Luis Guillermo Torres-Ribero
Address: Cra. 7 No. 40-62, Bogotá - Colombia
Phone: (57-1)3208320 ext 5382
e-mail: luis-torres@javeriana.edu.co

Juan Pablo Garzón
Address: Cra. 7 No. 40-62, Bogotá - Colombia
Phone: (57-1)3208320 ext 5309
e-mail: jpgarzon@javeriana.edu.co

Maria Paula Arias-Báez
Address: Cra. 7 No. 40-62, Bogotá – Colombia
Phone: (57-1)3208320 ext 5382
e-mail: arias.m@javeriana.edu.co

Angela Carrillo-Ramos
Address: Cra. 7 No. 40-62, Bogotá - Colombia
Phone: (57-1)3208320 ext 5392
e-mail: angela.carrillo@javeriana.edu.co

Enrique González
Address: Cra. 7 No. 40-62, Bogotá - Colombia
Phone: (57-1)3208320 ext 5361
e-mail: egonzal@javeriana.edu.co

Agents for Enriched Services (AES): A Generic Agent - Based Adaptation Framework

Luis Guillermo Torres-Ribero, Juan Pablo Garzón, Maria Paula Arias-Báez, Angela Carrillo-Ramos, Enrique González {luis-torres, jpgarzon, arias.m, angela.carrillo, egonzal}@javeriana.edu.co

***Abstract*—This paper presents AES (acronym of Agents for Enriched Services) framework; a generic framework that can be used to enrich services and adapting an original service request to what the context information, user profile and other multiple sources of information can bring in order to improve the end-user experience in using the framework. The framework is presented along with its architecture and the way it was implemented using agents, an example showing how this framework may improve collaboration and a study case are presented in order to validate the framework's functionality.**

***Index Terms*—Adaptation Methods & Techniques, Agent Adaptation, Support for Adaptive Collaboration, Adaptive Collaborative Learning**

Introduction

A lot of people from around the world use the web as a new gathering space, gigantic social networks and massive multiplayer on-line games prove that Internet can be seen as a global community without boundaries. Learning has also been enhanced by the World Wide Web by allowing multiple students from all over the world to access different virtual learning objects and collaborate to obtain new knowledge. Enterprises have also found benefit in the web allowing communication between workers to improve their collaboration; however, as established by Zhu [14] adaptability is a key element in a collaboration environment. As seen in [1] an adaptive collaboration framework for a virtual classroom application, adaptive collaboration greatly improves learning compared to a scenario where adaptation is not considered.

Cambridge Dictionary defines adaptation as “the process of changing to suit different conditions”, therefore, adaptation is a process that considers a lot of different variables in order to generate change in a particular entity. In software, adaptation requires awareness of external conditions and the ability to change a behavior proactively in order to adjust to changing conditions; Software agents have these features: they are proactive and capable of changing their behavior dynamically, making them an efficient tool when applying adaptation to a software environment. Even though there are Adaptation Mechanisms that don't use software agents, e.g. Demetriadis [6], who uses collaborative scripting for adaptation, this paper will mainly focus on adaptation using agents.

There are frameworks that use agents to enhance the adaptation process such as the framework for Complex Manufacturing Systems in [13] or the adaptive agents for sequential auctions like the one described in [12], which use adaptation with agents on a specific problem. There are also frameworks as ATRACO [10]that use a more general approach centered on ontologies. All of the works mentioned above use adaptation technologies either in specific applications or using specific technologies, this allows those frameworks to be efficient in their own specific area, however it is important to note that none of the previously mentioned frameworks are generic, that is, none allow the inclusion of new scenarios or new technologies to their adaptation process.

