Received 28 October 2014
Revised 30 January 2015
Accepted 31 January 2015

Generic framework for enriching services: a multiagent approach

Angela Carrillo-Ramos, Luis Guillermo Torres-Ribero,
María Paula Arias-Báez, Alexandra Pomares Quimbaya,
Enrique González, Julio Carreño and Juan Pablo Garzón Ruiz
*Department of System and Computing Engineering,
Pontificia Universidad Javeriana, Bogotá, Colombia, and*
Hervé Martin
Laboratoire d'Informatique, Université Joseph Fourier, Grenoble, France

Abstract

Purpose – This paper aims to present a detailed description of Agents for Enriching Services (AES), an agent-oriented framework that allows adapting a service in an information system. AES provides an adaptation logic that can be instantiated and extended to be useful in different domains. In previous works, we presented the adaptation mechanism of AES, which considers context aspects such as location, infrastructure; user aspects such as preferences and interests; and device aspects such as hardware and software features.

Design/methodology/approach – The first step was the definition of different profiles, mainly user and context profiles. Then the adaptation mechanism was defined, which considers these profiles. With this mechanism, the adaptation filters to apply them to the initial queries was specified. Finally, feedback was provided, which included implicit and explicit information from the user and the system. AES is an agent-based framework implemented in Java, using the multi-agent platform BESA and a rule-based engine Drools.

Findings – AES can be used as the starting point to adapt services by enriching them considering different stimulus whether they come from the environment, devices or user preferences.

Research limitations/implications – This work was tested in an academic environment and was only applied to enhance queries by using keywords. AES uses the query mechanism implemented in the system that invokes it.

Originality/value – This paper focuses on: an integrated view of AES including its formal description and details about its implementation. Particularly, it includes an exhaustive and formal definition of the filters used to create the adaptation rules and three different scenarios of the application of AES to adapt content according to user and context features. Finally, a comparison analysis is presented to highlight the strengths of our framework, specially its capacity of integration with systems that require providing user- and context-oriented services.

Keywords Adaptation, Enriched services, Multi-agent

Paper type Research paper

This paper is developed within the projects "AYLLU: Plataforma de Cooperación mediada por Agentes aplicada en un Contexto de E-learning Colaborativo" (supported by COLCIENCIAS register No: 1,203-489-25592) and "ASMA: Modelo de Adaptación Contextual en Sistemas Multiagentes". Both projects were supported by the Pontificia Universidad Javeriana (Bogotá, Colombia).

International Journal of Web Information Systems
Vol. 11 No. 2, 2015
pp. 246-268

1744-0084
DOI 10.1108/IJWIS-10-2014-0037

1. Introduction

The vast amount of information that a user must face in today's information systems has created the need for new mechanisms to help users to get the information that is really relevant to them. The process of adaptation contributes on this purpose. It enriches services with features that allow the adjustment of the information acquired, according to the characteristics of the context where an interaction between the user and a system occurs (Zemirli *et al.*, 2005).

The adaptation is helpful to minimize the amount of information provided to the user by selecting it according to her/his characteristics, access device and context (Gómez *et al.*, 2014).

In general, the adaptation of services has proved to be useful in different environments. As stated by Zhu *et al.* (2010), the adaptation is a key element in collaboration environments as well as in educational environments (Andrade *et al.*, 2007), where it has greatly improved learning processes. Knowing the student's relevant characteristics, for example, her/his learning style, the information can be displayed in the way that the student understands the information easier and faster. Another example of adaptive collaboration can be seen in the study by Lee and Field, 2005, where adaptive and automated collaboration aids in an Air Traffic Control environment where synchronized decisions must be made quickly and accurately by adapting the human–computer interaction to be more like a human–human interaction. The creation of adaptable software requires the awareness of external conditions and the ability to change the behaviour proactively.

There are different approaches and mechanisms to implement adaptation. Proposals like the Complex Manufacturing System (Mönch and Zimmermann, 2008) or the Adaptive Agents for Sequential Auctions (Ma, 2008), implement adaptation mechanisms specialized on a specific problem so the framework cannot be extended to other contexts that need services enrichment. Others like ATRACO (Kameas *et al.*, 2009) propose the use of more general approaches for adaptation, but using specific technologies that hinder the integration with other systems that do not have the same technology type (e.g. ontologies). All of these works allow improving the adaptation capability in their own specific areas (Torres-Ribero *et al.*, 2011; Cremene *et al.*, 2008). However, it is important to note that none of them are totally generic, that is, none allows the inclusion of multiple scenarios and multiple technologies to their adaptation process on any kind of services.

This paper presents a detailed description of a framework for adaptation called *AES* (Agents for Enriching Services) (Torres-Ribero *et al.*, 2011). Previous works presented different parts of design and implementation of the framework *AES* (Torres-Ribero *et al.*, 2011; Arias-Báez *et al.*, 2011a, 2011b). In this paper, we present a complete view of this framework, adding aspects such as:

- A description and comparison analysis of related works, highlighting the strengths of *AES*.
- A formal description of its components and the relationships between them.
- Implementation of the Filter rules, applied on an educational context.
- A new scenario of application.

AES can be used by any kind of information system, regardless the type of technology it uses, to enrich its services according to the specific requirements of the domain. The main contributions of *AES* are:

- its adaptation logic that can be instantiated and extended for any requirement;
- its flexibility because it can be used as starting point to adapt services by enriching them considering different stimulus whether they come from the environment, devices or user preferences in a generic way;
- *AES* can be used in different scenarios and considers context aspects such as location, infrastructure, socio-cultural characteristics; and
- it considers user aspects such as preferences, senses and interests to adapt the information and additionally, hardware and software features, connections type.

This paper focuses on a formalism to describe *AES* and some details about its implementation.

To present the *AES* framework, this paper is organized as follows: Section 2 presents related works about generic adaptation systems and adaptation systems using agents. Section 3 explains in detail the architecture of *AES* framework. Section 4 presents the use of *AES* in an educational environment. Section 5 describes the implementation of the *AES* Framework through agent-based components. Finally, Section 6 presents the conclusions and future work.

2. Related works

This section explores different systems created for adapting services. These systems use general features and components that can perform in different specifications.

Cremene *et al.* (2008) propose a type of adaptation system that focuses on the context. They argue that it is necessary to manage a context profile from observers who are responsible to decide which components to use, the way that they will be observed and extract information to adequate it to the context. Following the same goal, Bouyakoub and Belkhir (2008) present an adaptation platform for the multimedia documents presentation in *SMIL* format [The World Wide Web Consortium (W3C), 2001]. This platform manages the user profile for the adaptation of the multimedia content on heterogeneous devices (Barraza and Carrillo-Ramos, 2010).

Jiang *et al.* (He *et al.*, 2007) present an adaptation system content-based called Xadaptor that has adaptive mechanisms to display different types of content and organize those mechanisms using rules. It also manages the user profiles with information such as software and hardware features. Another generic framework that adapts the content to heterogeneous and mobile environments is CASHE (Chebbine *et al.*, 2005). It considers four profiles for adapting information: device, user, content and network restrictions. However, this approach is not flexible to work with other technologies and does not enhance general services.

