

Learning analytics tasks as services in smart classrooms

Jose Aguilar^{1,2} · Manuel Sánchez³ · Jorge Cordero⁴ · Priscila Valdiviezo-Díaz⁴ · Luis Barba-Guamán⁴ · Luis Chamba-Eras⁵

Published online: 8 February 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract A smart classroom integrates the different components in a traditional classroom, by using different technologies as artificial intelligence, ubiquitous, and cloud paradigms, among others, in order to improve the learning process. On the other hand, the learning analytics tasks are a set of tools that can be used to collect and analyze the data accumulated in a smart classroom. In this paper, we propose the definition of the learning analytics tasks as services, which can be invoked by the components of a smart classroom. We describe how to combine the cloud and multi-agent paradigms in a smart classroom, in order to

provide academic services to the intelligent and non-intelligent agents in the smart classroom, to adapt and respond to the teaching and learning requirements of students. Additionally, we define a set of learning analytics tasks as services, which defines a knowledge feedback loop for the smart classroom, in order to improve the learning process in it, and we explain how they can be invoked and consumed by the agents in a smart classroom.

Keywords Learning analytics as service · Smart classroom · Cloud computing · Ambient intelligences

✉ Luis Chamba-Eras
lachamba@unl.edu.ec

Jose Aguilar
aguilar@ula.ve

Manuel Sánchez
mbsanchez@unet.edu.ve

Jorge Cordero
jmcordero@utpl.edu.ec

Priscila Valdiviezo-Díaz
pmvaldiviezo@utpl.edu.ec

Luis Barba-Guamán
lrbarba@utpl.edu.ec

¹ CEMISID, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela

² Universidad Técnica Particular de Loja, Loja, Ecuador

³ Universidad Nacional Experimental del Táchira, San Cristóbal, Venezuela

⁴ Department of Computer Science and Electronic, Universidad Técnica Particular de Loja, San Cayetano Alto, Loja, Ecuador

⁵ Carrera de Ingeniería en Sistemas, Universidad Nacional de Loja, La Argelia, Loja, Ecuador

1 Introduction

The study of Ambient Intelligence (AmI) in education is a current hot topic [1]. The new advances in information technology, especially in ubiquitous computing that allows us to interact with all devices with computing capabilities as a whole, and in cloud computing that allows us to maintain and use educational services (learning resources, learning analytics tasks, etc.) in a distributed manner, are some of the main reasons for this interest. Particularly, in education the development of smart classrooms is one of the emerging challengers in the area of AmI. A smart classroom integrates in an unobtrusive manner the sensor technology, communication technology, and artificial intelligence, among others, into the classroom, in order to improve the learning process.

In this paper, we analyze one of the services that can be provided from the cloud in a smart classroom, the learning analytics (LA). In general, LA can be defined as the collection, analysis, and use of the data accumulated about learning processes [2]. LA uses statistical techniques, machine learning approaches, and data visualization

techniques, among others, to provide knowledge which will help to optimize student performance, highlight students' problems, improve pedagogical strategies, tune up educational platforms, etc. In this way, LA can improve the teaching and learning process using educational data, because it provides knowledge about what takes place over the trajectory of student learning.

In general, there are a lot of propositions of LA tasks for different types of problems in educational environments. Some of them are [2, 3]: To predict the student performance, to discover the different learning patterns of each student, to analyze collaboration and interaction in a learning environment, to analyze the evaluation strategies during the learning process, to analyze student motivation in the context of a learning process, etc.

In a smart classroom, the LA tasks can have two goals [4]: to analyze the learning process, and to analyze the student behavior. They must be combined for reaching the global goal: improve the learning experiences. The idea is to exploit the knowledge in a smart classroom, in order to follow the dynamics of teaching and learning of students. With LA, the smart classroom can supervise student performance, provide learning resources according to the student's learning style, link students studying similar topics, etc. The LA goal in a smart classroom is knowledge extraction to better understand the students and the manner how they learn. The aim is to generate knowledge about the students and their learning contexts, for the purpose of understanding and optimizing the learning process and the teaching environments.

Some recent works in LA, which can eventually be used in a smart classroom, are presented in the following. Purdue University has developed a prediction model, which extracts data from the Course Management System (CMS) and predicts the students with academic risk [2]. By using logistic regression mechanisms, the model predicts the student success in a given course. In [3] a descriptive model is defined to understand the different factors with incidence in the student dropout. Ref. [5] explores ways to define and configure student workgroups. The authors propose a method for configuration of groups of students, based on the analysis of their activities. In [6] a methodology is proposed for defining metrics that enables the calculation of the effectiveness of students when interacting with educational resources and activities in MOOCs. In [7] the problem of how to define a predictive model of students' performance, which is both practical and understandable for users, is addressed using the business intelligence paradigm, in order to exploit the data from the learning process. In [1] the overall student performance is predicted based on decision tree approaches. They used internal assessments in the VLE and concluded that classification techniques can be applied to educational data for

predicting the student's outcome. In [8] web usage mining was used to analyze the learning process and the activity of groups of students, in order to propose motivation strategies on LMS environments. In order to make the student modeling process much easier, [9] applies different classification techniques on the student data to predict their performance. In their research, they implement a tool, to discover how the students are behaving and progressing in the courses. In [10], they used a genetic algorithm and a decision tree-based classification, on student data to understand their different learning capacities. Finally, in [11] a Middleware for Smart Educational Environments is presented, based on the paradigm of cloud computing, which provides services of learning analytics in the cloud.

The previous works give an idea of a variety of research in the domain of LA. They show how the knowledge generated can be used to solve educational problems, support educational decision making, but at the same time, they pose new research questions. One of them is the goal of this work, i.e., how can LA be used to improve the learning process in a smart classroom? The LA tasks must provide learning indicators that genuinely give meaning about the learning process.

In previous work, we have developed a middleware based on a multi-agent system, called AmICL, to support smart classroom environments [12, 13]. This middleware proposes different levels, one for the management of the multi-agents community, another to manage the access to services, applications, etc., and the last one to characterize the different components (of software and hardware) of a smart classroom. The first level is composed of a multi-agent community to support the execution of multi-agents applications. This level follows the FIPA standard (see [12]). The second level has a component, the Services Management Layer (SML), which has the responsibility of finding, searching, etc., services required by the applications. The last level represents the different devices and software components in a smart classroom. Additionally, it has two agents: the first is a profile agent representing each student (SPA), and the second is the tutor agent (TA) to represent the professor. In [12, 13] AmICL is presented in detail, and in [1] a smart classroom is modeled as a multi-agent system, supported by AmICL. Some of the components of a smart classroom modeled as agents are the Student's Profile Agent (SPA), the Tutor Agent (TA), the Content Management System Agent (CMA), the Virtual Learning Environment Agent (VLE), and the Device Agents (DA) representing the different devices in the smart classroom (smartboards, robots, etc.).

In this paper, SML is very important because it combines AmI for education with academic services provided in the cloud. Using AmICL, the academic cloud services

can be invoked by the intelligent and non-intelligent agents in the smart classroom, to adapt and respond to the teaching and learning requirements of students. In this way, the SML allows to fuse the multi-agent paradigm with the cloud computing paradigm (see [12, 13] for more details). We use this layer in order to invoke the LA tasks required by the smart classroom.