This paper presents a generic framework called AES, Agents for Enriched Services, that allows the system to enrich services and provides to the software designer engineer with the flexibility of implementing technologies required for the specific user needs, making it an ideal framework to use as starting point to adapt services by enriching them considering different stimulus whether they come from the environment, devices or user preferences. In order to introduce the AES framework, this paper is organized as follows. In Section I some Agent-based adaptation systems are discussed, Section II presents the AES architecture and its components, Section III describes the agent model used to implement the given architecture, SectionIV describes the methodology to enrich a service using the AES framework approach, Section V presents a case study that validates the framework and finally Section VI presents the final conclusions and future work.

I. Agent-based Adaptation Systems

Adaptation using agents is a widely discussed matter with different approaches. One of them is using ontologies to represent knowledge and use them as a tool for changing a behavior towards the environment; Kameas shows, how through ontology alignment, a process in which equivalences between entities are shown, to build knowledge [10]. This approach uses both the knowledge in each agent and a collective knowledge to enhance adaptation process. However, this approach is based only on the use of ontologies, discarding other possible knowledge representation options, and it is supported on a SOA architecture limiting other possible architectural structures.

A different approach is the use of a large number of cooperative distributed agents [11] that make the adaptation process by sharing acquired knowledge, enabling agents to recognize user's needs and communicating them to other agents in order to improve the quality of the services the user needs. This approach implies a large quantity of agents interacting, which must be organized in order to achieve the goal efficiently. This is the objective of the work developed by Charrier [4], who proposes to organize, using a quadratic or logistic map, the agent's behavior adapting their interactions and the information stream according to their environment so they can effectively share knowledge. Other authors, such as Eliassi-Rad [7], turn user preferences into instructions for agents and compiling them into neural networks allowing the agents to learn. Other frameworks such as PUMAS [3] use adaptation through agents in a mobile environment; in this framework adaptation is based on user preferences, habits, technical constraints of his/her mobile device, and others aspects, in order to adapt the results of the query made; adaptation has also been applied in fields such as Information Retrieval as in AndreaBarraza[2], or in adapting preferences in a commercial establishment as in Diaz[5].

We can observe that there are different types of adaptation and different ways to achieve adaptation using agents as well, so it can be helpful to separate these types of adaptation in different categories. We can make a classification by analyzing two different perspectives: a behavioral perspective and a resource perspective.

The behavioral perspective focuses on the way agents behave; agent's behavior determines the way in which agents interact between them and with their environment. If their behavior has changes, it can affect the information they transmit and the way they in which react when an event occurs. Imanm [9] proposes a taxonomy following this approach:

- Internal Adaptation: In this category agents adapt internally to changes, without modifying their behavior towards the environment or their outcome. An example of this category is an agent which improves an inner algorithm making it faster or more efficient; however the response that it produces as an outcome and the services it provides remain unchanged.
- External Adaptation: In this scenario the agent adapts by changing its behavior towards its environment, however internal functionality changes are not made. An example of this category is an irrigation system that adapts to different climate conditions and waters different places according to climate changes, therefore its external behavior changes but its inner irrigation algorithm stays the same.
- Complete Adaptation: It refers to the case where both, the internal and external behavior of the agents change. An example of this is a robot that has a specific plan according to external conditions, it changes it internally thus modifying its external behavior.

Resources used by an adaptation framework can also be taken into consideration when adapting information; these resources often determine the quality of the outcome of an adaptation process. A resource perspective has been analyzed in [10]; this approach defines different kinds of adaptation categories based on the elements that are subject to adaptation:

- Artifact Adaptation: Artifacts that adapt according to the context, user preferences and the artifact's characteristics related to its environment.
- User Behavior Adaptation: Learn and adapt to the environment and the user's preferences.
- Interaction Adaptation: Provide transparency and ease of use to the end user.
- Network Adaptation: Provide efficient resources discovery and their management.
- Sphere Adaptation: Manages behavioral and structural changes of the environment in which different tasks are going to be made.