Adaptation using agents is a widely discussed matter with different approaches. Kameas *et al.* (2009) use ontologies to enhance adaptation process. A different approach presented by Kasiolas *et al.* (1999) uses a large number of cooperative distributed agents that make the adaptation process by sharing acquired knowledge. Charrier *et al.* (2009) propose to organize the agent's behaviour adapting their interactions and the information stream according to their environment so they can effectively share knowledge. Eliassi-Rad and Shavlik (2003) turn user preferences into instructions for agents and compile them into neural networks allowing the agents to learn. Frameworks such as *PUMAS* (Carrillo-Ramos *et al.*, 2005) use adaptation through agents in a mobile environment; this adaptation is based on user preferences, features of his/her mobile device, and other aspects to adapt the results of the query; adaptation has also been applied in fields such as Information Retrieval as in the study by Barraza and Carrillo-Ramos (2010), or in commercial applications as in the study by Marin-Díaz *et al.* (2010).

Imam (2004) propose an approach of adaptation using intelligent agents that can be grouped into three categories: internal, external and complete adaptation. Additionally, resources used by an adaptation framework can also be considered when adapting information; these resources determine the quality of the outcome of an adaptation process (Cremene *et al.*, 2008); this approach defines different kinds of adaptation categories based on the elements that are subject to adaptation. Those examples of adaptation systems have many restrictions that do not allow them to be completely generic, it means, none of them considers multiple profiles to enrich the service, they are tied to a single technology and, in some cases, are attached to a specific scenario of use.

Table I shows a comparison between the mentioned related works that adapt information. The comparison criteria are: Multi Variable: the system considers more than one profile to adapt the information. Multi Technology: the system is flexible, it means, it is not limited to a specific technology. Extensible To Other Context: the system can be used in many contexts and related scenarios (e.g. context, user, access device information). Enriches Any Kind of Services: whether the system is able to adapt any kind of service required by the application that uses it or not.

Adaptation system/ framework	Multivariable	Multi technology	Extensible to other context	Enriches any kind of services
AdaMS (Bouyakoub and Belkhir, 2008)	Yes	Not specified	No. Only education	No.
Xadaptor (He *et al.*, 2007)	Yes	Not specified	Yes	No
CASHE (Chebbine *et al.*, 2005)	Yes	Not specified	Yes	No
Kameas *et al.* (2009)	Not specified	No	Yes. Knowledge environment	No
Kasiolas *et al.* (1999)	No. Only user needs	No. Tied to agents	Yes. Knowledge environment	No
Charrier *et al.* (2009)	Not specified	No. Tied to agents	Yes. Knowledge environment	No
PUMAS (Carrillo-Ramos *et al.*, 2005)	Yes	No. Tied to agents	No. Only mobile environment	Yes
Imam (2004)	Not specified	No. Tied to agents	Not specified	No

Table I.
Related work

3. Adaptation for enriching services

This section presents the *AES* framework. The goal of AES is to enrich services according to domain specific requirements. It can be used by different types of information systems.

3.1 AES adaptation model

The main idea of the adaptation model is that an external application sends a request to enrich a query to *AES*. Then, the system gets the relevant characteristics of that query (user, context, device profiles) and calls some filters to get high abstraction information that allows the generation of the enriched query that will be sent to the external application. Finally, the system receives an external application feedback with the purpose of knowing if the enrichment was satisfactory or not and to update the profiles information (Figure 1). A detailed description can be found in the study by Arias-Báez and Carrillo-Ramos (2012).

AES is part of a project called *ASMA*. One of the requirements of this project was to provide the user with information tailored to his/her characteristics and needs, as well as his/her context. Thus, the idea of creating a framework that considers different features of the user, his/her access device and context with the purpose of providing relevant information on the right moment, place and device, considering who the user is and what he/she really needs of an information source or system was born.

As a pre-condition of the system that uses this framework is that it must have defined the adaptation mechanisms and filters to be considered to generate a new enriched query. We will present the design of *AES*, its components, their responsibilities and relationships (Section 3.2) and its prototype using software agents (Section 4).

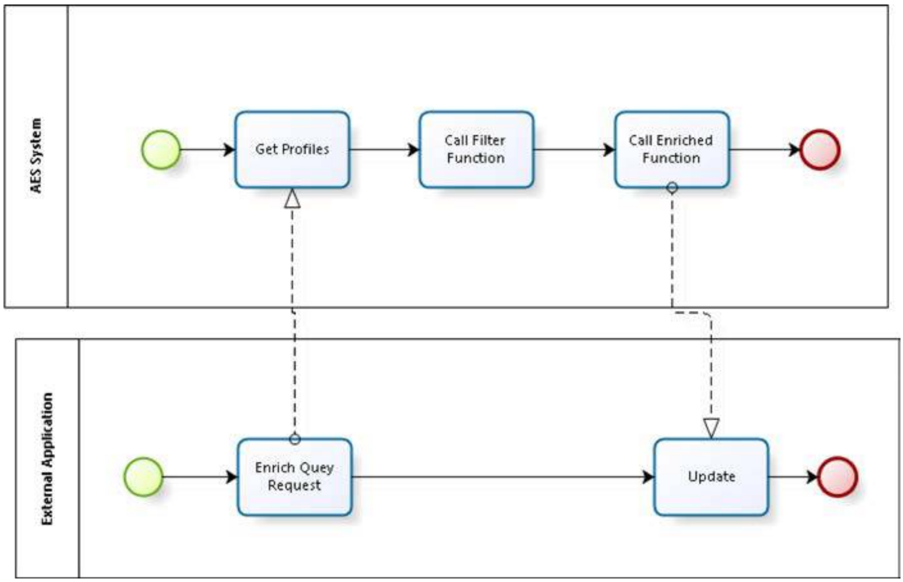


Figure 1.
AES general workflow

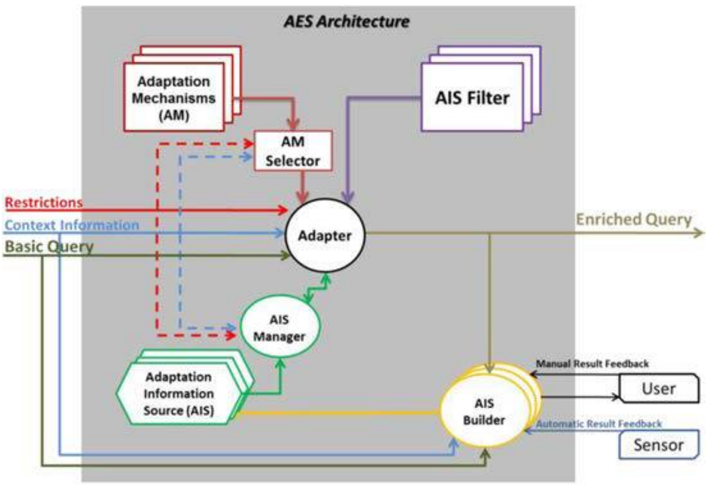
3.2 Architecture

The *AES* model is shown with its inputs, outputs, components and relations in Figure 2. The components that are represented with circles are dynamic components, while the components that are represented with rectangles are static components. A detailed description is in the study by Torres-Ribero *et al.* (2011), Arias-Báez *et al.*, 2011b).

3.2.1 System inputs and outputs. For the *AES* design, the following two definitions were considered:

(1) *INPUTS*: Q: the request of the user to the adaptation module:
   - *A.I. Context*: External information including the user's device, his/her context features when the interaction between user and system appears, location and so on.
   - *Particular restrictions*: Because the interaction between the user and system can be affected by certain limitations or restrictions; in some cases, several variables of the context can change. For this reason, the system should receive such special exceptions.
   - *Manual result feedback*: It is necessary to make feedback to the system provided by the user's initiative on quality and compliance expectations of the response (e.g. the user experience related to the enriched response generated).
   - *Automatic result feedback*: Observations made by the system related to previous adaptation processes or questions that have been solved without the user's initiative (e.g. system deductions).

