [13], together with techniques to manage data distribution/replication, may help to address this limitation. In particular, a SOA 2.0 based architecture can provide FLERSA tool capabilities such as scalability, interoperability, and business agility, through the service encapsulation. Replication is crucial to obtain data high-availability, especially in AmI environments. However, they pose additional challenges due to frequent changes in their execution context and often limited resources [14]. Thus, centralized services or static deployments must be avoided in order to provide a higher availability of services [15]. The architecture supports the synchronization and consistency management of distributed/replicated resources through two main services (monitoring and synchronization). In the present research work, that software architecture is applied to the FLERSA tool. In this way, it will allow end-users to collaborate everywhere and anytime by using current technologies for a more advanced knowledge sharing in order to improve business processes. The main goal is to demonstrate the usefulness of a SOA architecture in a specific application domain, as well as its enactment as a common base to facilitate the design and development of AmI and IoT applications regardless of technical details (e.g., specific data source platforms and technologies) and specific context conditions (e.g., wireless disconnections and battery status of the mobile devices).

The paper is organized as follows. Section 2 outlines related work. Sections 3 and 4 present, respectively, an overview of FLERSA for semantic information enrichment and the SOA 2.0 based architecture to support the synchronization and consistency management of distributed/replicated resources. Section 5 shows how this design solution is applied to FLERSA allowing end-users to collaborate everywhere and anytime for a more advanced knowledge sharing and providing FLERSA with a clear differentiation and independence of the client-side. Section 6 presents some of the new settings the proposal enables in a health case study, in which doctors from multiple medical departments and hospitals can collaborate in disease diagnosis. Finally, Section 7 discusses relevant points and summarizes conclusions.

2. Related Work

In the last years several collaborative semantic annotation systems have been proposed. The majority of these systems are asynchronous, while the synchronous systems usually use a whiteboard to share annotations between remote users. Whiteboards allow users see and modify the same document or image, but whiteboards are unstructured and they make it difficult to search and retrieve annotations difficult. In response to this limitation the other collaborative annotation applications emerged. The collaborative semantic annotation systems more commonly used are the semantic wikis. Wikis have a large numbers of users that can modify or create any page using their web browser easily. This kind of systems is asynchronous: a user modifies a page and this is available for the remainder users when it has been stored in the server. The problem of wiki-systems are that when they increase their size they become unstructured and therefore difficult to navigate. Several semantic wikis can be found such as SweetWiki [16] or IkeWiki [17].

There exist several annotation applications that allow users to create and modify annotations collaboratively. Vannotea [18] or eSports [19] allow users to create annotations on multimedia files collaboratively and synchronously. Vannotea is focused on secure collaborative actions; it is based on Annotea [20], for annotation structure; Jabber [21], to instant messaging; Shibboleth [22], a secure middleware to access web resources; and XACML [23], XML-based language to define control policies. Vannotea is developed in C# and it stores semantic annotations separately from the content; this allows users to annotate the same content with different annotations depending on the context. eSports is designed to support the distance coaching. The main contribution of eSports is that it allows annotating live video in real time collaboratively. It is based on NaradaBrokering middleware [24]; this middleware is based on the publish/subscribe messaging model. These applications present some limitations: they are mainly designed to work in desktops and therefore they have not taken into consideration disconnection scenarios that are necessary for mobile devices. Moreover, they have been designed for a specific platform and they are not easily portable to other platforms.

Recently, GATE Teamware [25] has been proposed as another collaborative text annotation tool. The main purpose of GATE Teamware is support complex annotations process on corpus. It is based on multirole model in order to reduce the conflict in annotation tasks. GATE Teamware is web-based and it has a layered architecture: user interface layer, which provides the different web user interfaces, according to user's role; executive layer, which implements user management and user authentication; and services layer, where, in this layer, web services such as storage service and annotation service can be found. However, GATE Teamware is an ad hoc solution; it has been designed to operate only with collaborative text annotation, whereas the service platform proposed aims to be a generic platform that can be adapted to any shared resource.

FIWARE [26] is a new platform that provides software components (called Generic Enablers) with functionalities

for a more rapid, modular, and flexible development of IoT applications. It can be considered a partial implementation of IoT-A. For instance, FIWARE has been applied to the design of a real e-health Remote Patient Monitoring with an agile software development methodology [27] and the agriculture domain [28]. In [29] the focus is on the automatic service migration and deployment between providers' cloud infrastructures within the FI-STAR FP7 project, in which several cloud services have been developed for the healthcare domain using the FIWARE platform. Our proposed architecture, including middleware and service platform, could be another specific implementation for one of these Generic Enablers (e.g., the Orion Context Broker), but based on replicated (decentralized) services, that is, following the vision present in Mobile Cloud Computing [30] and Fog Computing [31].

In [32] a comparison between smart city platforms and their supported features is presented. The CityPulse framework (as other similar ones) supports the development of semantic-based services focused on data discovery, analysis, and integration from different domains in IoT and social media [33] (also called Web of Things [34]) and include a more complete set of analytics tools. That framework provides a Resource Management Module for data distribution, but it does not address the consistent management of replicated information under concurrent operation and specific context conditions (e.g., disconnections).