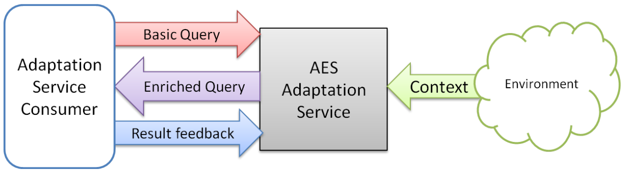

Figure 1. General AES Framework operation

Although these types of adaptation come from the definition of two different frameworks, it is possible to interpret them as the requirements that a generic adaptation framework should meet, in order to achieve a better service enrichment. Whether in the behavioral perspective or in the resource perspective, a generic framework should be able to provide enough flexibility to the user so it can take advantage of all of these types of adaptation and adjust them to its specific needs. The AES framework aims to provide such flexibility, taking advantage of agent's capabilities and provide technology-independent adaptation, so the user can have adaptation services made using its own logic, resources and particular point of view of its needs.

II. AES Architecture

The AES Architecture is based on a multiagent system that control resources and execute methods when needed. Those resources and methods can be implemented by the user of the framework, so it is the user who decides what type of resources and what type of adaptation needs in a particular scenario.

The architecture of AES will be explained while following an example of how this framework can be used in order to enhance collaboration. In the example, the service to be implemented aims to solve the problem that arises when a teacher of a certain course wants to create a group of students that will collaborate; the required working group must have well-balanced skills between its members and each member must have a role according to its abilities.

A. General Approach

In this paper, the concept of adaptation is defined as the ability to enrich an initial provided request by taking into account information from different sources not explicitly provided in the request. The source of the additional information can include user profile, device information and other types of relative data that may generate changes in a particular service query and will make it suitable for the context in which the request is made. The adaptation process greatly changes a request that doesn't consider information other than the information contained in the request itself.

The black box schema of the AES adaptation component is depicted in figure (see Figure 1). The AES Adaptation Service's goal is to provide a consumer (human or virtual)