This paper has the goal of presenting the LA tasks as services in a smart classroom. For that, we describe the way the agents of the smart classroom can invoke the LA tasks defined as services, using AmICL, but previously we define the relevance of the LA tasks in a smart classroom. Particularly, we propose a knowledge feedback loop based on LA tasks in a smart classroom. This paper is organized as follows: the next section presents the importance of the LA tasks in a smart classroom and defines a knowledge feedback loop using LA tasks; then, Sect. 3 presents AmICL and how it defines the LA tasks as services. Section 4 presents a case study in a smart classroom and, finally, the conclusions.

2 Learning analytics in a smart classroom

The different components of a smart classroom generate a large amount of data related to their activities, which offers the possibility of extracting valuable knowledge that may be employed to improve the learning process, to support the educational decision making of its components, among other things. For example, due to the different learning skills of the students, the course activities are performed in a dissimilar fashion by the students. Nevertheless, a smart classroom can overcome this problem by applying LA tasks to personalize the learning environment for each student. For that, the smart classroom requires an understanding and identifying of the different learning capacities of the students, to provide the necessary guidance to the students to improve their capabilities. In this way, the smart classroom can manage the diverse educational resources according to the profile of each student. The LA must provide the correct view of each student and the skills that the students have gained in the different activities during the learning process.

Sharing and manipulating this type of knowledge in real time is an enormous challenge in a smart classroom, in order to improve its behavior. This is the domain of application of LA tasks in a smart classroom. LA in a smart classroom must attempt to leverage all data present in it, to provide insight into the activities taking place within the learning process. LA must assess every aspect of the learning process in the smart classroom. In general, LA has two specific goals in a smart classroom [14]:

- To understand the learning process: in this case, LA must generate indicators about the current learning process (paradigm, methods, tools, etc.);
- To understand the students' behavior: in this case, the LA tasks must generate indicators about the performance of each student.

In order to use the LA tasks in a classroom, the data must be collected and prepared, but more important, the knowledge generated by the LA tasks must be relevant for the different components of the smart classroom. Particularly, this knowledge must be a vital feedback for them, specifically, to define the pedagogical model to apply in the smart classroom. LA tasks seek to understand entire systems and to support the components of the smart classroom in the decision making, in order to reach the following goals:

- Determine the reforms on the learning processes;
- Identify interventions and improvement strategies in a given moment to improve the students' performance.

In general, in the case of a smart classroom its components have different sources of data, they exploit the metrics generated by the LA tasks in different ways, among others aspects. In this way, the components of a smart classroom have different requirements of knowledge, and the knowledge generated by the LA tasks can be used in different ways. That requires the definition of the LA tasks as services, in a way that they can be used by the components of the smart classroom when they require.

However, it is very important to define how to organize the LA tasks to be used in a smart classroom. In the literature, there are a lot of LA tasks [1–3]. We propose a closed loop of LA tasks for the generation of knowledge in a smart classroom, in order to guarantee its autonomic behavior and adaptive process. For that, the LA tasks are defined as services, which help to understand the learning process and the students' behavior, to the components (agents) in the smart classroom. Particularly, the LA tasks can be classified according to their utilization [4]:

- *Determination of the educational context* The LA tasks' aim is to describe and to define the current situation in the learning process and the environment where it is carried out. Some of the LA tasks are: (a) Monitor efforts and progress of the students; (b) Analyze the evaluation process; (c) Discover the learning styles of the students; (d) Identify the students' problems and needs.
- *Planning of the improvement in the learning process* Once recognizing the learning process, it is necessary to determine what improvements can be introduced to it. In that sense, LA tasks must propose strategies, mechanisms, etc., which help to improve the dynamics

of learning that is taking in the smart classroom. In this case, some of the LA tasks required are: (a) Determine reforms on the learning processes; (b) Identify interventions and improvement strategies in a given moment; (c) Search resources and activities to recommend.

- *Analysis of the impact of the improvements* After introducing the strategies, etc. proposed by the LA tasks of the previous phase, implemented by the components of the smart classroom, it is necessary for new LA tasks to assess the impact of them in the learning process that takes place in it.

There is a repetitive cycle of these three phases, where their outputs are the inputs of the others. In this way, it is defined as a knowledge feedback loop, in order to optimize the learning process provided by the smart classroom. It is important to remark that LA tasks are exploited individually by the components (agents) of the smart classroom.

Particularly, in order to test the idea of a knowledge feedback loop, in this paper, we propose a set of LA tasks for the smart classroom, in order to reach the following set of goals:

- Monitor the learning process in the smart classroom;
- Define and uncover students' problems and needs;
- Identify improvement strategies of the learning process.

According to the previous classification, the LA tasks proposed are described in the following.

- *Determination of the educational context*: in this case, we propose three LA tasks as services to determine the current educational context:
 - *Recognition of the people in the smart classroom*: this task allows determining the students and professor present in the smart classroom;
 - *Description of the learning process in the smart classroom*: this task allows knowing the dynamic of the current learning process, with all the details (tools and learning resources used, etc.);
 - *Characterization of the learning process and students based on the assessment activities in the smart classroom*: this task defines the educational style of the students, the topics where they have problems or are not covered, etc.
- *Planning of the improvement in the learning process*: with the previous information, now, we define three LA tasks as services to improve the current learning process;
 - *Recommendation of learning resources in a smart classroom*: this task proposes new learning

resources for the learning process, based on the previous information;

- *Discovery of educational communities for a smart classroom*: this task identifies learning communities that are interesting for the students (they study similar topics, they have similar problems, etc.), in order to promote the collaboration and exchange;
- *Introduction of augmented reality as support at the learning process of theoretical concepts in a smart classroom*: this task proposes new presentations of the learning resources, closer to the interests of the students, in order to ease their understanding.
- *Analysis of the impact of the improvements*: now, we propose two LA tasks as services to determine the impact of the previous LA tasks in the learning process;
 - *Prediction of the performance in a learning process for a course*: this task allows predicting the overall performance of the students, after the introduction of the new strategies;
 - *Evaluation of the quality of the learning resources used in a smart classroom*: this task evaluates the quality of the new learning resources introduced to the learning process of the smart classroom. It has a high impact for the further recommendations.

3 An intelligent middleware to provide LA tasks as services for a smart classroom

This section presents AmICL and SML.