3. FLERSA: Semantic Annotation on CMS

FLERSA [11] has been developed to transform a CMS into its semantic equivalent, in order to partially mitigate the lack of semantic content of the current web and take advantage of the multiple benefits offered by semantic web technologies. The main originality of the tool is to use the manual and multiple-users annotations that can be added at any moment, to learn to automatically annotate documents. Furthermore, these annotations may be related to any kind of pieces of an html document (the whole document, a node, a set of nodes, or a text segment). The main functions that the tool enables are the following ones: creation of annotations associated with a range of text, editing/deleting existing annotations, clearing all annotations in the document, permanent storage of annotations, creation of global annotations to a web page, where the scope of the annotations are whole pages, visualization of RDF generated for the page (W3C's RDFa Distiller [35]), and ontology-based queries about properties that have been annotated. Inference is made in the taxonomies of concepts when the search is conducted by the annotation properties and automatic generation of semantic annotations.

In the system architecture layers (Figure 1), the white boxes represent system components and the shadow ones are related with semantic capabilities which are detailed below:

(i) In the core layer are placed the Operating System (that provides network services) and the Web Server.

(ii) The data management layer is made up of system components responsible for both content storage of Web documents and also the annotations on them
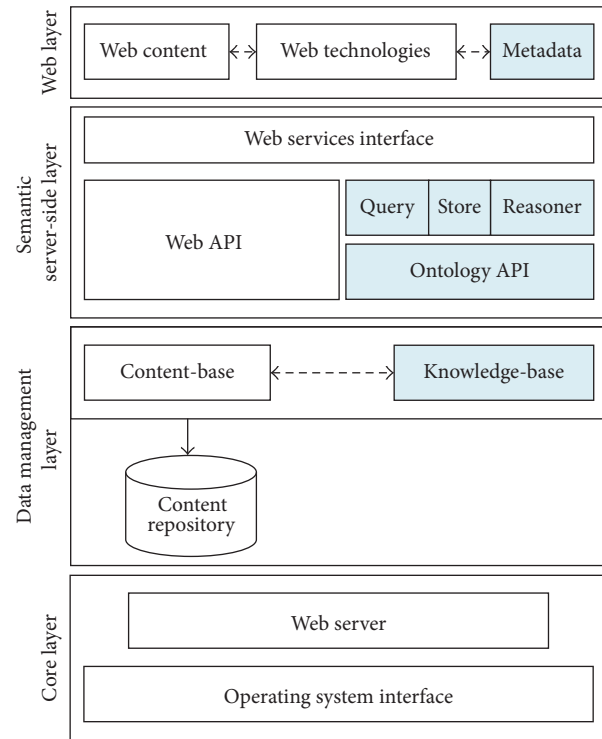

FIGURE 1: Architecture of FLERSA tool.

and the knowledge base consisting of the ontologies of the system.

(iii) The server-side layer is where server application services are developed. All message traffic between web clients requesting services and the programs that provide them takes place at this level. In this layer the implementation of programs that serve the web interface is carried out. The programs implemented here make use of programming libraries and Application Programming Interface (APIs) that provide the underlying layers. Among the most frequently used functions provided by these APIs the facility for storage and retrieval of information, facilities for working with visual objects in the front-end programming, and facilities for working with ontologies are worth highlighting.

(iv) Finally, the web interface layer is located at the uppermost level of abstraction of the system architecture, where the user performs all interaction with the semantic annotation tool. At this level the contents of web documents coexist with metadata and with web technologies in charge of modifying web documents in runtime to provide them with semantic annotations in the form of metadata and also to achieve timely message handling, by using server-side services to provide the functionality of the tool.

However the FLERSA tool presents a main limitation: the tool has a high coupling with Joomla! [12] as the CMS

is used as the underlying web infrastructure; that is, server-side (module) and client-side (server interaction) implementations cannot be easily adapted to another system. Thus, the collaborative capability, that is, the possibility of making multiple semantic annotations simultaneously by some users, depends on Joomla and it is not provided. Besides, although the most common environment for semantic web tools is a web browser, a lot of enterprise applications could need to integrate web annotations with annotations made on enterprise documents/data (e.g., for strategic scanning, technological watching, or social monitoring), which is not possible with the current FLERSA design.

4. A Software Architecture to Support Information Sharing and Collaboration

Collaborative systems are complex, this is challenging in analysis, modelling, and development [36]. One of the main tasks to be solved in collaborative systems is to maintain data consistency when they are simultaneously shared by several users [37]. Nowadays, in the absence of standardized methods for the synchronization of the shared data replicas, most of the proposed solutions are planned in an ad hoc manner. By taking into account the possibility of an increasing number of users and resources to be managed in very dynamic environments, this entails a higher complexity in the correct synchronization of these resources. Thus, a SOA 2.0-based architecture [15] has been proposed, which intent to provide a common basis for the consistent management of shared information in collaborative systems. It consists of two main services (Figure 2):