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
```

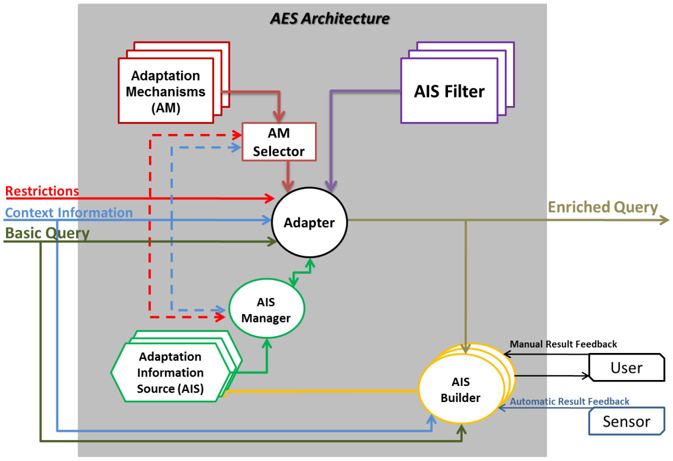

Figure 2. AES Architecture Model

with the functionality to enrich a request by considering additional information not provided in the initial request, enhancing the initial request making it more precise according to the information available at the moment. This additional information takes into account features such as user's context and profile in order to adapt information to user needs and characteristics.

The AES component works with different inputs to produce the desired output: the Basic Query input is the initial request where the user specifies his/her requirements and additionally particular restrictions that must be taken into consideration in the adaptation process; the Context input includes environmental information, access device information, basic user information, among other information that will be used to enrich the Basic Query. All this input information is analyzed and joined to the Basic Query in order to produce an output that will consider all of the context elements around the request to produce an Enriched Query output. Another input of the adaptation component is the Result Feedback information that allows updating the knowledge that the system uses for its adaptation purposes; for instance the user profile and preferences, among other context oriented information may change according to the information contained in the Result feedback. The Result Feedback input can come from a lot of sources that include, but are not limited to, user feedback.

The following scenario shows which are the inputs and outputs of the AES: in the group creation example the Adaptation Service Consumer would be whether a teacher (human) or a platform aimed to enhance collaboration among a group of computer science students. The Basic Query input to the framework will be basic characteristics of the group (Basic Query example: "well-balanced group"), the Context Information input will contain course information, required abilities for the particular group and goal of the group (*i.e.* what is the purpose of forming the group). The teacher/platform will then obtain as Enriched Query output the group formation query that takes into account the inputted characteristics and the environment information (Enriched Query example: "group that includes two students with high programming skills, one student with medium software design skills and one student with high management skills ").

That group will be adapted to the course goal of making a small software project that must have all of the design and development process while having a team leader. The Enriched Query output for group formation will consider the best scenario given the Basic Query and the Context Information in which the group creation request was triggered. The teacher may give a Result Feedback to the framework telling it whether the result was adequate or not. If the result was not adequate the framework will correct, based on the Result Feedback, the inner mechanisms to provide a better Enriched Query output in the future.

The process that made the adapting of the initial input and context will be described along with the different components of the architecture, shown in Figure 2.

The AES Architecture 2 shows in more detail the inputs and outputs of the framework, the Context input in figure 1 remains as the Context Information input in figure 2; the Basic Query input in Figure 1 is shown as separated inputs in Figure 2: Basic Query input and Restrictions Input, The Feedback Input in Figure 1 is shown in Figure 2 as Manual Result Feedback and Automatic Result Feedback. The Architecture of the AES framework shows the different components that interact in the generation of an Enriched service Query taking into account the different inputs and components of the architecture. Note that some components are represented as ovals, other components as rectangles and one component has a hexagon; the hexagonal-shaped component are the Information Sources that will be used for adapting a query, these Adaptation Information Sources must contain information before a query arrives; circular-shaped components are components that use, and interact with, resources represented by a rectangular-shaped notation; rectangular-shaped components are resources that both store information and/or contain methods that will be executed by the circular-shaped components when needed.

Resources are main components of the AES architecture; they contain information or procedures that are used to enrich a query that entered the framework. Information stored in the different resources will be used by the adaptation service to take additional information and join it with the original query. Resources also store procedures or methods that will enhance the adaptation process; these methods are flexible and can be implemented by the designer engineer with the algorithms that best fit the end user's requirements.

In the AES architecture the basic resources are the AIS (Adaptation Information Sources) these resources store information on different elements that are relevant to the enrichment of the initial query, *e.g.* a user profile or access device information. The AIS must be filled with information and stored in order to make adequate query enrichment. Once the AIS contain information the adaptation service is ready to start.