3.1 AmICL

AmICL is an Autonomic Reflective Middleware that enables the integration of objects (agents) in the educational environment, with educational resources in the cloud, in a flexible and adaptive way, so that objects in the smart classroom can adjust their behavior according to the context, and the requirements of the users. The details of AmICL can be found in [12, 13]. This middleware proposes five levels (see Fig. 1). The IE physical management layer represents the different devices in the environment, defined as agents. MMAL is composed of a multi-agent community to support the execution of multi-agent applications. This level follows the FIPA standard (see [12, 13]). The Services Management Layer (SML) has the responsibility of finding, searching, etc., services required by the applications. ILL represents the different software components in the educational platform. Particularly, it has two agents: the first is a profile agent to represent each student (SPA), and the second is the tutor agent (TA) to represent

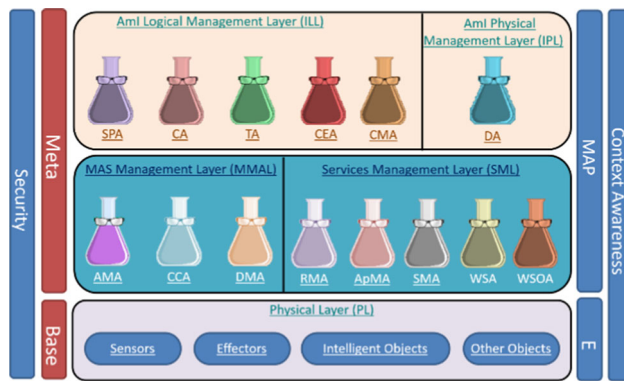


Fig. 1 AmICL architecture

the professor. Finally, IPL is where the different devices and software of a smart classroom (SaCI) are deployed.

The difference between AmICL and other similar middleware resides in the way that combines the AmI with academic services, which provides the cloud learning paradigm (C-Learning). In AmICL, the academic cloud services are combined with intelligent and non-intelligent objects in the environment, to adapt and respond to the users' teaching and learning requirements. In particular, the new aspect that fuses AmI with the cloud computing paradigm is SML [11]. This layer has some agents, which are an extension of FIPA standard, because the layer not only handles the resources and applications, but also educational services (locals and in the cloud), to adapt the AmI to current learning conditions required by its users.

In general, the agents of the middleware are, in MMAL: Agent Management of Agents (AMA), CCA (Communication Control Agent), and DMA (Data Management Agent), they are defined in [12, 13]; SML is defined by the Services Management Agent (SMA), the Web Service Agent (WSA), the Web Service Oriented Agent (WSOA), the RMA (Resource Management Agent), and the ApMA (Applications Management Agent). The agents of ILL are: the Student's Profile Agent (SPA), the Collaborative Agent (CA), the Tutor Agent (TA), the Content Management System Agent (CMA), and the Collaborative Environments Agent (CEA). Finally, the IE Physical Management Layer is composed of a single type of agent, the Device Agent (DA).

3.2 SML

A fundamental key in this research is to provide to SaCI agents the ability to invoke Web Services in the cloud. This point is very important, because the learning analytic tasks used in SaCI are available as services in the cloud. However, software agents use, commonly, FIPA protocols and

languages to communicate, while the Web Services use SOAP to exchange messages. To deal with this issue, AmICL includes a subsystem to allow a bidirectional communication between FIPA agents and Web Services. This subsystem is composed of the agents SMA, WSA, and WSOA in the SML layer, as shown in Fig. 1.

Specifically, this layer defines three agents (they are the bridge between the AmI and the cloud):

- **SMA:** controls, records, and manages Web Services (WS) available on the system so when an agent requires a specific service, it should contact this agent to locate the WSA that characterizes that service. The SMA is a hybrid between what is known in WS as Universal Description, Discovery, and Integration (UDDI), which is a platform-independent framework to describe, discover, and integrate WS, with an enterprise service bus (ESB) (this software architecture model is used in the design and implementation of communication between applications based on the SOA paradigm). SMA provides to SaCI the capability of register, deregister, and discover a Web Service by other agents of SaCI;
- **WSA:** is the logical representation of a Web Service in SaCI. It characterizes the WS, so it knows how to invoke them (it has their interfaces) and which are the requirements necessary to access the WS. The WSA has access to the services description file (WSDL) and creates a local proxy to consume the services' operations. There is one instance of it for each Web Service on the SMA. WSA allows other agents of SaCI to invoke a WS in the cloud in a natural way, by transforming the ACL message into a SOAP message and vice versa. In this way, agents and WS in the cloud can communicate transparently;
- **WSOA:** it is a façade for SaCI agents that need to expose their capabilities as a Web Service in the cloud, and it allows other SOA systems to invoke tasks of the SaCI agents in a transparent way. WSOA is a service-oriented agent [11–13], and this means that WSOA exposes the functionalities of its owner agent, as Web Services. Normally, WSOA receives a request to execute a task from a WS in SOAP format, transforms the request to an ACL message, and informs to its owner agent, who executes the task and informs the result to the WSOA. It takes the results, creates a SOAP message and sends the result to the requesting WS.

WSOA and WSA allow a bidirectional communication between agents of SaCI and WS in the cloud, so agents can consume WS and the WS can request the execution of agent tasks. However, in this paper we will only explain the

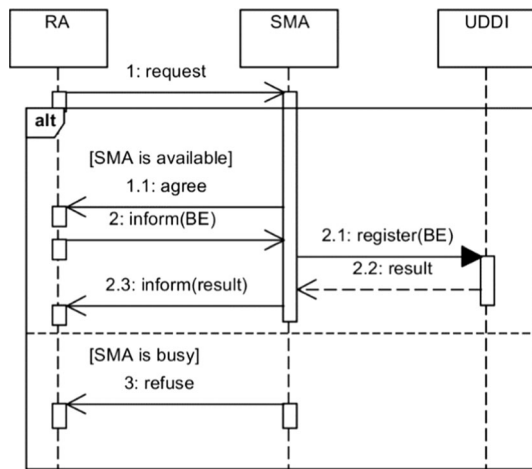


Fig. 2 Registering a business in the SaCI UDDI

communication from agents to the WS (how the agents of SaCI can consume tasks of LA, offered as WS in the cloud).

In this section, we will use some acronyms to refer to some elements of SaCI, namely: (1) RA: Requester Agent, the agent making a request like register, discover, or invoke a Web Service; (2) WSA: the Web Service Agent; (3) SMA: the Service Management Agent; (4) UDDI: The Universal Description Discovery and Integration component of SaCI.

3.3 Registering a Web Service in SaCI

The register of a Web Service in SaCI can be done by a human agent (by using the graphic interface provided by the SMA) or by a software agent. This process is completed in two phases. The first one is to register the business that provides the service. The second is to register the services associated with that business. Figure 2 shows the communication between the agents of SaCI at the moment of a business registration.

This process starts when the RA agent requests to SMA, the registration of a new business (1). SMA can reject (3) or accept (1.1) the request according to its occupation. When SMA accepts the request, it informs to RA (1.1). RA responds sending all the information about the business in an ACL message (2), the content of this ACL message is an XML text, structured using the template defined in Table 1. SMA receives the information and uses the UDDI API to invoke an operation over it to register the business (2.1). The UDDI proceeds to register the business in its database and returns the result about this operation to SMA (2.2), who resends this result to the RA (2.3), to inform whether the business was registered in the UDDI or not; this result also contains information about the unique identifier of the business in the UDDI (the business key), which is needed to register the services provided by the business, and also serves as input for the white, yellow, and green pages.