(2) *OUTPUTS*: Enriched Q: the system generates a new enriched Query with the elements that have the system, according the system input. In case the query entered cannot be enriched, the enriched Q is equal to the initial query Q.



Source: Torres-Ribero *et al.* (2011)

Figure 2. AES architecture model

3.2.2 Static components. Some components are not autonomous and they are invoked as methods:

- *Adaptation mechanism*: Adaptation libraries to enrich the query Q; it consists of an enrichment function (Φ) and a list of filters that can be handled by Φ.
- *Adaptation mechanism selector*: Component responsible for selecting the Φ function.
- *Filters*: Function which parameters correspond to a profile list of the same category and determines through subcategories which field of the user profile is used to filter information. Their result is high-level information that will be used to enrich the query Q.
- *Adaptation Information Sources (A.I.S.)*: Each A.I.S. corresponds to a profile (set of relevant features related to represent a concept). That profile has a category (e.g. user, location, device, group, etc.) and fields (e.g. name, description) that are represented as subcategories and an ID. The next examples are part of A.I.S.; however, they are a representation that can be implemented in any kind of technology, e.g. ontologies. A detailed description can be found in the studies by Carrillo-Ramos *et al.*, 2011; Arias-Báez and Carrillo-Ramos, 2012:
  - *User*: Profile that contains user relevant features such as his/her basic data, preferences (outcome, collaboration, presentation, location and activity preferences) and interests.
  - *Location*: This profile contains environmental features according to the interaction between the user and system, such as the user location, infrastructure and available resources.
  - *Device*: This profile contains the relevant features that the user device has such as the connection type, hardware (screen resolution, memory, processor and autonomy), software (browser, applications and operative system).

Profile definition depends on the system that will use the framework *AES*. Internally, each characteristic of the profile is represented by a tuple (attribute, value and number), where the attribute is the aspect to be evaluated (e.g. colour), the value can be one or more information fields of the possible values the attribute can take to describe the context (e.g. red) and the number corresponds to the quantification of the value priority of that attribute with respect to what the user believes or is inferred from the interactions between the user and the system (e.g. the system can deduce that as the user has chosen to change the background colour to red in 60 per cent of the time, his/her favourite colour is red). The *FIA* may have multiple attribute values.

3.2.3 Dynamic components. They are autonomous components that communicate among themselves and have the ability to invoke the static components:

- *A.I.S. Manager*: Considers adaptation information needs of the adapter; this component looks for the AIS to provide information, extracts it and returns it to the Adapter.
- *A.I.S. Builder*: Its purpose is to provide feedback and update the AIS through processing of the Manual Result Feedback and/or Automatic Result Feedback (see System Inputs). These updates may correspond to a base value or a set of deductions and they are implemented by the method “update”.

- *Adapter*: This is the main component of the system, responsible for using all the components to enhance the request (i.e. to generate the Q enriched from the Q and Adaptation Information.) This provides the parameters (AIS) to the filter. Adapter gets the AIS with a request to the AIS Manager.

3.2.4 Formal description. Let AS be an Adaptation Service provided by *AES*. AS contains the components and functions that will produce, given the specified inputs, an enriched output that will consider the original entries related to the context and some previously obtained data useful for adaptation purposes.

AS receives three inputs:

(1) $Q \in \check{Q}$ where Q represents a Basic Query and $\check{Q}$ represents the set of possible basic queries.

(2) $C \in \check{C}$ where C represents a Contextual Information and $\check{C}$ represents the set of possible Contextual Information.

(3) $R \in \check{R}$ where R represents a Contextual Restriction and $\check{R}$ represents the set of possible of Contextual Restrictions.

The output of the AS is $EQ \in \check{(EQ)}$, where EQ represents an Enriched Query and $\check{(EQ)}$ represents the set of possible Enriched Queries.

The definition of an AS service is modelled as a function adaptService that relates the query and its context to produce an enriched query:

$$\text{adaptService}: \check{Q}, \check{C}, \check{R} \rightarrow \check{(EQ)} \quad (1)$$

Additionally, the AS service has three main components: the Adapter (Adap), The Adaptation Information Manager (AISMgr) and the Adaptation Information Builder (AISBldr). AS components represented by a tuple are:

$$AS = \langle \text{Adap}, \text{AISMgr}, \text{AISBldr} \rangle \quad (2)$$

The Adap is composed of a set (represented in square-brackets) of Adaptation Mechanisms (AM) and an Adaptation Mechanism Selector (AMSelector) function. The AMSelector function receives as input C and R; based on the information provided by those inputs plus an additional set of Adaptation Information Source (AIS) input provided by the AISMgr component, the AMSelector selects from the set of Adaptation Mechanisms one of the Adaptation Mechanisms that best suit the given C and R.

The adapter can be described as:

$$\text{Adapter} = \langle [\text{AM}], \text{AMSelector}(\,) \rangle \quad (3)$$

The AMSelector function can be described as:

$$\text{AMSelector}: C, R, [\text{AIS}], [\text{AM}] \rightarrow \text{AM} \quad (4)$$

The set of AIS is obtained through the AISMgr component of the Adapter. The AISMsgr component manages the Adaptation Information Sources (AIS) available on the service. Each AIS contains a category and a set of attributes that describes that category.

When the AMSelector function is invoked, the function requests the AISMgr, by sending C and R, the set of AIS that applies to the given C and R.

The AISMgr is composed by a set of adaptation information sources (AIS) and a search AIS function.

The AISMgr can then be described as:

$$\text{AISMgr} = \langle [\text{AIS}], \text{searchAIS}(\,) \rangle \tag{5}$$

The searchAIS function takes the context and the particular restrictions as input and selects among the set of available AIS, the subset of AIS that best fits the given C and R. The searchAIS function can then be described as:

$$\text{searchAIS: C, R}, [\text{AIS}] \rightarrow [\text{AIS}] \tag{6}$$

Once the set of AIS has been obtained from the searchAIS function, the AMSelector function goes through the set of AM searching for the most adequate AM given the obtained set of AIS.

Each AM is composed of a set of filters and an enriching function. Each filter has a category that must match an AIS category and a filtering function that will generate, according to the attributes in the AIS, High Abstraction Information (HAI).

An adaptation mechanism (AM) can be described as:

$$\text{AM} = \langle [\text{F}], \text{enrichQuery}(\,) \rangle \tag{7}$$

A filter (F) can be described as:

$$\text{F} = \langle \text{category}, \text{filterAIS}(\,) \rangle \tag{8}$$

The filterAIS function can be described as:

$$\text{filterAIS: AIS} \rightarrow \text{HAI} \tag{9}$$

The enrichQuery function in the AM combines the given context C, the particular restrictions R, the original search Query Q and the set of each HAI given by each filter to produce an enriched query EQ:

$$\text{enrichQuery: Q, C, R}, [\text{HAI}] \rightarrow \text{EQ} \tag{10}$$

Once the Enriched Query (EQ) has been produced, it is sent to the last component, the AIS Builder (AISBldr) which will take the generated Enriched Query (EQ) and/or through sensor information (S) and User information (U) will determine whether the enriched query efficiently combined the different contextual elements and restrictions and will update the set of AIS that were used during the process in order to provide a better answer in the future.