The Adaptation Service receives an input, *i.e.* a Basic Query and Particular Restrictions; the Basic Query input contains the basic data concerning the request of the Adaptation Service Consumer, this is the query that will be enriched. Another input is the Particular Restrictions input; this input acts as a constraint, telling the Adaptation Service that some information in the basic query must remain independently of what the Adaptation Service generates as an ideal output.

Once the Basic Query and the Particular Restrictions enter the framework the context information is obtained and the Adaptation Information sources are requested to the AIS Manager in order to obtain additional information based on the context in which the query was made. After the profiles have been obtained from the AIS manager, another resource aids in the Adaptation Service process, the Adaptation Mechanism Selector.

The Adaptation Mechanism Selector selects, among every Adaptation Mechanism available, the Adaptation Mechanism that best fits the adaptation needs, the criteria used to determine the "best fit" of the Adaptation Mechanism is contained in a procedure or method in the Adaptation Mechanism Selector; this procedure is flexible and the designer engineer is able to provide an appropriate algorithm based on the consumer needs.

All of the Adaptation Mechanisms have associated AIS Filters; the AIS Filters contain methods that take AIS information as input and produces high abstraction information as output, the selected Adaptation Mechanism will use all of the associated AIS Filters in the obtained AIS information and will generate a set of high abstraction information (corresponding to the output of each associated filter), once all of the AIS Filter's methods have been executed, the set of resulting high abstraction information will be used by the Adaptation Mechanism in a procedure that enriches the query. This enrichment procedure will take into account the Basic Query input, the Particular Restrictions and the High Abstraction Information obtained. The algorithm to enrich a query is also flexible and can be implemented by the designer engineer. Once the enrichment method is executed, the output is the enriched query that will be returned to the consumer; optionally a feedback may be received by the service, and that feedback will update, using the AIS Information Builder, the information stored in the AIS if required in order to provide a better Adaptation Service in the future. The basic algorithm that the Adaptation Service uses to enrich a query is represented in Algorithm 1.

Following the group formation example the AIS would be all of the students' profiles, including their skill levels in the course topics, the access device information and the course topics and goals. A series of Adaptation Mechanisms would include the Adaptation Mechanism for group forming; this Adaptation Mechanism would have the appropriate AIS_Filters to form a well-balanced group, the AIS_Filters would analyze each student profile and will generate High Abstraction Information regarding each student skill in certain subjects.

The Basic Query "Form a well-balanced group" and the Particular Restriction "not all students in the group" would be example inputs; the Context would be the course information and the goal of the group to be formed. Once this information enters the Adaptation Service, the Adaptation Mechanism for forming a well-balanced group that don't include all of the students is selected and all of the AIS_Filters will filter the basic information and produce as output the skill level for each student and additional information that may be relevant to the group forming.

Finally the High Abstraction Information is enriched by the selected Adaptation Mechanism with the Basic Query and the Particular Restriction as inputs. The Resulting Enriched Query would be: "A group that includes two students with high programming skills, one student with medium software design skills and one student with high management skills".

AIS, AIS Filters and Adaptation Mechanisms functionality are described in the following sections in a general way.

B. Adaptation Information Sources and Learning

Adaptation Information Sources (AIS) are the components in where all of the information relevant to the application context is stored; these AIS will be the main providers of additional information to be taken into account while in the adaptation process. AIS represent the characteristics of the elements that interact with the adaptation service; due to the uncertainty of what information could be used in a particular context, it is highly recommended that the AIS stores as much relevant information available. The AIS can store any type of information, whether it is a user profile, characteristics of a device, location characteristics, etc.; this will allow the service to make a generic approach of the kind of information can be contained in the AIS.

In order to make the Adaptation Service work adequately the information stored in the AIS must be precise and up-to-date, even though the AIS must be filled with information before the Adaptation Service is used, the initial stored information may have to change in time due to changes in a profile or new information that must be considered and wasn't considered when the AIS were created. To enable information changes in the AIS the Adaptation Service also provides a learning functionality that will be handled by the AIS Builder. The AES Builder will take a feedback if there is one and will analyze if the feedback contains information regarding the final output of the Adaptation Service. If the feedback is negative, *i.e.* an Enriched Query didn't accomplish what the end user expected, the AIS Builder will change the AIS information that may have affected the enrichment of the query and the proper output generation.