Table 1 XML document used to register a business in the UDDI of SaCI

```

<?xml version="1.0" encoding="UTF-8"?>
<businessDetail generic="2.0" xmlns="urn:uddi-org:api_v2" operator=
"http://www.mycompany.com/uddi" truncated="false">
  <businessEntity>
    <names>
      <name xml:lang="en">MyBusiness Name in English</name>
      <name xml:lang="es">Nombre de Mi Empresa en Español</name>
      ...
    </names>
    <descriptions>
      <description xml:lang="en">MyBusiness Description in english</description>
      <description xml:lang="es">Descripción de mi empresa en español</description>
      ...
    </descriptions>
    <contacts>
      <contact useType="employee position">
        <personName>employee name</personName>
        <emails>
          <email useType="Business">email1@companydomain</email>
          <email useType="Private">email2@domain</email>
          ...
        </emails>
      </contact>
    </contacts>
  </businessEntity>
</businessDetail>
  
```

Table 2 XML document used to register a Web Service in the UDDI of SaCI

```

<?xml version="1.0" encoding="UTF-8"?>
<businessServiceDetail generic="2.0" xmlns="urn:uddi-org:api_v2" operator=
"http://www.mycompany.com/uddi" truncated="false">
  <businessService>
    <businessKey>unique identifier of the business in the UDDI</businessKey>
    <names>
      <name xml:lang="en">Service Name in English</name>
      <name xml:lang="es">Nombre del servicio en Español</name>
      ...
    </names>
    <descriptions>
      <description xml:lang="en">Service Description in english</description>
      <description xml:lang="es">Descripción del servicio en español</description>
      ...
    </descriptions>
    <bindingTemplate>
      <accessPoint urlType="http">uri of the service</accessPoint>
      <tModelInstanceDetails>
        <tModel key="UUID of the technical model in the UDDI"/>
        <tModel key="UUID of the another technical model in the UDDI"/>
        ...
      </tModelInstanceDetails >
    </bindingTemplate>
  </businessService>
</businessServiceDetail>

```

The XML text shown in Table 1 defines the structure of a business entity object of the UDDI API (see [15]). That structure is composed by a unique identifier called business key, the name and the description of the business in different languages, and the contact information. Once the business has been registered, we can register the WS offered by it. This process is very similar to the process described in Fig. 2; however, in this case, the RA sends a request to SMA to register a Web Service. SMA can reject or accept the request. If SMA accepts, proceed to inform to RA. RA sends an ACL message to SMA with the information of the Web Service to be registered. The content of this ACL message is a text in XML format using the template described in Table 2. SMA proceeds to register the service using the corresponding UDDI API. The UDDI executes the process to register the Web Service and return the resulting data to the SMA. SMA sends this data to RA, so RA can know whether the Web Service was registered in the UDDI database or not.

Table 2 shows the information of the Web Service to be registered. This information fits into the structure of a business Service object of the UDDI API. The fields needed to register the WS are: the business key of the business offering this service, name and description of the Web Service in different languages, the access point of the Web Service (normally the URI of the Web Service), and the technical models (*tModel* is a data structure used to represent the type of the service, see [12, 13]) of the service, what can be used to discover the services in a more specific way.

3.4 Discovering WS by agents of SaCI

Once the WS has been registered in the UDDI of SaCI, the other agents of SaCI can discover those services by searching by name, description, or technical model. The SMA is in charge of executing the search request and creating a WSA instance that characterizes the service found. WSA is an intermediary point (proxy or gateway) between the requesting agent and the Web Service endpoint being invoked. The process of discovering a Web Service is detailed in the conversation of Fig. 3.

If SMA rejects the request, the conversation ends. However, when SMA accepts the request, proceed to inform to RA (1.1). RA sends the search criteria to SMA [(2) the search criteria can include a combination of business name, service name, service description, or technical model]. SMA prepares the search statement and executes it using the UDDI API (2.1). The UDDI returns the search result (2.2), and the resulting data are similar to the XML text shown in Table 2, though containing the information on all the services found. SMA sends this data to the RA (2.3); then, RA selects the service to be invoked according to its needs (3). Next, the RA informs its decision to SMA (4) sending an ACL message, whose content includes the URI of the selected Web Service. SMA proceeds to locate a WSA (or instance one, if this does not exist) that characterizes this Web Service (4.1), finally, SMA sends the agent id (AID) of the WSA selected as the proxy of the Web Service to be invoked (4.2).

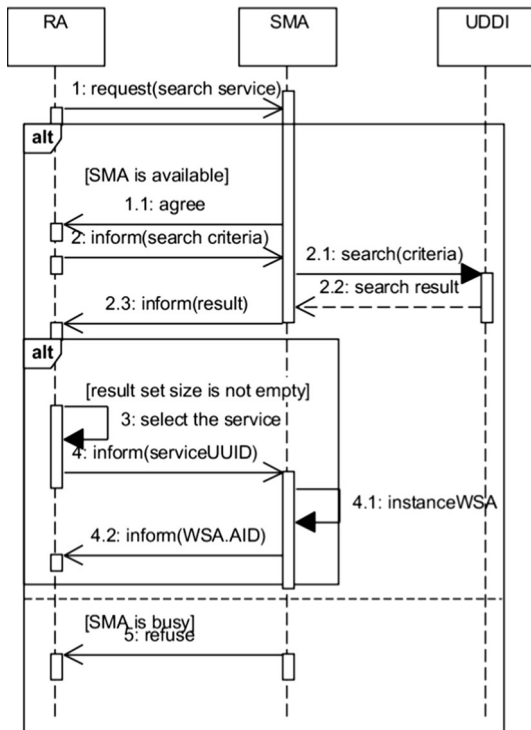


Fig. 3 Discovering a Web Service by SaCI's agents

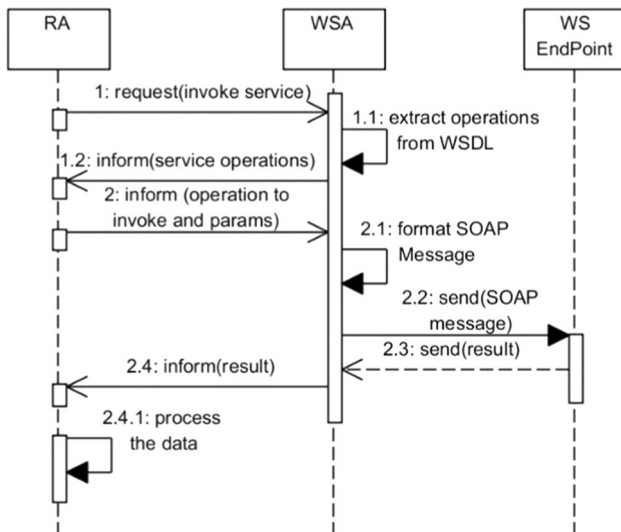


Fig. 4 Invoking a Web Service by a SaCI agent

3.5 Invoking a Web Service by agents of SaCI

Once the RA discovers a Web Service through SMA, it obtains the AID of a WSA, which serves as a proxy to invoke the Web Service endpoint. At this point, RA can communicate with that WSA to invoke the Web Service and get some result from it. Figure 4 shows the process of invoking the Web Service.