(i) *Monitoring Service.* This service gathers all events related with modifications on shared data. This information can fulfil several purposes, for example, version control or security logs. For the synchronization purposes, this information is required by the specific synchronization algorithm to be applied to know the occurrence and order of the modifications on the resource. The monitoring service is able to communicate under two different paradigms (SOA 2.0 [13, 38]): (1) the Publish-Subscribe paradigm, to know the modifications produced by the users on the shared data, and (2) the Request-Response paradigm, for example, when the synchronization service inquires the monitoring service about the modifications produced on a specific resource in a specific time interval. Thus, the use of an EDA ("Event-Driven Architecture") approach [39], specifically the concept of event, allows the developers to provide a reusable service. The monitoring service has been designed taking advantage of the low coupling in the communications between the sender and receiver provided by EDA; thus, it is able to monitor any kind of event. In this way, regarding resource monitoring, in the specialization of the platform it is only necessary to design the structure of the events that will be sent when the shared resources are modified.
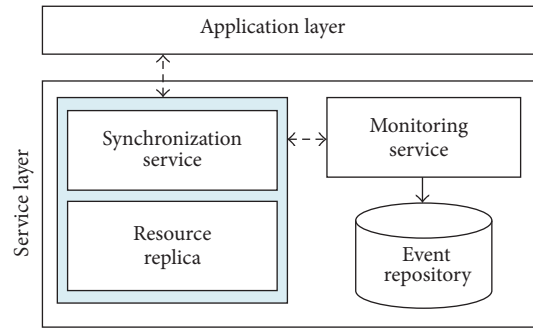

Figure 2: Architecture of the generic service platform to support sharing and collaboration.

(ii) *Synchronization Service.* The service platform proposed aims to be a generic platform that can be applied in any application domain within the collaborative systems. However, as the synchronization algorithms are dependent of the resource type and its specific nature and usage, it is not possible to provide a general service for the synchronization. Namely, the conflicts that could be generated in the concurrent modification of images are not the same as that in the modification of, for example, plain text, as well as the processes or policies to resolve them. For this reason, and regarding the goal of providing a reusable service, the synchronization service is designed as abstract service, which must be specialized according to each particular resource to be synchronized. This abstract service uses the monitoring service to obtain information about changes on the different replicas of the shared resource. In this way, and in order to provide a more generic service platform, it has been designed considering two different levels: (1) the common part related to manage the resource synchronization, which is identified and located into the abstract service and its composition with the monitoring service; in this way, the synchronization service, according to the information received from the monitoring service can detect the actions that have been applied to other replicas of the resource, but not applied to its associated replica; (2) the specialization of this service, where, once the inconsistencies are detected, they should be resolved. This level will depend on the requirements associated with the resource kind and the use of the resource in a specific domain (i.e., on the same resource different synchronization policies could be applied).

5. From FLERSA Tool to FLERSA Service

FLERSA tool has been developed as a module of Joomla! in order to transform this CMS into its semantic equivalent. This is the reason why there is a high coupling between the tool and the CMS. As a solution to this limitation, we propose to adapt FLERSA to the new architectural design followed by the SOA 2.0 architecture described in the previous section. This
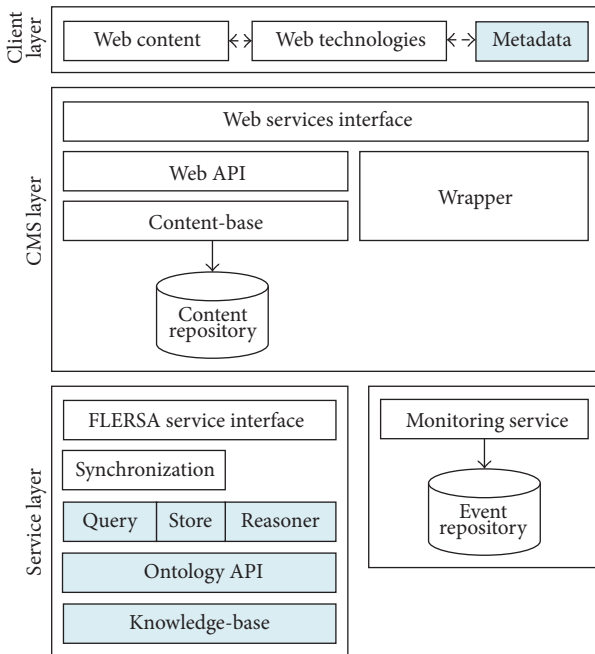
Figure 3: The SOA architecture for the FLERSA tool.

will lead to provide FLERSA with a clear differentiation and independence of the client-side, the CMS (Figure 3).

In order to adapt an existing tool or service to the proposed architecture, the developer must first clearly identify what is the functionality of the tool that must be implemented in the server-side. In the FLERSA case, this functionality is about the information retrieval, storing, and reasoning on the knowledge base, which will now be implemented in the specialization of the synchronization service, in order to take advantage of the architecture proposed. Currently, the services that support this architectural design have been implemented both in C++ and in C#. Nevertheless, the programming language is only a technical issue that does not affect the interoperability of the architecture, since standard protocols for exchanging information (e.g., SOAP) have been adopted, as well as communication approaches for loosely coupled components (e.g., EDA).