A feedback can be generated automatically of manually, if the feedback is done manually it requires the end user to provide a feedback whether by making a comment or by filling out a form with questions regarding the satisfaction of the user with the generated result; once the feedback is made the AIS Builder will update, if required, the information in the AIS that needs the new or changed information. An automatic feedback would be generated if the user of the Adaptation Service is an external entity such as another software and that entity raises an automatic response, *e.g.* number of used elements that were shown due to an Enriched Query, and if the result is not satisfactory the information will be used to update information stored in one or many AIS in order to improve the adaptation process.

Once the learning process has taken place, future requests to the Adaptation Service will take into account the new information and the AIS Manager will provide the AIS with the updated information.

C. Filters and Adaptation Mechanisms

Other framework elements are Filters and Adaptation Mechanisms, these two work in conjunction in order to produce the output, Filters are associated with Profiles and obtain high abstraction information from them; Continuing with the example of conforming a group of students a Filter would be a set of methods that according to a student abilities designates the most adequate role to play in the group.

Adaptation Mechanisms are a set of filters and a method of enrichment, the Adaptation mechanism first obtains the high abstraction information from each filter and finally, using its enrichment method, combines that information with the original input to produce an enriched output; in the group conforming example an Adaptation Mechanism has filters regarding the age of the students and their learning style, another Adaptation mechanism has filters regarding students' abilities and expertise; hence the latter is the selected Adaptation Mechanism to enrich the group conforming service due to its characteristics.

The Adapter, described in the next section, is the component that selects the most adequate Adaptation Mechanism and applies the enrichment method to produce the enriched service.

D. Adapter

The adapter is the main component of the AES framework; it receives the initial input and based on the information it receives, that is the service query, the context and particular restrictions, selects the most adequate Adaptation Mechanism to enrich the service, after the most adequate Adaptation Mechanism is selected, the Adapter takes each filter associated with the Adaptation Mechanism and requests the filter's associated profiles to the Profile Manager, after the profile manager delivers those profiles, the Adapter executes the filtering function in each filter, hence obtaining high abstraction information. After all of the filters have generated high abstraction information the Adaptation Mechanism's method is executed and the service query is enriched and outputted.

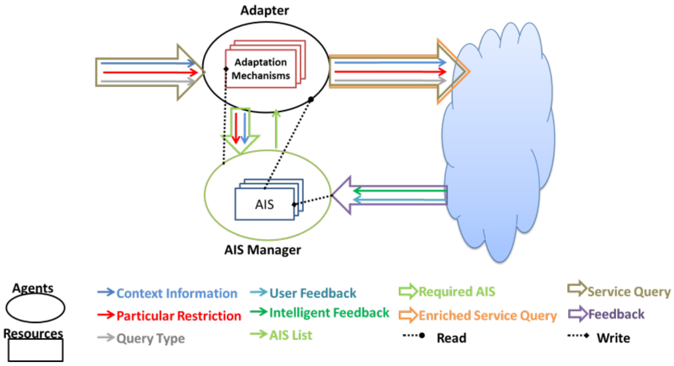

Figure 3. Agent Model implementation of AES Architecture

Additionally there can be a feedback that is attended by the Adaptation Information Builder and, if required, updates a profile in order to improve future uses of the AES Adaptation Service.

III. Agent Model of AES

Making an analysis of the main features in software agents and the requirements of an adaptation service, this technology has been selected as the most adequate approach for making a generic framework that provides such a service. The process for adapting information requires the analysis of large amount of information in a rather short period of time and additionally the environment may change rapidly and those changes must be taken into account as they happen; software agents provide an adequate solution for issues present in an adaptation process, due to software agents ability to be scalable and to distribute work, they are able to quickly analyze and interpret large amounts of information; another characteristic of software agents is their ability to react to environmental changes hence making them able to sense when a change has occurred and changing the way the information is handled if the changes in the environment requires it.

The main reason for this model to be considered multiagent is that, due to the increasing use of software agents in many software frameworks or products, this model is intended to interact with other agents that may be working within the Adaptation Service Consumer (see Figure 1) making a seamless addition of the Adaptation Service to a consumer.

In the following sections the agent model for the AES architecture is shown, followed by the description of each category of components modeled.