The conversation starts when RA sends a request to invoke the Web Service to WSA (1). WSA uses the WSDL file of the Web Service that represents, to obtain all the operations of the Web Service (1.1) and informs about that to RA (1.2, e.g. a Web Service to recommend digital contents have operations like: *findByAuthor*, *findByTitle*, *findByTopic*, etc.). This information is sent to the RA in the content of the ACL message, using an XML document like is shown in Table 3. The RA informs back to WSA about the operation to be invoked, as well as the values of its input parameters; to do that, it sends an ACL message (2) whose content is an XML document structured as described in Table 4. WSA receives this information and creates a SOAP message (2.1) what is used to invoke the Web Service endpoint through the SOAP API (2.2). WSA waits until it gets a response from the Web Service, and when it happens (2.3), takes the result data and resends it to RA (2.4) through an ACL message using the template in Table 5. RA uses those results according to its needs.

The information showed in Table 3 corresponds to the segment of the WSDL file that defines the operations of the Web Service. We are using the same format of the WSDL file to share information between agents, because we can take advantages of the existing SOAP parsers to obtain the information from the XML document (there is no need to write a new parser). The information contained in this XML document is the input and output parameters of each operation as well as the type of each parameter, and the name of each operation of the Web Service.

Table 4 shows the XML document used by an agent to inform to WSA, the operation of the Web Service that will be invoked, as well as the values of its input parameters. In this case, the name of the operation is “Operation1” and its input parameters are “param1Name” and “param2Name.” Once the WSA has these data, it can proceed to invoke that operation of the Web Service, by sending a SOAP message to it.

On the other hand, Table 5 shows the XML document containing the resulting information returned by the Web Service. This document allows RA to know that the Web Service returns 2 values (value1 and value2), as well as its values. This template corresponds to the response of a Web Service in SOAP format, whose format of response message was defined in Table 3 (Operation1Response).

4 Specification of the LA as services

In this section, we give some examples of LA tasks as services. For that, we define how to specify the LA tasks defined in Sect. 2 in the context of AmICL, considering several aspects, including:

Table 3 XML document containing the operations of the Web Service to be invoked

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- input param of the service operation1 -->
<message name="Operation1Request">
  <part name="param1Name" type="xs:string"/>
  <part name="param2Name" type="xs:integer"/>
  ...
</message>
...
<!-- output param of the service operation1 -->
<message name="Operation1Response">
  <part name="value1" type="xs:string"/>
  <part name="value2" type="xs:decimal"/>
</message>
...
<portType name="PortName">
  <operation name="Operation1">
    <input message="Operation1Request"/>
    <output message="Operation1Response"/>
  </operation>
  ...
</portType>
```

Table 4 XML document used to inform the operation to be invoked in the Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>Operation1</name>
  <param1Name>Param 1 Value as String</param1Name>
  <param2Name>Param 1 Value as integer</param2Name>
</invokeOperation >
```

Table 5 XML document used by WSA to return the result of the Web Service invocation

```
<?xml version="1.0" encoding="UTF-8"?>
<Operation1Response>
  <value1>Result value1 as string</value1>
  <value2>Result value2 as decimal</value2>
</Operation1Response>
```



Fig. 5 Original face (left) followed of the weighting scheme for 7 × 7 cells (right)

- Their specification as WSA (their registration as services);
- Their invocation by the agents of SaCI (their invocation as services);
- The utilization of their results by the agents of SaCI (the return of results);

4.1 Recognition of the people in the smart classroom

It is the first service of the first group of LA tasks, namely “Determination of the educational context.”

Service goal: face recognition of students to enter at the system.

Service description: the goal of this service is face recognition. The process first detects the presence of a face in an image or video stream and, then, subsequently identifies the face. This service has two phases:

- Face detection through local binary patterns [16], Haar cascade [17] and super vector machine algorithms [18], and other segmentation cascade classifier methods [19];
- Image Recognition.

Specification of the service: in this section, we will present an overview of the LBPs (local binary patterns) for the face recognition algorithm. The goal in the first phase is to have a dataset of images of students (10 images for each student). The service takes each image of a face and uses the local binary patterns (LBPs) approach as the face recognition algorithm. The original image of the face is divided into 7 × 7 cells (see Fig. 5).

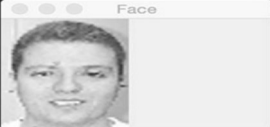
The white cells (such as the eyes) weighed 4 times more than the other cells. This means that we take the LBP histograms from the white cell regions and multiply them by 4. Light gray cells (mouth and ears) contribute 2 times more (i.e., we take the LBP histograms for these cells and multiply by 2). Dark gray cells (inner cheek and forehead) only contribute 1×. Finally, the black cells, such as the nose and outer cheek, are totally disregarded and weighed 0×. The weighted 7 × 7 LBP histograms are concatenated together to form the final feature vector (Table 6).

Table 6 XML document used to register this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<businessServiceDetail generic="2.0" xmlns="urn:uddi-org:api_v2" operator=
"http://saci.utpl.edu.ec/uddi" truncated="false">
  <businessService>
    <businessKey>743aab2a-58f4-11e6-8b77-86f30ca893d3</businessKey>
    <names>
      <name xml:lang="en">Recognition for the access to the smart classroom </name>
      ...
    </names>
    <descriptions>
      <description xml:lang="en">The goal in this service is face recognition, the
      first process is detecting the presence of a face in an image or video stream,
      and then subsequently identifying the face </description>
      ...
    </descriptions>
    <bindingTemplate>
      <accessPoint urlType="http">saci.utpl.edu.ec/facerecognition</accessPoint>
      <tModelInstanceDetails>
        <tModel key="8ad2b614-689b-11e6-8b77-86f30ca893d3"/>
        ...
      </tModelInstanceDetails >
    </bindingTemplate>
  </businessService>
</businessServiceDetail>
```

[INFO] gathering predictions...				
	precision	recall	f1-score	support
person_009	1.00	1.00	1.00	7
person_025	1.00	1.00	1.00	7
person_020	1.00	1.00	1.00	7
person_015	1.00	1.00	1.00	2
person_001	0.75	1.00	0.86	3
person_004	1.00	1.00	1.00	5
person_026	0.80	1.00	0.89	4
person_001	1.00	1.00	1.00	8
person_005	1.00	0.50	0.75	5
person_023	1.00	1.00	1.00	3
person_014	1.00	1.00	1.00	6
person_026	1.00	1.00	1.00	6
avg / total	0.98	0.97	0.97	63

[INFO] Prediction: person_025, Actual: person_025, Confidence: 47.32
[INFO] Prediction: person_004, Actual: person_004, Confidence: 52.62


Fig. 6 Precision in the recognition of student**Table 7** XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>FaceDetection</name>
  <image>Matrix of face
  recognition</image>
</invokeOperation >
```

The K -NN algorithm is used to find the closest distance between the detected face and the training data (faces dataset). K -NN (with $K = 1$) is performed with the Euclidean distance to find the closest face in the training data (see Fig. 6).