The AIS builder (AISBldr) can be described as:

$$\text{AISBldr} = \langle \text{updateAIS}(\,) \rangle \tag{11}$$

And the update AIS (updateAIS) function can be described as:

$$\text{updateAIS: Q, C, R, [AIS], EQ, U, S} \rightarrow \text{[AIS]} \quad (12)$$

$$\text{enrichQuery(Q, C, R, [HAI]): EQ} \quad (13)$$

$$\text{enrichQuery(Q, C, R, [F}_1\text{.filterAIS(AIS}_1\text{),}$$
$$\text{F}_2\text{.filterAIS(AIS}_2\text{),...F}_N\text{.filterAIS(AIS}_N\text{)): EQ} \quad (14)$$

$$\text{enrichQuery(Q, C, R, [F}_1\text{.filterAIS(searchAIS(C, R, [AIS],}$$
$$\text{F}_2\text{.filterAIS(searchAIS(C, R, [AIS])),...]): EQ} \quad (15)$$

3.3 Interaction protocols

Figures 3 and 4 present the messages between agents and their interaction order to enrich a query (Q). The interaction between components is determined by the framework, so it does not need to be implemented by the programmer; however, it is important to know which are the main components and how they interact to enrich a query.

Figure 3 shows a description of the called methods:

- an external application requests to enrich a Q and sends the context, the restrictions and the actual Q to the Adapter Agent as parameters of the enrichQuery method.

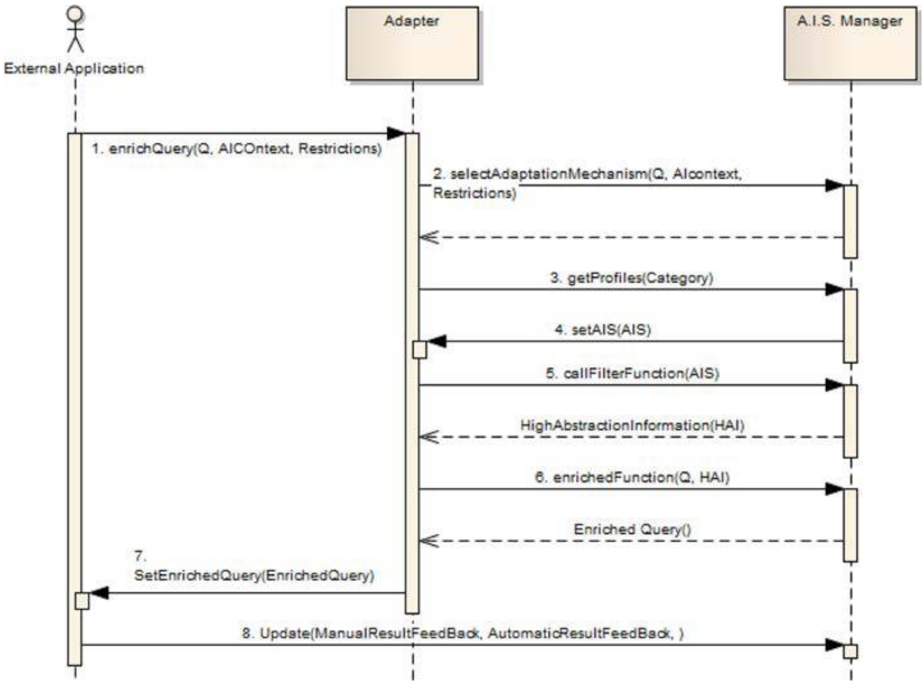


Figure 3. Collaborative diagram of the framework

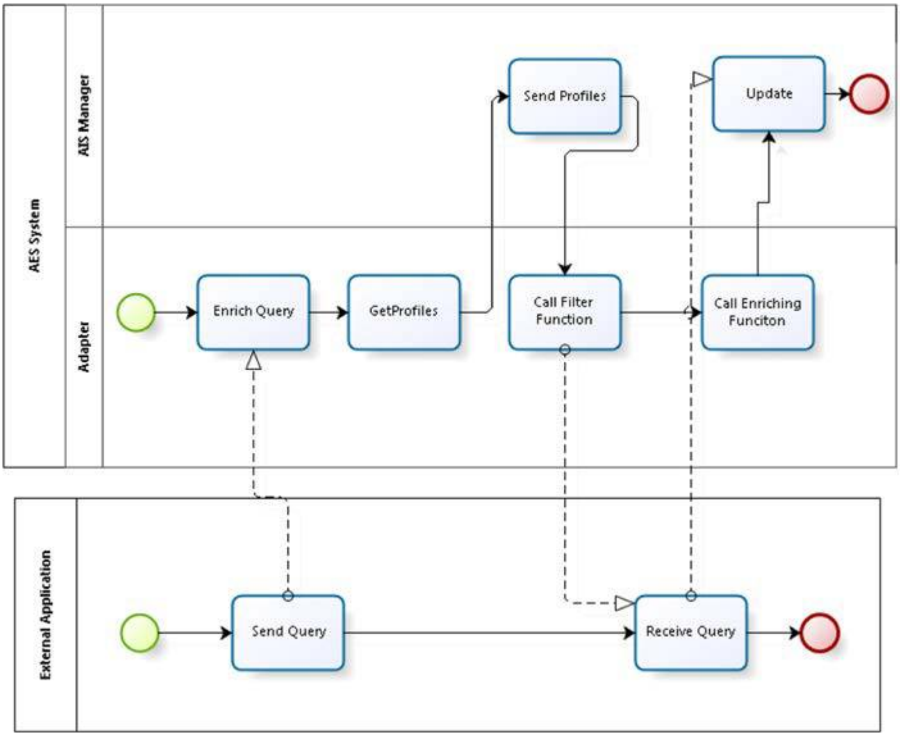


Figure 4.
Activity diagram

- The Adapter selects, according to its implemented logic, the best Adaptation Mechanism taking into account the Q, the context and the restrictions.
- The Adapter sends to AIS Manager a request for AIS through the getProfiles method.
- The AIS are returned and those AIS will fill the filters inside the Adaptation Mechanism selected.
- The Adapter generates high abstraction information with each filter.
- The Adapter enriches the Q using the high abstraction information obtained earlier.
- The enriched Q is returned to the external application that called the enrichQuery method initially.

Figure 4 displays the internal structure for each component as an activity diagram.

The algorithm used by the framework is shown in Figure 5. This algorithm represents in a deeper way how each component is called and used in the predefined interaction protocol between components (Torres-Ribero *et al.*, 2011). As shown in Figure 5, in line 4, the *getAISInformation* function will select the profiles that best fit, according to the context and the initial query. Those profiles will be used to determine the best adaptation mechanism. In line 5, the function *findBestAdaptationMechanism* will select among the available adaptation mechanisms, the one that best fits the criteria

Algorithm 1 AES Basic Algorithm

```
READ basic_query
READ particular_restrictions
READ context
CALL getAISInformation with basic_query,
     particular_restrictions, context RETURNING
      ais_List
CALL findBestAdaptationMechanism with
     basic_query, particular_restrictions,
     context, ais_List RETURNING
     best_adaptation_mechanism
FOR each ais_Filter in
     best_adaptation_mechanism
 FOR each ais_Information in ais_List
  IF ais_Filter applies to ais_Information
   CALL ais_Filter_doFilter with
        ais_Information RETURNING
        high_Abst_Info
   STORE high_Abst_Info in high_Abst_Info_List
  END IF
 END FOR
END FOR
CALL best_adaptation_mechanism_enrichQuery
     with Basic_Query, Particular_Restrictions,
      Context, high_Abst_Info_List RETURNING
     enriched_Query
```

Source: Torres-Ribero *et al.* (2011)

Figure 5.
AES basic algorithm

the framework user gives. Once the best adaptation mechanism is selected the associated filters will be read and those filters will be applied to the corresponding profiles. Each filter will obtain from the basic profile data high abstraction level information, through inference, deductions or conditionals.