In Figure 2 the different components in the AES architecture are shown, in Figure 3, an agents representation of the architecture is shown. Note that Interactions are similar to interactions previously shown in Figure 2, to represent agents and their related resources, those resources represented as rectangle-shaped components are shown in its corresponding agent, represented by a circle-shaped components. Filters are considered as part of Adaptation Mechanism so they are not shown in Figure 3. Another difference is the absence of the Adaptation Information Builder, the reason for this change is that the Profile Manager Agent will include responsibilities of both Profile Manager and Adaptation Information Builder.

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 a n Active component indicates the Inactive component when to execute the method and also provides the required parameters that method will use.

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 3. There are two main Active Components in the framework: The Adapter Agent and the Profile Manager Agent. 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 are the Adaptation Mechanisms and Filters used by the framework; its behavior includes reacting to an incoming query, sending a profile request to the Profile Manager, receiving profiles sent by the Profile Manager, executing each filter's filtering method and finally execute the enrichment method to generate the output.

The Profile Manager's state is composed by recent profiles and the context plus restriction information (called new Context). Its behavior is to react to the Adapter's request of profiles and looking for them in a persistence component (if any) or in its state, and updating profiles upon feedback.

Feedback sources are shown in Figure 3 as a cloud to represent that feedback's source can be a lot of different components, whether it's a user or another agent or any other possible source.

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. Inactive components of the AES framework are: Filters, Profiles and Adaptation Mechanisms.

Profiles contain information regarding the elements that interact with the enrichment process, storing information relevant with services offered by the framework.

Filters contain information regarding their associated Profiles and the filtering method, which takes associated profiles and generates high abstraction information.

Adaptation Mechanisms contain associated Filters and a method to enrich the query based on the high abstraction information, the initial service query , the context and the particular restrictions.

IV. Enriched Service Development

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

A. Analysis

The first phase must be the analysis phase, in which the user must first identify services he/she would like to provide, after correctly identifying those services, the user must identify the different enrichment opportunities in a service, for example, in the 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, so the complete service with its enrichment opportunities is fully characterized.

Once these steps are made the user must identify the different profiles it needs, different characteristics and information that can be stored and that will help enriching the services, the filters it needs, *e.g.* 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 to produce an enriched output.

B. Design and Deploy

Once the elements have been identified they must be mapped into all the inactive components of the framework. The basic element are Profiles, the AES Framework provides flexibility so the user can decide the persistence method that will be used to store those Profiles, whether is a database, plain text files, *etc.*

After the Profiles have been designed, created and stored the next step is to create Filters and associate them with relevant profiles and implementing the Filter's 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 net 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 different Filter combinations are set in order to achieve the enrichment of the implemented service. 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 service. 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 different restrictions in order to validate the framework's functionality.

V. Case Study

In order to validate the framework, and implementation was made using BESA (Behavior-oriented Event-driven Social-based Agent framework) [8] a generic agent framework developed in the Pontificia Universidad Javeriana and in the Java programming language. In the test case, aside from implementing AIS, AIS Filters (with each corresponding filter function) and Adaptation Mechanisms (with each corresponding enrichment function) it was determined some features that the user must have control over. One of these features is the way the restrictions are applied to the context, a function that must be implemented by the user; later on the process the Profile Manager component selects AIS according to what the adapter sends; this selection process is also implemented by the user so it can determine such criteria as priorities, number of possible results, among others. Finally the update functionality of the Profile Manager was implemented in order to change a Profile according to feedback provided to the framework.

The test was made using information based on an access device, a specific location characteristics and a basic user profile, as a result a simple query inputted was enriched and a new, more precise, query was generated by the application.