Table 8 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<FaceDetectResponse>
  <name>person_009</name>
  <confidence>95</confidence>
</FaceDetectResponse >
```

Finally, the face is associated with the prediction in the previous step (Table 7).

The name of the person associated with the face with the smallest Euclidean distance is chosen as the final classification (Table 8).

Results: the recognition of face to enter in the smart classroom has a value of 95% of confidence.

4.2 Description of the learning process in the smart classroom

Service goal: supervise the progress and performance of students during the learning process in the smart classroom.

Service description: in a smart classroom [1], one of the main features is the ability to adapt to the needs of students; on the other hand, in a classroom there is a lot of information about the teaching–learning process. Therefore, it is necessary to apply tasks of learning analytics for supervising the progress and performance of students

Table 9 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>ClassifyStudents</name>
  <Students>SetStudents (with score,
  genre, etc.)</Students>
</invokeOperation >
```

Table 10 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<ClassifyStudentsResponse>
<Classification>SetClassificationStudents</Classification>
  <confidence>95</confidence>
</ClassifyStudentsResponse>
```

during the learning process. This service is registered in the same way that the previous service.

Specification of the service: the service begins to obtain a classification of students considering parameters such as status (approved, reprobate), qualitative grade [poor (0–13), insufficient (14–27), sufficient (28–29), satisfactory (30–32) or (33–35), remarkable (36–38) outstanding (39–40)], and gender (male, female—see Table 9).

Results: the service returns as result the percentage of approvals and the percentage of approval by gender (see Table 10).

4.3 Characterization of the learning process and students based on the assessment activities in the smart classroom

Service goal: apply LA tasks in assessment process to determine the skills developed by the students in a smart classroom.

Service description: a way to analyze the learning process in a smart classroom is by using the information about the progress of the students. For that, it is necessary to apply LA tasks in the evaluation process. In particular, it is possible to determine the skills developed by the students, according to their scores in the assessments. In a smart classroom, these assessments are available for the student through the VLE.

According to [20], VLE has some activities to help students to interact with the contents, peers or the teacher, such as questionnaires and homework. In this case, the

Table 11 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>evaluationPatterns</name>
  <Course>C010</Course>
  <Question>Set_Questions</Question>
  <Score>Set_Score_Students</Score>
</invokeOperation>
```

Table 12 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<evaluationPatternsResponse>
  <Pattern> If Chat I and
  Chat II and Vid-Coll I then TOTAL
  </Pattern>
</evaluationPatternsResponse>
```

questions' assessment is based on the learning domain that covers a topic using the Bloom's taxonomy, which considers six categories [5, 20], namely knowledge, comprehension, application, analysis, synthesis, and evaluation.

Specification of the service: the LA service has the following components: (1) Extract the information from the smart classroom around the evaluation process; (2) Label the questions; (3) Classify the questions according to the Bloom's taxonomy; (4) Analyze the questions; (5) Discover the pattern of the evaluation process in a smart classroom. These components are detailed in [20] (see Table 11).

Results: this service returns the relationships among the factors presented during the evaluation process in a smart classroom, and the measures of support and confidence to validate the results of the model. Then, the agent VLE uses this knowledge about the learning process to determine aspects not covered during the evaluation, or new tools, activities, etc., to be incorporated in a course, to improve the performance of the students, etc. (see Table 12).

4.4 Recommendation of learning resources in a smart classroom

It is the first service of the second group of LA tasks, "Planning of the Improvement of the learning process: with the previous information."

Service goal: it provides high-quality recommendations of online learning activities or educational resources to

Table 13 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>intelligentRS</name>
  <resources>dataset of educational
resources and users</resources>
  <indicators>Indicators of the use
and preferences of the
users</indicators>
</invokeOperation >
```

Table 14 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<intelligentRSResponse>
  <resources>learning resources
</resources>
  <activities>learning activities
</activities>
</intelligentRSResponse>
```

students in the smart classroom, to assist the learning process.

Service description: the main components of this service are [21]:

- A knowledge acquisition mechanism based on learning algorithms;
- A knowledge explicit modeling, which represents all the knowledge necessary to recommend;
- A reasoning mechanism to infer information from the stored knowledge;
- A criticality system based on the automatic inference capability of the recommender system;

This system not only recommends items based on specific knowledge about how the item features meet user needs and preferences, like classical knowledge-based recommendation approaches, this system also uses the rating of the items like the collaborative approach, discovers the aspects, interests, properties that the user would like about the items (criticality system), infers the rating of the items, etc.

The SR exploits all the knowledge, which is obtained automatically (by learning mechanisms), and is modeled appropriately, in order to be used by its reasoning mechanism to infer how much the user needs the item, if the user's goals are satisfied by the current items, etc. The register of this service in SaCI is shown in Table 15.

Specification of the service: this service uses a general framework for an intelligent recommender system (IRS)

Table 15 XML document used to register this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<businessServiceDetail generic="2.0" xmlns="urn:uddi-org:api_v2" operator=
"http://saci.utpl.edu.ec/uddi" truncated="false">
  <businessService>
    <businessKey>743aab2a-58f4-11e6-8b77-86f30ca893d3</businessKey>
    <names>
      <name xml:lang="en">Recommender system of learning resources </name>
      <name xml:lang="es">Sistema Recomendador de Recursos Educativos</name>
      ...
    </names>
    <descriptions>
      <description xml:lang="en">Provide high-quality recommendations to students in
the smart classroom, this mean, that can recommend on-line learning activities
or educational resources, to assist the learning process </description>
      <description xml:lang="es">Provee recomendaciones de alta calidad a los
estudiantes in el salón de clase inteligente, que pueden ser actividades de
aprendizaje online o recursos educativos para dar soporte al proceso de
aprendizaje.</description>
      ...
    </descriptions>
    <bindingTemplate>
      <accessPoint urlType="http">saci.utpl.edu.ec/reslr</accessPoint>
      <tModelInstanceDetails>
        <tModel key="fbc88f9a-68c0-11e6-8b77-86f30ca893d3"/>
        ...
      </tModelInstanceDetails >
    </bindingTemplate>
  </businessService>
</businessServiceDetail>
```

Table 16 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>IdentifyCommunities</name>
  <id_student>set of
  students</id_student>
  <trust_model>the trust model to be
  used</trust_model>
</invokeOperation>
```

Table 17 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<IdentifyCommunitiesResponse>
  <id_communities> the id of the
  communities discovered</id_communities>
  <StudentsbyCommunities>
    The set of students' member of
    each community
  </StudentsbyCommunities>
  <Pattern_communities>
    The set of patterns of each
    community based on the activities and
    resources used
  </Pattern_communities >
</IdentifyCommunitiesResponse>
```

**Fig. 7** Student interaction with the “Aumentaty Author” [23]

that exploits knowledge, learns, discovers new information, infers preferences, and criticisms, among other things [20]. The intelligent recommender system is based on this framework, using Fuzzy Cognitive Maps (FCMs). In [21] the utilization of IRS based on FCMs, the experiments, and the analysis of the results are presented. The invocation of this service by the agents of the smart classroom is presented in Table 13.

Results: this service returns the learning activities and resources recommended to students on the smart classroom. The agents use this type of recommendation for giving support to students in the learning process involved in the smart classroom (see Table 14).

4.5 Discovery of educational communities for a smart classroom

Service goal: identify communities of students during the learning process.

Service description: identifies virtual educational communities (VLC) through a trust model [22]. This model uses the principles of social computing and quantifies the degree of affinity between students, which are part of the learning communities.