Once all the filters have been applied, the function *enrichQuery* is called and it will take the high abstraction level information returned by each applied filter. It combines that information with the context and the basic Q in order to produce an enriched Q.

4. Development

Considering the requirements of *AES*, an analysis of different technologies allowed us to conclude that software agent technology is the most appropriate approach for developing it. The process for adapting information requires the analysis of large amount of information in a rather short period of time. In addition, the environment may change rapidly and those changes must be taken into account as they happen; software agents provide the following properties (Zemirli *et al.*, 2005), allowing it to be an adequate solution for issues present in an adaptation process:

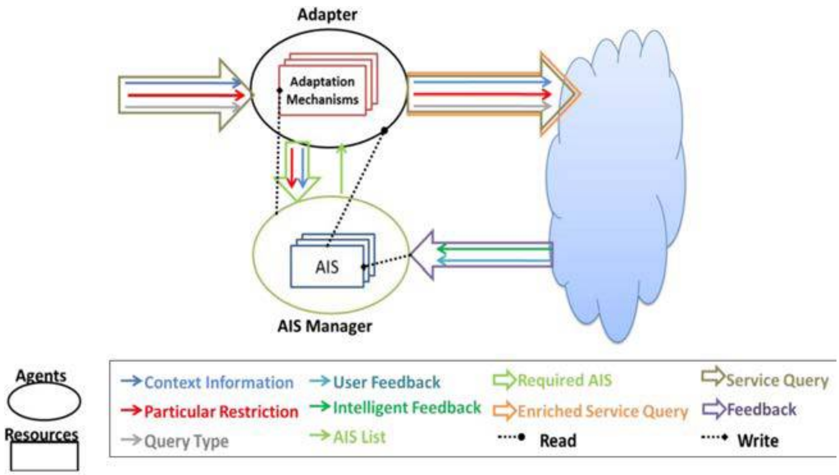
- *Autonomous*: Agents are autonomous to the extent that they make their own decisions without direct human intervention.
- *Sociable*: Agents can interact with other agents using a specified agent communication language to cooperate and carry out complex tasks.
- *Reactive*: Agents perceive and respond to changes (events) that occur in their environment.
- *Pro-active*: Agents can show goal-oriented behaviour by taking initiative over their actions.

The following sections show the agent model for the AES architecture, followed by the description of each category of components modelled. Figure 1 shows the different components in the *AES* architecture; Figure 6 presents the agent representation of the architecture. Circular-shaped components are Active Components and Rectangular-shaped components are Inactive Components. Inactive components can be seen as resources used by Active Components, i.e. an Inactive component represents a determined state, stores information and may include a method that analyzes the stored information; however, the method cannot start by its own, it is required that an Active component indicates the Inactive component when to execute the method and also provides the required parameters that method will use.

Note that Interactions are similar to interactions previously shown in Figure 1, to represent agents and their related resources, those resources represented as rectangle-shaped components are shown in its corresponding agent, represented by circle-shaped components. Filters are considered as part of Adaptation Mechanism so they are not shown in Figure 6. Another difference is the absence of the Adaptation Information Builder, the reason for this change is that the AIS Manager Agent will include responsibilities of both AIS Manager and Adaptation Information Builder.

4.1 Components

Active components are implemented as software agents, which have the ability of being proactive and to communicate between them. A description of this interaction is represented in Figure 6. There are two main Active Components in the framework, both components can react proactively to an event in their environment and modify their behavior according to the context and profiles they handle: The *Adapter* Agent's state contains the Adaptation Mechanisms and AIS Filters used by the framework; its behavior includes reacting to an incoming query, sending a profile request to the AIS Manager, receiving profiles sent by the AIS Manager, executing each filter's filtering method and finally, executing the enrichment method to generate the output. The *A.I.S.*



Source: Torres-Ribero *et al.* (2011)

Figure 6. Agent model implementation of AES architecture

Manager's state is composed by recent profiles, the context and restriction information. The AIS Manager behavior is to react to the Adapter's request of profiles, looking for them in a persistence component (if any) or in its state, and updating profiles upon feedback. Feedback sources are shown in Figure 6 as a cloud to represent that feedback's source can be different components, whether it is a user, another agent or any other possible source.

An Inactive Component represents a determined state. It stores information and may include a method that analyzes the stored information; however, the method cannot start by its own, it is required that an Active Component indicates when to execute the method and with which required parameters. Inactive components of the *AES* framework are: *A.I.S. Filters*: contain information regarding their associated AIS and the filtering method. *A.I.S.*: contain information regarding the elements that interact with the enrichment process. *Adaptation Mechanisms*: contain associated AIS Filters and a method to enrich the query based on the high abstraction information, the initial service query, the context and the particular restrictions.

4.2 Enriched service development

This section describes a methodology to create adaptive services using the *AES* framework. It first centres in how the analysis of a service must be made to correctly represent it using the framework.

The first phase must be the analysis phase, in which the user (administrator user) must first identify services he/she would like to enrich; for example, in a group creation service, the opportunity to enrich the selection of its members based on their abilities is one enrichment opportunity. After identifying the enrichment opportunities, the steps for the enrichment must be characterized and put together along with the service model.

Once these steps are made, the user must identify the profiles required, different features that will help enriching the services, the AIS Filters required, i.e. all of the possible high abstraction information that can be inferred from the context or profile and, finally, the Adaptation Mechanisms that will use those filters and how the user will merge the high abstraction information with the original input.

Identified elements must be mapped into all the inactive components of the framework. The basic elements are AIS; the *AES* Framework provides flexibility in the persistence method that will store those AIS. After the AIS have been created and stored, the next step is to create AIS Filters and associate them with relevant profiles and implementing their filtering method to obtain high abstraction information. This method can be implemented using any mechanism that the user determines, whether it is a simple logic statement or a complex neural network logic. The framework is generic and allows the implementation of multiple technologies in the filtering method. Once the filters have been designed and created, the Adaptation Mechanisms must be implemented. The number of Adaptation Mechanisms depends on how many Filter combinations are needed to achieve the enrichment of a basic query. The next step is to create the enrichment method, which, based on the information returned by the filtering method in each Filter, combines them with the information contained in the input and generates the enriched query. The enrichment method can also be implemented using different methods to associate information and produce information enrichment. Finally, a test should be made by sending a service query with different context and restrictions to validate the framework's functionality.

To use the framework, a series of components must be built based on the basic components provided, i.e. Adaptation Mechanisms, Filters and Profiles. The latter (Figure 7) represents the basic building block that the framework uses and it must be implemented as an extension of the Profile provided in the framework taking into account a profile category (e.g. User profile, Location Profile, *etc.*). Each profile may have all the adequate attributes the framework user considers fit to its context and also he/she is able to create as many profiles and categories he/she needs in the application context. In the study case, we implemented three types of profiles: device, location and user (see profiles examples on Section 2.2.2: Static components) (Arias-Báez and Carrillo-Ramos, 2012).