Once the server-side functionality of the tool is implemented, as the specialization of the synchronization service, it is necessary to identify which is/are the shared resource(s), the actions that users can perform on it, and the possible inconsistencies that can arise because of those actions. This is one of the most important steps in the adaptation process, owing to the fact that the correctness of the resource will depend on the right identification of the possible inconsistencies and the resolution policies applied. In the FLERSA case, the shared resource is a knowledge-base, whereas the actions that can be carried out on the resource are add, modify, and delete semantic annotations. Moreover, these annotations can be performed on the whole document, a node, a subset of nodes, or a text segment, and each of these elements can present several annotations. Therefore, the conflicts during the use of the FLERSA service can be caused

by deletions or modifications coming from different users on the same annotation. In order to solve this kind of conflicts, a version control has been implemented, where a deletion or modification is not permanent and it is possible to revert to the previous version of the annotation.

All of these actions are represented as events in the new architectural design. This will facilitate the management and broadcasting of the actions performed on the shared resources along the concerned entities. In the proposed platform, the BlueRose communication middleware [38] is used, which provides a Publish-Subscribe service and an interface for event managing. An event is represented by means of a pair topic-attributes, where each attribute is a pair key-value. In this way, the topic denotes the event type, which is unique in the system, whereas with the set of attributes it is possible to represent information of a wide range of complexity.

The flexibility and low coupling provided by the EDA approach have made it possible to design and implement a monitoring service, which is designed and implemented to monitor any kind of event. In the FLERSA case, three types of events taking into account the possible actions that a user can perform have been considered: add, modify, or delete an annotation. The generated events contain information related with the user who performs the action, the content related with the annotation (the whole document, a node, a subset of nodes, or a text segment), the timestamp, and information related with the semantic content of the annotation (see Figure 4). These events are stored in a NoSQL database (i.e., a nonrelational database), specifically MongoDB. The NoSQL systems arise to address the scalability problems of the traditional databases (i.e., relational) by means of a more flexible storage structure. Particularly, the absence of data schema allows storage of any information as a register with a key-value structure. This makes a NoSQL database ideal to store any kind of event generated in the system, whereas in a traditional database it would be necessary to create a new table to store each new kind of event. To this regard, the monitoring service translates the events from BlueRose format to MongoDB (JSON format) and vice versa.

With the server-side being implemented, as a specialization of the synchronization service, and together with the identified events, the functionality of the client-side must be additionally implemented. This can be done by creating a new application or adapting an existing one. In the latter, to make use of third-part applications, the common solution is to implement an intermediate entity (known as wrapper) that is capable of translating the petitions of the client to the server and vice versa, and thus it is also capable of adapting existing interfaces to the new service as well. In the FLERSA case, the client-side is located in the CMS module, where a wrapper function is implemented. In this way, if the CMS changes then only a new corresponding implementation of the wrapper function will be required in order to use the new FLERSA service. Web documents and metadata, like RDFa, are also in the client layer. These resources are downloaded to the device temporally when the client accesses the CMS.

In addition to the benefits initially mentioned obtained through the adaptation of FLERSA to the new architectural
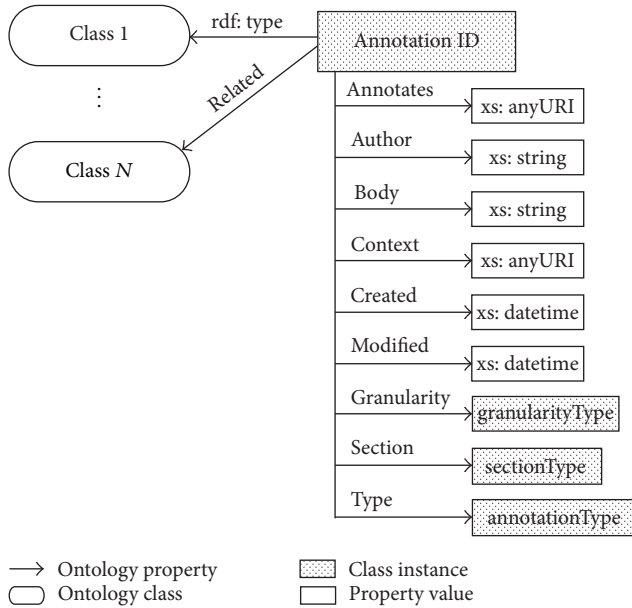
FIGURE 4: Graph showing an instance of the class "annotation" implemented by FLERSA [11].

design proposed, this SOA architecture also manages a number of additional events at system and infrastructure levels [15], as, for instance, the battery level of the device or network topology. This, together with replication and caching techniques, allows to provide a context-aware solution and therefore guarantee the quality attributes of the FLERSA service in AmI environments where the context conditions (e.g., disconnections) are continuous and they can affect the proper functioning of the service.