Specification of the service: this service uses the trust model defined in [22]. First, the students interact in VLC for about four weeks; then, the VLE agent invokes the service of data mining for the discovery of the communities. Then, the service obtains the dataset (id_student, trust_value, id_activities/resource) from the trust model. The service returns the predominant patterns of the communities. The invocation of this service by the VLE agent is shown in Table 16.

Results: this service returns the relationship between the communities and students, and the pattern of the communities, based on the activities and resources used in it. These values are summarized in Table 17.

4.6 Introduction of augmented reality as support at the learning process of theoretical concepts in a smart classroom

Service goal: reinforce learning theoretical concepts (LTC).

Service description: shows scenes of augmented reality (AR) for students, in order to reinforce the theoretical concepts taught by the teacher.

Specification of the service: this service uses the SRA (the agent that provides services of AR) defined in [23]. SaCI uses markers, visors that are managed by the VLE agent, and displayed by the SRA (see Fig. 7). It can be used in different ways, for example, to show scenes where the multivariate calculus is taught. The teacher and students share the markers, using the software “Aumentaty Author” and the VLE agent. The VLE agent requests the data of the students and of the course (id_student, id_sceneAR, id_-markeractivities/resource), in order to determine when visualize the scenes. Then, it invokes this service as shown in Table 18.

The AR helps in the learning process, because it allows the contextualization of the knowledge in the teaching–learning processes in SaCI. Specifically, one of the goals of SaCI is allowing the interaction between the students with the real world. In that sense, AR allows the contextualization of the knowledge to the needs of the real word of each student, in order to generate an individualized context for each student. In previous experiences of the utilization

Table 18 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>ReinforcementLTC</name>
  <id_sceneAR>the id of
  sceneAR</id_sceneAR>
  <id_marker> the id of marker
</id_marker>
</invokeOperation>
```

**Fig. 8** Student interaction with the VLE agent [23]**Table 19** XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
< ReinforcementLTCResponse>
  <Scene id="scene 001">
    <markerId>id_mkrl </markerId>
    <markerId>id_mkr3 </markerId>
    ...
  <students_using>
    <student_id>student1</student_id>
    <student_id>student2</student_id>
    ...
  </students_using>
  <markers-list>
    <makerId>id_mkrl</makerId>
    ...
  </markers-list>
  </Scene ID>
</ReinforcementLTCResponse>
```

of RA in a SaCI [23], the teacher and students shared markers, the SRA provides surveys to evaluate the quality of the learning process using RA, and the students answer the surveys at the end of a course via the VLE agent.

Results: this service returns markers in the scenes of AR (see Fig. 8). Additionally, it can show the list of students using markers and the list of markers more used. These values are summarized in Table 19.

4.7 Prediction of the performance in a learning process for a course

It is the first service of the last group of LA tasks, “Analysis of the impact of the improvements.” The register of this service in SaCI is shown in Table 20.

Service goal: this service predicts the academic performance of a student.

Service description: the goal of this service is to predict the student’s score at any time of the academic semester. Figure 9 shows the prediction process [14, 24], the previous courses are the historical data from one or two years ago about the scores of the tests. The current course is where it is applied the predictive algorithm. In this LA task clustering algorithms are used, based on machine learning techniques: Naive Bayes, KNN, Bayes Net, etc.

Specification of service: the prediction process includes the following steps:

- Upload the historical data. One or more files with the course scores. The data files have the .csv or .xls format and the following name structure: <Subjetct_name>_Curso_<Begin_year>_<End_year>;
- Request prediction. The service looks for the best quality in the classification between the different clustering algorithms;
- Obtain the result. The states of prediction are: “Approve”, “Suspend,” and “Excellent” and are determined for each student, according to his scores in the exams (E1, ..., E3) and practices (PART 1) (Table 25). Its invocation by the agents of SaCI is shown in Table 21.

Results: the result is predictions of the “states” of the students, which SaCI can use to take some decision about new activities to be included during the learning process, to avoid the students with “Suspend” state (see Table 22).

4.8 Evaluation of the quality of the learning resources used in a smart classroom

Service goal: evaluate the quality of the Open Educational Resources (OER).

Service description: a function of the VLE agent in SaCI is shown OER. The rapid growth of OER provides new opportunities for teaching and learning, therefore, is important to validate how they contribute to the learning process. The service analyzes the use of OER in SaCI, in order to obtain a ranking of the best OER, according to their popularity and scores of the students.

Specification of the service: the service begins the operation to obtain the ranking of popularity of OER, considering parameters such as resource type (book, image, software). It builds the learning resources pattern of the

Table 20 XML document used to register this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<businessServiceDetail generic="2.0" xmlns="urn:uddi-org:api_v2" operator=
"http://saci.utpl.edu.ec/uddi" truncated="false">
  <businessService>
    <businessKey>743aab2a-58f4-11e6-8b77-86f30ca893d3</businessKey>
    <names>
      <name xml:lang="en">Score prediction</name>
      ...
    </names>
    <descriptions>
      <description xml:lang="en">The goal of this services is to predict the student's
        score at any time of an academic semester </description>
      ...
    </descriptions>
    <bindingTemplate>
      <accessPoint urlType="http">saci.utpl.edu.ec/scrpredict</accessPoint>
      <tModelInstanceDetails>
        <tModel key="eea0342a-68c6-11e6-8b77-86f30ca893d3"/>
        ...
      </tModelInstanceDetails >
    </bindingTemplate>
  </businessService>
</businessServiceDetail>
```

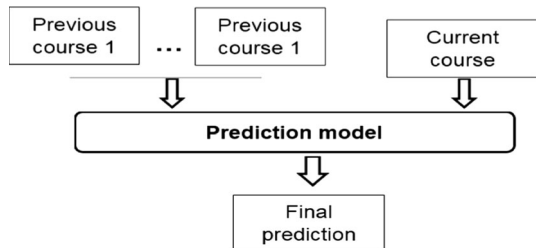


Fig. 9 Prediction process

Table 21 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>predict</name>
  <data>
    <file>historical data filename URI (.csv
    or .xls) with score</file>
  </data>
  <type>Prediction fast or comprehensive
  prediction</type>
</invokeOperation >
```

best OERs for a level. For each OER, we analyze its properties, in order to determine the general properties, which must have an OER for the students in a given course. We calculate the quality of each property *i* of an OER, using Eq. 1:

Table 22 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<predictResponse>
  <student id="1">Excellent</student>
  <student id="2">Approve</student>
  <student id="3">Suspend</student>
  ...
</predictResponse>
```

$$Q(i) = \frac{\sum_{k=1}^n \sum_{\exists j \text{ Property} REA_{ki}^j = 1} \sum_{j \in \text{course_}k} P_k \left(\frac{PU_j^k}{TotalRanking_k} \right)}{n} \tag{1}$$

where *n* is the number of courses in a given level, *m_j* is the number of OERs used in the course *j*, *course_k* are the courses in a level, *P_k* is the average performance of the students in the course *k*, *PU_j^k* is the percentage of utilization of this OER *j* during the course *k*, *TotalRanking_k* is the global utilization of the OERs during the course *k*, and *PropertyREAJ_{ki}^j* is a binary value (1 if the property *i* has the OER *j* used in the course *k*; 0 otherwise). We select the properties *i* for the OER pattern, where *Q(i) > threshold*. Its invocation by the agents of the smart classroom is shown in Table 23.

Results: the service returns the ranking of popularity of each OER, according to the properties of each OER

Table 23 XML used to invoke this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<invokeOperation>
  <name>RankingPopularityOER</name>
  <SetOER>Set of OERs to evaluate
  their quality</SetOER>
</invokeOperation >
```