This case study proved the flexibility of the framework due to its use of Object inheritance that allows the user to extend new classes from basic classes overriding the key methods and attributes that the framework uses to enrich a service. These AIS Filters, AIS and Adaptation Mechanisms can modify the basic attributes and operations of their parent classes in order to use the most suitable methodologies or technologies available in order to solve a particular adaptation need of an AES framework user.

VI. Discussion and Future Work

This paper defines adaptation as the enrichment of an element, taking into account external context, user profile, and access device characteristics among others to take basic information and enhance it merging them with all the additional information available.

The AES Architecture, a generic agent-Based adaptation framework used to enrich different services is described. The way agents collaborate and proactively react to different events in their environment makes them the most suitable solution for adapting information using the different resources that are needed by the AES framework.

An example of how collaboration can be enhanced by using this framework is presented in an example while describing the AES framework components. Finally a Study Case is presented in order to validate the frameworks operation and the results of this study case are described.

The future work includes using this framework in an agent based CSCW framework (Computer Supported Cooperative Work) focused on educational collaboration called AYLLU as a method to enrich different services that enhance cooperation between students and identify their learning style in order to present them the learning object that best fits their profile.

Acknowledgments

This paper is developed within the project “AYLLU: Plataforma de Cooperación mediada por Agentes aplicada en un Contexto de E-learning Colaborativo” with COLCIENCIAS registry No: 1203-489-25592 and contract CT 599-2009. Additionally, it is developed in the frame of the project “ASMA: Modelo de Adaptación Contextual en Sistemas Multiagentes”. Both projects are supported by the Pontificia Universidad Javeriana (Bogotá, Colombia).

References

[1] M.T. Andrade, H. Kodikara Arachchi, S. Nasir, S. Dogan, H. Uzuner, A.M. Kondoz, J. Delgado, E. Rodriguez, A. Carreras, T. Masterton, and R. Craddock. Using context to assist the adaptation of protected multimedia content in virtual collaboration applications. In *Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on*, pages 233 –242, 2007.

[2] Angela Carrillo-Ramos Andrea Barraza. Basic requirements to keep in mind for an ideal agent-based web information retrieval system in ubiquitous environments. In *12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010)*, pages 548–555, 2010.

[3] Angela Carrillo-Ramos, Jérôme Gensel, Marlène Villanova-Oliver, and Hervé Martin. Pumas: a framework based on ubiquitous agents for accessing web information systems through mobile devices. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1003–1008, New York, NY, USA, 2005. ACM.

[4] R. Charrier, C. Bourjot, and F. Charpillet. Study of self-adaptation mechanisms in a swarm of logistic agents. In *Self-Adaptive and Self-Organizing Systems, 2009. SASO '09. Third IEEE International Conference on*, pages 82 –91, 2009.

[5] David Marín Díaz, Alejandro Rico Zuluaga, Angela Carrillo-Ramos, and María Paula Arias Baez. An adaptation model applied to commercial establishments services. In *8th International Conference on Advances in Mobile Computing and Multimedia (MoMM2010).*, pages 405–408, 2010.

[6] S. Demetriadis and A. Karakostas. Adaptive collaboration scripting: A conceptual framework and a design case study. In *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, pages 487 –492, 2008.

[7] Tina Eliassi-Rad and Jude Shavlik. A system for building intelligent agents that learn to retrieve and extract information. *User Modeling and User-Adapted Interaction*, 13:35–88, February 2003.

[8] Enrique Gonzalez, Jamir Avila, and Cesar Bustacara. Besa: Behavior-oriented, event-driven, social-based agent framework. In *PDPTA*, pages 1033–1039, 2003.

[9] I.F. Imam. Adaptive applications of intelligent agents. In *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*, volume 1, pages 7 – 12 Vol.1, 2004.

[10] A. Kameas, C. Goumopoulos, H. Hagras, M. Gardner, T. Heinroth, W. Minker, A. Meliones, D. Economou, Y. Bellik, and G. Pruvost. A pervasive system architecture that supports adaptation using agents and ontologies. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 148 –153, 2009.

[11] A. Kasiolas, F. Nait-Abdesselam, and D. Makrakis. Cooperative adaptation to quality of service using distributed agents. In *Parallel Processing, 1999. Proceedings. 1999 International Workshops on*, pages 502 –507, 1999.

[12] Huiye Ma. Adaptive agents for sequential vickrey auctions with homogeneous goods. In *Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation*, pages 482–487, Washington, DC, USA, 2008. IEEE Computer Society.

[13] L. Monch and J. Zimmermann. An ontology to support adaptive agents for complex manufacturing systems. In *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pages 531 –536, 282008-aug.1 2008.

[14] Haibin Zhu, Ming Hou, and MengChu Zhou. Establishing the foundation of adaptive collaboration. In *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pages 546 –554, May 2010.