The integration of FLERSA in the proposed SOA architecture has strengthened three of the initial design requirements of the tool [11]. (1) *Requirement 2*. Collaborative design/user-centred: as a result of the integration, FLERSA service allows now the concurrent and distributed edition of HTML documents. (2) *Requirement 5*. Evolution of documents (document and annotation consistency): the synchronization service and the synchronization policies implemented guarantee the consistency of documents, as well as allowing users to know the existence of conflictive modifications and to recover previous versions of documents. (3) *Requirement 8*. Integration: now FLERSA functionality has been encapsulated in a service, which facilitates the reusability and interoperability of the tool.

6. Case Study

The FLERSA service can be useful in several scenarios in the eHealth domain, where the semantic web can help to retrieve information [40] and share the patient's medical histories created in different health centres and the semantic interoperability of distributed information systems in eHealth [41] for the collaborative decision support in disease diagnosis [42].

Specifically, the collaboration between various specialists for the diagnosis of patients with strange symptoms [43] is of special interest to illustrate the usefulness of FLERSA Service. On one hand, health is a wide knowledge area with a complex taxonomy, where any department or research group can define some particular protocol or vocabulary. Thus, to establish semantic relations between concepts or procedures is a mandatory step to achieve collaboration between different specialists or health institutions. On the other hand, it is unusual the existence of medical centres containing every health speciality, being generally more common the existence of specialised centres. Therefore, a tool that allows the distributed collaboration could provide clear benefits.

Figure 5 shows a general scenario about how users can work with FLERSA service and how they can collaborate. In Figure 5 are depicted two web servers and five users. Several users access the FLERSA service by using different types of devices. The web users use web browsers to access the service through web servers as front-end, while the rest of users use the service through an application deployed in a mobile device (e.g., smartphone or tablet) as front-end.

For existing applications, like web systems and other applications, a specific wrapper function for each one is needed, which translates the requests that web client makes to web site to the FLERSA service. In case of new applications, the wrapper function is not required, given that the interface of FLERSA service can be used directly by these applications.

Specific scenarios can be considered in Figure 5, for instance, if a doctor (e.g., "Web User 1" in Figure 5) creates a new semantic annotation in an HTML document (e.g., a patient report). This action will be propagated as an event through the system, and it will be registered by the monitoring service and applied by the FLERSA service to the knowledge base. In this way, this change will be reflected on the Drupal CMS as well as the Joomla! CMS and the rest platforms and technologies in the network. That is, FLERSA is now platform independent, which increments its scalability and interoperability. In this situation, the "Web User 1" could be collaborating with specialists that belong to another clinical centre and work on Joomla! (e.g., "Web User 2" and "Web User 3"), or even with mobile users.

The process that a user follows to create an annotation on an HTML document is depicted in the Figure 6. Note that an annotation can be fully or partially overlapped with others. FLERSA resolves this creating different SPAN sections with unique IDs.