Other components are AIS Filters (Figure 8). Each filter must have a category that matches one of the categories used in the profiles and the name of the fields it access. This helps the filter to identify which fields are the ones relevant to the filter. Each filter inherits from the basic one provided in the framework; it must override a method called *doFilter*(). This method will implement the logic behind the inference system. Each *AIS Filter* will filter those profiles with the same category. An AIS Filter can be seen as a decision-making process that is based on a characteristic of the AIS it handles; for example, if an *AIS Filter* handles an AIS related to a user profile, it may return a different result according to the user's gender; for instance, if the user is male, the High Abstraction Information will contain different information than the information shown if the user is a female.

In the Study case, a filter for each profile type was implemented, a location filter that based on the building will tell the user the quality of the wireless connection, a device filter that will tell the resolution of the screen and, finally, a user filter that will tell what level of expertise he/she has on a particular subject. The *doFilter* method was implemented using conditionals based on various fields of each profile.

Finally, Adaptation Mechanisms (Figure 9) were implemented as subclasses of the Adaptation Mechanism class provided by the framework. These adaptation mechanisms have a set of associated filters; each adaptation mechanism may be different and must implement three main functions: *applyRestrictions*, *enrichQuery and returnFilters*. The *applyRestrictions* function is the function that takes into account the particular restrictions and applies them to the context so that certain filters are executed or not. The *returnFilters* function will return the filters the adaptation mechanism uses

Figure 7.
Profiles example

Figure 8.
Filters example

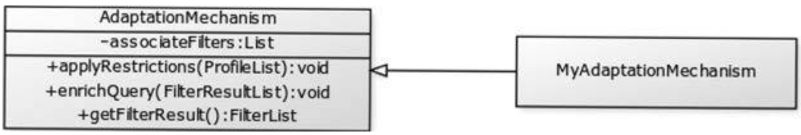


Figure 9.
Adaptation mechanism example

taking into account the new context obtained from *applyRestrictions*. Finally, the *enrichQuery* function is implemented as the function that takes all of the high abstraction information that each used filter inside the adaptation mechanism generates and mix them with the original query to make the enriched query. The implementation of this last method may be just appending each filter result or more complex decisions based on the technology the user needs.

An *Adaptation Mechanism* is, in essence, the components that interpret all the high abstraction information returned by each associated filter and merges them with the original query to produce the enriched query; for example, the query John Made was: "Macroeconomics Books" With the high abstraction information returned by the filters we can know that John is an expert on macroeconomics, that he learns best visually and that he needs low resolution pdf files to see them in his device. *The Adaptation Mechanism* takes this information and enriches the query to: "Low Resolution Macroeconomics Books for experts, format: PDF". To implement an *Adaptation Mechanism* the basic methods described above must be implemented in the class that inherits from the basic Adaptation Mechanism class provided by the framework.

5. Study case

This section presents different applications of AES that validated its adaptation services and its generic capabilities. We defined three scenarios:

(1) adaptation in a learning environment;

(2) adaptation for conforming groups; and

(3) adaptation for agent-based cooperative services.

Since Education is a domain in which information adaptation plays a key role by considering a student's particular characteristic features and context to provide better learning objects (Baldiris *et al.*, 2012), *AES* was used in a study case which goal consists in selecting a Virtual Learning Object (VLO) to a certain student with the purpose of giving a personalized VLO to a student according his/her learning style, preferences and access device features. A detailed description is in the study by Torres-Ribero *et al.*, 2011.

The initial query consists in select a VLO for a certain student: suppose that the professor needs to select a math VLO for the student John that best suits his characteristics. John enters into the system through his mobile phone (IA Context). The different profiles that the system stores are: student profile and the device profile. From these profiles, the system selects those who can enrich the initial Query (using the Profile Manager) for this case, are selected John (from Student profiles) and RT50 (from device profiles).

The first filter receives John's profile that contains information such as his name, birth date, learning styles, display and format preferences. That filter takes the relevant information according to the context. The second filter receives the device profile that has information about it reference, battery levels, supported formats and also takes the relevant information to generate high-level information (Figure 10).

High abstraction information is generated through the use of business rules. The first filter shows that John has an active-visual learning style (according to the Felder-Silverman test) and returns the type of active-visual material that matches the display preferences of John and his educational context (Higher education). The second filter makes the same process and returns the size of the John's device (small) from its resolution.

Considering that the students are in higher education, his learning style is Active-visual, that is, he learns through active visual stimuli, and knowing that his device has low resolution and small size, the system shows that the selected VLO should be dynamic and their size (in Kb) should be small (enrichment function). After receiving the result of each filter, the system executes a method in the adjustment mechanism on these results, which determines the importance of each filter.

If the device filter is executed and then the student's filter, the resolution of the device is low; therefore, the system deduces that the VLO to be presented must be low resolution and low weight (Kb). Additionally, according to the fact that the type of information that best suits John learning style is visual, the system deduces that the VLO should be interactive and contain figures (Adapter). The adapter produces an enriched query comprising the active-visual material to match the preferences of John (Enriched query). For example, simulations, charts, experiments and self-assessments, low-resolution figures or text, for Higher education, topic math (Figure 10).

To carry out the scenario proposed above, the AIS component, the Filters and the Adaptation mechanisms were implemented. The AIS used for this scenario were the user profile and device profile. However, it was necessary to create the student profile with attributes related to the educational context:

- learning style, approach or learning environment that enhances student performance. These styles were taken from the learning styles of the Felder-Silverman;
- *Display Preferences*: Depending on the device size, what type of material the student prefers to display information; and
- *Preferred Format*: According to display preferences of the student (e.g. video), what type of formats the student prefers (e.g. wmv, mp4, etc.).

The next step was to create the filters. Each filter has a category that corresponds to a profile type. Filters are applied only if its type matches the profile type. Therefore, the filters implemented were student Filter and device Filter according to the AIS created.

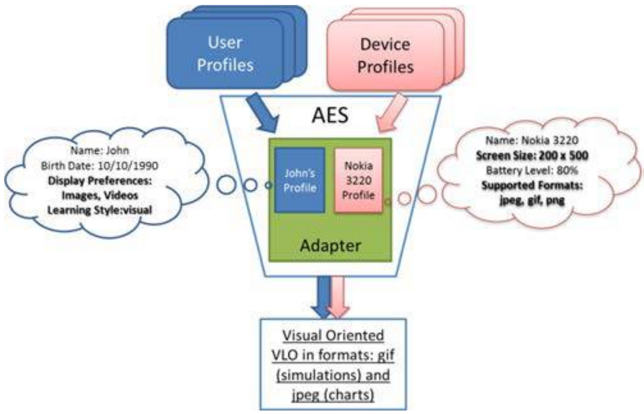


Figure 10. Separation of relevant information within the student filter and the filter device (in bold)

These filters were implemented using a rule engine called Drools (Red Hat, 2006), which allows the implementation of rules to generate deductions.

The first set of rules was based on the student's learning style, taking into account the IEEE LOM standard for VLO.

The first rule (Figure 11) takes into account user characteristics: a student who prefers active learning; this is, a student who learns through tasks, group discussions and similar activities; and learns faster when the information is represented visually.

The second rule (Figure 12) considers the characteristics of a student who prefers active learning and learns faster when the information is presented verbally as a lecture or discussion.

The third rule (Figure 13) applies to students that rather have a reflective learning, i.e. students that like to be isolated from distractions while learning, repeating the information and learn faster to visual stimuli.