However, under this new configuration, as it has been mentioned conflicts can occur due to concurrent editing. For instance, if "Web User 1" modifies an annotation on a patient report, while "Web User 2" deletes it, FLERSA service is in charge of maintaining a consistent version of the shared resource. In this case, the users could have created the conflict because of a distraction, a temporal disconnection of one of them, and thus the changes were not reflected to each other in time (i.e., they ignore the action performed by the other user), or intentionally. For this reason, the policy implemented in FLERSA service is to apply the last change but maintain the previous version of the annotations in order to avoid information loss, notifying to the users of the existence of a
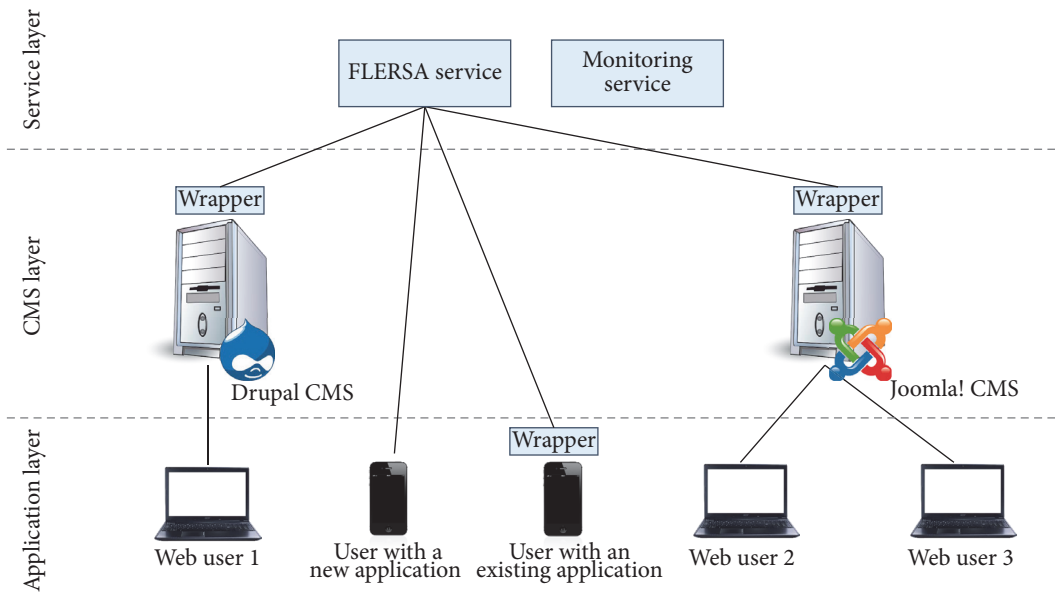
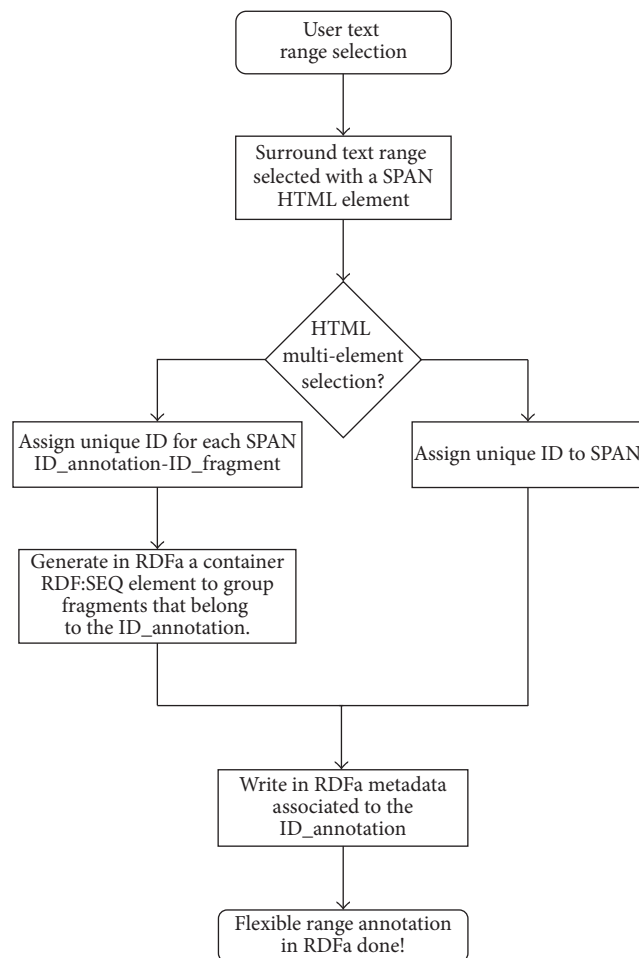FIGURE 5: System architecture for a FLERSA service deployment scenario.

FIGURE 6: Annotation process flow diagram [11].

potential conflict. In this way, the users can mediate to decide the correct state of the resource, but guarantee that both are consistent versions.

The architecture can react to context changes in order to guarantee the proper functioning of the service. This feature is of special interest for mobile users, where, for example, specialists would want to work together while they move across the hospital or they are travelling (e.g., by train), where the connection can be lost easily at certain points during that travel. In this case, the FLERSA service receives events regarding the connection loss, and together with caching techniques, it can allow users to the user continuing working transparently to the connection lost. The modifications made on the local copy of the resource are stored locally at the device during the disconnection, as a set of events. Later, when the connection is recovered, the FLERSA service will be able to synchronize the modifications that the user has made offline with the modifications of the rest of the users that were online. When FLERSA service (as specialization of the synchronization service) receives the request of reconnection from the client that was offline, together with the actions that he/she has performed in that period on the cache copy of the resource (as a set of events), it will inquire the monitoring service about what changes have been made in the main copy of the resource during that period of time by the rest of the users. Once FLERSA service has the two set of ordered events, it can detect the conflictive events (i.e., the conflictive modifications) and apply the versioning policy described above. In a similar way, the application could decide to start working locally when the battery of the device is low and synchronize the changes later on, when the energy connection is not critical.

7. Conclusions

In this paper, the evolution of the FLERSA tool towards a SOA proposal for consistent knowledge sharing and collaboration in AmI environments and IoT has been presented. The proposal provides FLERSA tool capabilities such as *scalability* [44], since the services are loosely coupled; *interoperability* [45], since it is possible to expose any existing data source like service and to implement workflows that allow exchange of information between different services and platforms through a communication protocol; and *business agility* [46], thanks to service reusability and the use of access and publication standards. The design of tools (like FLERSA) based on the proposed architecture aims to provide a general solution that opens up new possibilities for the development and deployment of other AmI applications by making use of different technologies and heterogeneous platforms in IoT. For example, the FLERSA service can be now deployed simultaneously in several Cloud providers and IoT nodes (smartphones, on-board car systems, etc.), obtaining benefits such as *higher availability*, where the resources or services will be also available wherever an Internet connection exists; *transparency*, where the user no longer has control of the geographical location of service; and the resources which can be increased or decreased as needed.

The SOA 2.0-based architecture for the FLERSA deployment is designed to provide support to the distributed collaboration in environments that exhibit discontinuous operation. To this regard, it combines caching and replication techniques together with a context-aware approach to provide a solution to address complex interactions between users in AmI environments in a transparent way. Therefore, it provides a common basis to handle the changing execution context in which AmI and IoT applications can be deployed.

Currently, we are still working in FLERSA with the focus on supporting automated annotation processes in the healthcare environment. We are collaborating with professionals from a public regional hospital in Granada, who have proven the low error rate while categorizing physiology documents [47]. Nowadays, we are involved in a research project with that hospital; the aim of the project under way is the use of the automated annotation system at anywhere and anytime, but limited to the hospital building environment, to generate metadata into the existing electronic clinical records, using concepts and properties from medical ontologies. In short, the multiple benefits offered by semantic web technologies would take advantage in the Hospital Healthcare Information Systems such as enhanced search capabilities, multidisciplinary queries, and alert autogeneration for specialist, among others.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

References

[1] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.

[2] S. Ricciardi, D. Careglio, G. Santos-Boada, J. Solé-Pareta, U. Fiore, and F. Palmieri, "Towards an energy-aware internet: modeling a cross-layer optimization approach," *Telecommunication Systems*, vol. 52, no. 2, pp. 1247–1268, 2013.

[3] United Nations, Internet of Things Global Standards Initiative, http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx.

[4] Y. Sun, H. Yan, C. Lu, R. Bie, and Z. Zhou, "Constructing the web of events from raw data in the web of things," *Mobile Information Systems*, vol. 10, no. 1, pp. 105–125, 2014.

[5] T.-M. Tsai, H.-K. Yu, H.-T. Shih, P.-Y. Liao, R.-D. Yang, and S.-C. T. Chou, "Ontology-mediated integration of intranet web services," *Computer*, vol. 36, no. 10, pp. 63–71, 2003.

[6] A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, and Y. Warke, "Managing semantic content for the Web," *IEEE Internet Computing*, vol. 6, no. 4, pp. 80–87, 2002.

[7] World Wide Web Consortium (W3C), "Amaya," January 2017, http://www.w3.org/Amaya/.

[8] B. Adrian, J. Hees, I. Herman, M. Sintek, and A. Dengel, "Epiphany: adaptable RDFa generation linking the web of documents to the web of data," in *Knowledge Engineering and Management by the Masses*, vol. 6317 of *Lecture Notes in Computer Science*, pp. 178–192, Springer, Berlin, Germany, 2010.

[9] D. Bonino, F. Corno, and L. Farinetti, "DOSE: a distributed open semantic elaboration platform," in *Proceedings of 15th IEEE International Conference on Tools with artificial Intelligence*, pp. 580–588, IEEE, Sacramento, Calif, USA, November 2003.

[10] Natural Language Processing Group, "Melita from the University of Sheffield," January 2017, http://nlp.shef.ac.uk/melita.

[11] J. L. Navarro-Galindo and J. Samos, "The FLERSA tool: adding semantics to a web content management system," *International Journal of Web Information Systems*, vol. 8, no. 1, Article ID 17024862, pp. 73–126, 2012.

[12] Open Source Matters (OSM), Joomla!, https://www.joomla.org/.

[13] P. Krill, "Make way for SOA 2.0," InfoWorld, http://www.infoworld.com/t/architecture/make-way-soa-20-420.

[14] F. Palmieri and A. Castiglione, "Condensation-based routing in mobile ad-hoc networks," *Mobile Information Systems*, vol. 8, no. 3, pp. 199–211, 2012.

[15] G. Guerrero-Contreras, J. L. Garrido, S. Balderas-Diaz, and C. Rodriguez-Dominguez, "A context-aware architecture supporting service availability in mobile cloud computing," *IEEE Transactions on Services Computing*, 2016.

[16] M. Buffa and F. Gandon, "SweetWiki: semantic web enabled technologies in Wiki," in *Proceedings of the International Symposium on Wikis (WikiSym '06)*, pp. 69–78, ACM Press, Odense, Denmark, August 2006.

[17] S. Schaffert, "IkeWiki: a semantic Wiki for collaborative knowledge management," in *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '06)*, pp. 388–393, Manchester, UK, June 2006.

[18] R. Schroeter, J. Hunter, J. Guerin, I. Khan, and M. Henderson, "A synchronous multimedia annotation system for secure collaboratories," in *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (e-Science '06)*, p. 41, IEEE, Amsterdam, Netherlands, 2006.

[19] G. Zhai, G. C. Fox, M. Pierce, W. Wu, and H. Bulut, "eSports: collaborative and synchronous video annotation system in grid computing environment," in *Proceedings of the 7th IEEE International Symposium on Multimedia (ISM '05)*, pp. 95–103, December 2005.

[20] J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R. R. Swick, "Annotea: an open RDF infrastructure for shared Web annotations," *Computer Networks*, vol. 39, no. 5, pp. 589–608, 2002.

[21] P. Saint-Andre, K. Smith, M. Wild et al., Jabber.org, http://www.jabber.org/.

[22] J. Knight, S. Cantor, D. Fisher et al., Shibboleth.net, http://shibboleth.net/.

[23] OASIS Group, *eXtensible Access Control Markup Language (XACML) Version 3.0*, 2013, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.

[24] S. Pallickara and G. Fox, "NaradaBrokering: a distributed middleware framework and architecture for enabling durable peer-to-peer grids," *Lecture Notes in Computer Science*, vol. 2672, pp. 41–61, 2003.