The last rule (Figure 14) is applied to students that have a reflective learning and are prone to learn from a verbal source of information such as audio.

```
Rule: Visual - Active
WHEN
$student: StudentProfile(studyStyle == "Active") and StudentProfile(learningStyle == "Visual")
$vlo: VLOType(learningMaterial=="Simulation")
THEN
for(int i=0; i<$vlo.getVloVisActStyleArray().length; i++)
    $vlo.addMaterial($vlo.getVloVisActStyleArray()[i]);
```

Figure 11.
Visual active rule

```
Rule: Verbal- Active
WHEN
$student: StudentProfile(studyStyle == "Active") and StudentProfile(learningStyle == "Verbal")
$vlo: VLOType(learningMaterial=="Exercise")
THEN
for(int i=0; i<$vlo.getVloVerActStyleArray().length; i++)
    $vlo.addMaterial($vlo.getVloVerActStyleArray()[i]);
```

Figure 12.
Verbal active rule

```
Rule: Visual - Reflective
WHEN
$student: StudentProfile(studyStyle == "Reflective") and StudentProfile(learningStyle == "Visual")
$vlo: VLOType(learningMaterial=="Diagram")
THEN
for(int i=0; i<$vlo.getVloVisRefStyleArray().length; i++)
    $vlo.addMaterial($vlo.getVloVisRefStyleArray()[i]);
```

Figure 13.
Visual reflective rule

```
Rule: Verbal- Reflective
WHEN
$student: StudentProfile(studyStyle == "Reflective") and StudentProfile(learningStyle == "Verbal")
$vlo: VLOType(learningMaterial=="Index")
THEN
for(int i=0; i<$vlo.getVloVerRefStyleArray().length; i++)
    $vlo.addMaterial($vlo.getVloVerRefStyleArray()[i]);
```

Figure 14.
Verbal reflective rule

The rules engine obtains the VLO type material according to the student's learning style. The second set of rules was based on the educational context (Higher education) taking into account the student's age. Those results are inputs for the method that adds them with the display preferences of the student. Note that, at this point, the characteristics of the device have not been considered; therefore, many display preferences can appear along with the types of material chosen for the student's VLO.

For the filter device, the device resolution was considered to deduce the size of it.

Finally, the adaptation mechanism was implemented. It joins the high abstraction information obtained as answers by each filter; in this case, the VLO types for the student with his display preferences and the size of the device. The adaptation mechanism takes both answers and validates them. That validation consists on selecting the VLO types that match the size of the device. This is to ensure that the selected VLO is supported by the device. For example, if the device is small, the only display preferences considered are those that can be displayed on small devices.

In previous works (Torres-Ribero *et al.*, 2011; Carrillo-Ramos *et al.*, 2011; Arias-Báez *et al.*, 2011a; Arias-Báez and Carrillo-Ramos, 2012), we have presented different scenarios of the *AES application*. This is a new scenario: When John makes a search in Colombian educational portal, he can get about more than 2,600,000 results (Figure 15). These results are general and are presented to any user who performs the same query.

Using *AES* to obtain the enriched query and included it into the "*Colombia aprende*" portal, the number of results decreases significantly, about 4,130 results (Figure 16).

The results are more specific because the enriched query has more information about the educational context of John (Secondary education), the appropriate Learning

Figure 15.
Results from the query "math"

Figure 16.
Results from the enriched query simulation, table, experiment, self-assessment + document + secondary education + math (typed in Spanish)

resource type (Simulation, Table, Experiment and Self-assessment) and restrictions on his access device (low resolution).

Another study case was the usage of AES with MATEO (Arias-Báez *et al.*, 2013), a framework used to form groups of people based on their characteristics. The flow of information in this scenario can be followed with the following example:

A manager wants to create teams in his/her enterprise, with the roles: secretary, presenter and designer (one person each). Using preloaded profiles for both: the skill requirements and each person's skills within the enterprise, MATEO organizes the possible roles of each participant and also assigns the generic role "Participant" to the people that don't have the required skills for the selected scenario. Once MATEO provides all the possibilities, AES uses an Adaptation Mechanism to enrich the team creation service by selecting the most relevant and accurate role assignations taking into account the context and restrictions used by the manager. AES is used to enrich the "Team Forming" service with the characteristics of the different roles required by the team and the candidate skills, capabilities and competences. Once the enrichment is achieved, MATEO is the responsible of making the match and forming the teams.

The last scenario consists in the application of AES in AYLLU (Carrillo-Ramos *et al.*, 2012), a framework that provides collaboration-oriented services. This platform is agent-based and was applied with the motto: "Learn to cooperate and cooperate to learn". Having acquired the experience from the aforementioned scenarios, the framework AES was used in conjunction with the framework for collaborative work AYLLU in order to provide resources according to the specific needs of a workgroup. Additionally, together with MATEO, AES was used to enrich the creation of workgroups considering all of the participant skills, competencies, characteristics and preferences. AES was also used in AYPUY, a component that manages and search distributed resources to enrich the search queries that obtained the appropriated resources for the different activities involved in the cooperative services.

The general architecture for AYLLU and its components is shown in Figure 17.

6. Conclusions and future work

AES can be used to adapt any environment through the use of filters and adaptation mechanisms according to the application context. It can be integrated with another system that seeks to enrich its services by adapting the information it handles. Additionally, *AES* provides an adaptation logic that can be instantiated and extended in a flexible way; thus, it can be used to adapt services by enriching them considering

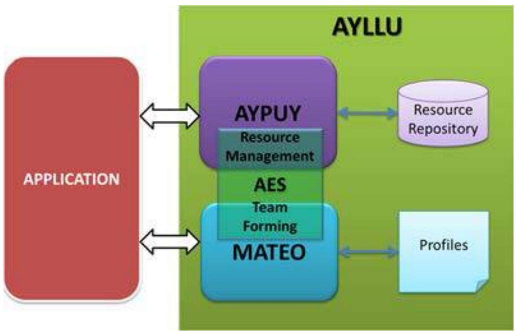


Figure 17. AYLLU general architecture

different stimulus whether they come from different scenarios. This framework considers context aspects such as location, user aspects and access device aspects to adapt the information.

Likewise, AES has tested to be useful due to the possibilities of customization of elements that have a defined profile in a standard format, such as the IEEE LOM (IEEE, 2014) as the ones used in the first study case or custom elements for profile definition as used in the second study case.

Moreover, as seen with the third study case, AES can be adapted to enrich different elements at the same time with two different structures. In this case, two completely different profile definitions such as people profiles and resource profiles can be used with the AES framework at the same time and the adequate filters and adaptation mechanisms will be applied in order to provide an enriched service.

As future work, it is necessary to make tests on other scenarios (e.g. education) and articulate *AES* to different projects whether they use agents or not. For example, this framework can be used in an educational collaboration context to enhance cooperation between students and identify their learning style to present them the learning object that best fits their profile.

References

Andrade, M.T., Arachchi, H.K., Nasir, S., Dogan, S., Uzuner, H., Kondoz, A.M., Delgado, J., Rodriguez, E., Carreras, A., Masterton, T. and Craddock, R. (2007), “Using context to assist the adaptation of protected multimedia content in virtual collaboration applications”, *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, *New York, NY*, *12-15 November*, pp. 233-242.

Arias-Báez, M.P. and Carrillo-Ramos, A. (2012), “Enriched team forming with contextual features”, *2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI*), IEEE, pp. 1-9.