[25] K. Bontcheva, H. Cunningham, I. Roberts et al., "GATE teamware: a web-based, collaborative text annotation framework," *Language Resources and Evaluation*, vol. 47, no. 4, pp. 1007–1029, 2013.

[26] FIWARE Foundation, FIWARE—Core Platform for the Future Internet, https://www.fiware.org.

[27] M. Fazio, A. Celesti, F. G. Marquez, A. Glikson, and M. Villari, "Exploiting the FIWARE cloud platform to develop a remote patient monitoring system," in *Proceedings of the 20th IEEE Symposium on Computers and Communication (ISCC '15)*, pp. 264–270, Larnaca, Cyprus, July 2015.

[28] J. A. López-Riquelme, N. Pavón-Pulido, H. Navarro-Hellín, F. Soto-Valles, and R. Torres-Sánchez, "A software architecture based on FIWARE cloud for precision agriculture," *Agricultural Water Management*, 2016.

[29] S. Sotiriadis, L. Vakanas, E. Petrakis, P. Zampognaro, and N. Bessis, "Automatic migration and deployment of cloud services for healthcare application development in FIWARE," in *Proceedings of the 30th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '16)*, pp. 416–419, March 2016.

[30] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[31] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review Archive*, vol. 44, no. 5, pp. 27–32, 2014.

[32] D. Puiu, P. Barnaghi, R. Tonjes et al., "CityPulse: large scale data analytics framework for smart cities," *IEEE Access*, vol. 4, pp. 1086–1108, 2016.

[33] M. Giatsoglou, D. Chatzakou, V. Gkatziaki, A. Vakali, and L. Anthopoulos, "CityPulse: a platform prototype for smart city social data mining," *Journal of the Knowledge Economy*, vol. 7, no. 2, pp. 344–372, 2016.

[34] A. Gyrard, P. Patel, S. K. Datta, and M. I. Ali, "Semantic web meets internet of things (IoT) and web of things (WoT)," in *Proceedings of the 15th International Semantic Web Conference (ISWC '16)*, October 2016.

[35] World Wide Web Consortium (W3C), "RDFa Distiller," January 2017, http://www.w3.org/2007/08/pyRdfa/.

[36] M. Noguera, M. V. Hurtado, M. L. Rodríguez, L. Chung, and J. L. Garrido, "Ontology-driven analysis of UML-based collaborative processes using OWL-DL and CPN," *Science of Computer Programming*, vol. 75, no. 8, pp. 726–760, 2010.

[37] S. B. Davidson, H. Garcia-Molina, and D. Skeen, "Consistency in a partitioned network: a survey," *ACM Computing Surveys (CSUR)*, vol. 17, no. 3, pp. 341–370, 1985.

[38] C. Rodríguez-Domínguez, K. Benghazi, M. Noguera, J. L. Garrido, M. L. Rodríguez, and T. Ruiz-López, "A communication model to integrate the request-response and the publish-subscribe paradigms into ubiquitous systems," *Sensors*, vol. 12, no. 6, pp. 7648–7668, 2012.

[39] R. Meier, "Communication paradigms for mobile computing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 4, pp. 56–58, 2002.

[40] M. Bhatt, W. Rahayu, S. P. Soni, and C. Wouters, "Ontology driven semantic profiling and retrieval in medical information systems," *Journal of Web Semantics*, vol. 7, no. 4, pp. 317–331, 2009.

[41] P. Kataria and R. Juric, "Sharing e-health information through ontological layering," in *Proceedings of the 43rd Annual Hawaii International Conference on System Sciences (HICSS '10)*, pp. 1–10, Honolulu, Hawaii, USA, January 2010.

[42] E. Dogdu, "Semantic web in eHealth," in *Proceedings of the the 47th Annual Southeast Regional Conference*, ACM, Clemson, SC, USA, March 2009.

[43] F. Amato, A. R. Fasolino, A. Mazzeo et al., "Ensuring semantic interoperability for e-health applications," in *Proceedings of the International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS '11)*, pp. 315–320, IEEE, June-July 2011.

[44] T.-H. Yang, Y. S. Sun, and F. Lai, "A scalable healthcare information system based on a service-oriented architecture," *Journal of Medical Systems*, vol. 35, no. 3, pp. 391–407, 2009.

[45] C. Seo and B. P. Zeigler, "DEVS namespace for interoperable DEVS/SOA," in *Proceedings of the Winter Simulation Conference (WSC '09)*, pp. 1311–1322, Austin, Tex, USA, December 2009.

[46] N. Bieberstein, S. Bose, L. Walker, and A. Lynch, "Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals," *IBM Systems Journal*, vol. 44, no. 4, pp. 691–708, 2005.

[47] J. L. Navarro-Galindo, J. Samos, and M. J. Muñoz-Alférez, "A hybrid approach to text categorization applied to semantic annotation," in *Database and Expert Systems Applications*, vol. 7447 of *Lecture Notes in Computer Science*, pp. 39–47, Springer, Berlin, Germany, 2012.