Arias-Báez, M.P., Pavlich-Mariscal, J.A. and Carrillo-Ramos, A. (2013), “Forming adapted teams oriented to collaboration: detailed design and case study”, Conformando Equipos Adaptados Orientados a Colaboración, Diseño Detallado y Caso de Estudio, Vol. 80, pp. 87-95.

Arias-Báez, M.P., Torres-Ribero, L.G., Carrillo-Ramos, A., Pomares, A. and González, E. (2011a), “Adaptation for enriching services taking into account mobile contextual features”, *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia - MoMM ’11, p*. 236, available at: http://dl.acm.org/citation.cfm?doid=2095697.2095744

Arias-Báez, M.P., Torres-Ribero, L.G., Carrillo-Ramos, A., Pomares, A., Garzón, J.P. and González, E. (2011b), “AES: a generic framework for adapting information in multidimensional contexts”, *XXXVII Conferencia Latinoamericana de Informática*, pp. 886-901.

Baldiris, S., Graf, S., Fabregat, R. and Mendez, N.D.D. (2012), “Searching for and positioning of contextualized learning objects”, *International Review of Research in Open and Distance Learning*, Vol. 13 No. 5, pp. 76-101.

Barraza, A.P. and Carrillo-Ramos, A. (2010), “Basic requirements to keep in mind for an ideal agent-based web information retrieval system in ubiquitous environments”, *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, *ACM, New York, NY*, pp. 550-557.

Bouyakoub, F. and Belkhir, A. (2008), "AdaMS: an adaptation multimedia system for heterogeneous environments", *New Technologies, Mobility and Security*, Tangier, pp. 1-5, available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4689069 (accessed 5 September 2014).

Carrillo-Ramos, A., Arias-Báez, M.P., Torres-Ribero, L.G., Pomares, A. and González, E. (2012), "AYLLU: architecture for collaboration in a distributed environment", *Proceedings of the IADIS International Conference Collaborative Technologies*, *Lisbon*.

Carrillo-Ramos, A., Arias-Báez, M.P., Torres-Ribero, L.G., Ruiz, J.P.G. and Gonzalez, E. (2011), "A generic adaptation contextual model based on agents", *2011 6th Colombian Computing Congress, CCC*, Manizales, 4-6 May, pp. 1-6.

Carrillo-Ramos, A., Gensel, J., Villanova-Oliver, M. and Martin, H. (2005), "PUMAS: a framework based on ubiquitous agents for accessing web information systems through mobile devices", *Proceedings of the 2005 ACM Symposium on Applied Computing*, ACM, New York, NY, pp. 1003-1008.

Charrier, R., Bourjot, C. and Lorraine, I. (2009), "Study of self-adaptation mechanisms in a swarm of logistic agents", *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, *San Francisco, CA*.

Chebbine, M.T., Obaid, A., Chebbine, S. and Johnston, R. (2005), "Internet content adaptation system for mobile and heterogeneous environments", *Second IFIP International Conference on Wireless and Optical Communications Networks*, *IEEE*, pp. 346-350.

Cremene, M., Riveill, M. and Miron, C. (2008), "Adaptation platform for autonomic context-aware services", *IEEE International Conference on Automation, Quality and Testing, Robotics, 2008*, *Cluj-Napoca*, *22-25 May*.

Eliassi-Rad, T. and Shavlik, J. (2003), "A system for building intelligent agents that learn to retrieve and extract information", *User Modeling and User-Adapted Interaction*, Vol. 13, available at: http://link.springer.com/article/10.1023/A:1024009718142 (accessed 5 September 2014).

Gómez, S., Zervas, P., Sampson, D.G. and Fabregat, R. (2014), "Context-aware adaptive and personalized mobile learning delivery supported by UoLmP", *Journal of King Saud University - Computer and Information Sciences*, Vol. 26 No. 1, pp. 47-61.

He, J., Gao, T. and Hao, W. (2007), "A flexible content adaptation system using a rule-based approach", *Knowledge and Data Engineering*, Vol. 19 No. 1, pp. 127-140.

IEEE (2014), "IEEE learning technology standards committee (LTSC)", available at: https://ieee-sa.centraldesktop.com/ltsc/ (accessed 27 January 2014).

Imam, I.F. (2004), "Adaptive applications of intelligent agents", *Proceedings ISCC 2004 Ninth International Symposium on Computers and Communications*, Vol. 1, pp. 7-12, available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1358373

Kameas, A., Goumopoulos, C., Hagras, H., Gardner, M., Heinroth, T., Minker, W., Meliones, A., Economou, D., Bellik, Y. and Pruvost, G. (2009), "A pervasive system architecture that supports adaptation using agents and ontologies", *I-SPAN 2009 - The 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, Kaohsiung, pp. 148-153.

Kasiolas, A., Nait-Abdesselam, F. and Makrakis, D. (1999), "Cooperative adaptation to quality of service using distributed agents", *Proceedings of the 1999 ICPP Workshops on Collaboration and Mobile Computing (CMC'99)*, Industrial Applications on Network Computing (INDAP), Multimedia Network Systems (MMNS), Security (IWSEC), Parall, pp. 502-507.

Lee, P.U. and Field, M. (2005), "Understanding human-human collaboration to guide human-computer interaction design in air traffic control key factors that drive the collaborative behavior in ATC responsibilities and control".

Ma, H. (2008), "Adaptive agents for sequential Vickrey auctions with homogeneous goods", *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, *Vienna*, *10-12 December*, pp. 482-487.

Marin-Díaz, D., Rico-Zuluaga, A., Carillo-Ramos, A. and Arias-Báez, M.P. (2010), "An adaptation model applied to commercial establishments services", *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia - MoMM '10*, p. 397, available at: http://portal.acm.org/citation.cfm?doid=1971519.1971587

Mönch, L. and Zimmermann, J. (2008), "An ontology to support adaptive agents for complex manufacturing systems", *32nd Annual IEEE International on Computer Software and Applications, COMPSAC '08*, Turku, 28 July-1 August, pp531-536.

Red Hat, I. (2006), "Drools", available at: www.drools.org/ (accessed 1 January 2014).

The World Wide Web Consortium (W3C) (2001), "Synchronized multimedia integration language (SMIL 2.0)", available at: www.w3.org/TR/2001/REC-smil20-20010807/ (accessed 1 January 2014).

Torres-Ribero, L.G., Garzon, J.P., Arias-Baez, M.P., Carrillo-Ramos, A. and Gonzalez, E. (2011), "Agents for enriched services (AES): a generic agent - based adaptation framework", *Proceedings of the 2011 International Conference on Collaboration Technologies and Systems, CTS 2011*, *Philadelphia, PA*, *23-27 May*, pp. 492-499.

Zemirli, N., Lechani Tamine, L. and Boughanem, M. (2005), "Accès personnalisé à l'information: proposition d'un profil utilisateur multidimensionnel", *Proceedings of the 7th International Symposium on Programming and Systems (ISPS'2005)*, Algiers, 9-11 May.

Zhu, H., Hou, M. and Zhou, M.C. (2010), "Establishing the foundation of adaptive collaboration", *2010 International Symposium on Collaborative Technologies and Systems (CTS)*, Chicago, IL, 17-21 May, pp. 546-554.

Further reading

González, E., Bustacara, C. and Ávila, J. (2003), "BESA: arquitectura para Construcción de Sis-temas MultiAgentes", *Conferencia Latinoamericana de Estudios en Informática*.

Corresponding author

Luis Guillermo Torres-Ribero can be contacted at: luis-torres@javeriana.edu.co