Table 24 XML used by WSA to return the result of this Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<RankingPopularityOERResponse>
  <Ranking> Set of OERs ordered
  according to their quality </Ranking>
  <PatternOER>
    <CourseID>001 </ CourseID >
    <Pattern>pattern of OERs to be used
    in this course </Pattern>
    ...
  </PatternOER>
</RankingPopularityOERResponse>
```

Table 25 Result of prediction

ID_USER	E1	E2	E3	PART 1	STATE PLANNED
1	10	1	1	3, 5	Approve
2	0	0	0	0	Suspend
3	0	6	0	5	Approve
4	4	0	0	0	Suspend
5	4	8, 5	7, 5	7	Approve
6	10	9	9	8	Excellent
7	0	0	0	3, 75	Suspend
8	9	0	0	2, 75	Approve
9	10	9, 8	9, 75	6, 55	Excellent
10	5	9, 5	7, 5	2, 35	Approve
11	0	3	0	3,75	Suspend
12	0	0	0	0	Suspend
13	5	7	4, 5	3, 25	Suspend
14	0	0	0	0	Suspend
15	7, 8	9, 5	0	4	Suspend

contained in the final OER pattern. Additionally, it returns the OER pattern of the courses. These responses are summarized in Table 24.

5 Conclusions

In this paper, we have proposed a framework to support LA as a service in smart classrooms. Our approach works with two different paradigms, in order to support LA as a

service: multi-agent systems and cloud computing. We have shown the functionality of the cloud component of AmICL, to guarantee the access to the academic services in the cloud for any agent. In our case, these services are LA tasks (Table 25).

Additionally, we have proposed a knowledge feedback loop in a smart classroom based on LA tasks, in order to improve the learning process. Each LA task is implemented as a service and can be used at the moment that is required, for those who require it. The components (agents) of the smart classroom can invoke the services individually, in a very easy way, to adapt them to the requirements of the students. In particular, the invocation of the services is transparent, in order to exploit the knowledge present at the smart classroom. Thus, AmICL uses the LA paradigm, to allow the adaptation of the agents to the situations and requirements that occur in the smart classroom. In this way, our approach is very adaptable, scalable, and interoperable.

We have tested LA services on the cloud for complex tasks, which are used by the smart classroom in order to improve the quality of the learning process. In general, the scalability and flexibility properties of an AmICL are exploited very well for invoking the LA tasks.

According to our knowledge, there are no previous works that involve the definition of tasks of learning analytics as services, and a knowledge feedback loop based on these learning analytic tasks, in order to reach the objectives of SaCI. The closed loop is possible to develop, because our framework combines the cloud and multi-agent paradigms in SaCI, making possible the invocation of services that can perform the tasks of learning analysis, dynamic and emergently. This represents a great contribution, because it allows to extend SaCI with tools of discovery of knowledge in the cloud.

Acknowledgements Dr Aguilar has been partially supported by the Prometeo Project of the Ministry of Higher Education, Science, Technology and Innovation of the Republic of Ecuador.

References

- Valdiviezo, P., Cordero, J., Aguilar, J., Sánchez, M.: Conceptual design of a smart classroom base on multiagent system. In: de International Conference on Artificial Intelligence (ICAI'15), pp. 471–477 (2015)
- Larsson, J.A., White, B.: Learning Analytics. From Research to Practice. Springer, New York (2014)
- Riofrio, G., Encalada, E., Guaman, D., Aguilar, J.: Business intelligence applied to learning analytics in student-centered learning processes. In: XLI Conferencia Latinoamericana en Informática (CLEI), pp. 567–576 (2015)
- Aguilar, J., Valdiviezo, P., Cordero, J., Riofrio, G., Encalada, E.: A general framework for learning analytic in a smart classroom. In: Valencia-García, R., et al. (eds.) Technologies and Innovation.

- Communications Computer and Information Science Series, vol. 658, pp. 214–225. Springer, Berlin (2016)
5. Cruz Benito, J., Therón, R., García Peñalvo, F.J., Lucas, E.P.: Discovering usage behaviors and engagement in an Educational Virtual World. *Comput. Hum. Behav.* **47**, 18–25 (2015)
 6. Muñoz Merino, P.J., Ruipérez Valiente, J.A., Alario Hoyos, C., Pérez Sanagustín, M., Kloos, C.D.: Precise effectiveness strategy for analyzing the effectiveness of students with educational resources and activities in MOOCs. *Comput. Hum. Behav.* **47**, 108–118 (2015)
 7. Gómez Aguilar, D.A., Hernández García, Á., García Peñalvo, F.J., Therón, R.: Tap into visual analysis of customization of grouping of activities in eLearning. *Comput. Hum. Behav.* **47**, 60–67 (2015)
 8. Hershkovitz, R., Nachmias, R.: Learning about online learning processes and students' motivation through web usage mining. *Interdiscip. J. Knowl. Learn. Objects* **5**, 197–214 (2009)
 9. García-Saiz, D., Zorrilla, M.: E-learning web miner: a data mining application to help instructors involved in virtual courses. In: 4th International Conference on Educational Data Mining, Netherlands, pp. 323–324 (2011)
 10. Kalles, D., Pierrakeas, C.: Analyzing student performance in distance learning with genetic algorithms and decision trees. *Appl. Artif. Intell.* **20**(8), 655–674 (2006)
 11. Sánchez, M., Aguilar, J., Cordero, J., Valdiviezo-Díaz, P., Barba-Guamán, L., Chamba-Eras, L.: Cloud computing in smart educational environments: application in learning analytics as service. In: Rocha, A., Correia, M., Adeli, H., Reis, P., Mendonca, M. (eds.) *New Advances in Information Systems and Technologies*, pp. 993–1002. Springer, Berlin (2016)
 12. Sánchez, M., Aguilar, J., Cordero, J., Valdiviezo, P.: Basic features of a reflective middleware for intelligent learning environment in the cloud (IECL). In: Asia-Pacific Conference on Computer Aided System Engineering (APCASE'15), pp. 1–6 (2015)
 13. Sánchez, M., Aguilar, J., Cordero, J., Valdiviezo, P.: A smart learning environment based on cloud learning. *Int. J. Adv. Inf. Sci. Technol. (IJAIST)* **39**(39), 39–52 (2015)
 14. Valdiviezo-Díaz, P., Cordero, J., Reátegui, R., Aguilar, J.: A business intelligence model for online tutoring process. In: 45th Annual Frontiers in Education (FIE) Conference, pp. 2224–2232 (2015)
 15. Kumar, S.A., Vijayalakshmi, M.: Efficiency of data mining techniques in edifying sector. *Int. J. Adv. Found. Res. Comput.* **1**(8), 56–62 (2014)
 16. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: European Conference on Computer Vision. Springer, Berlin, pp. 469–481 (2004)
 17. Xiao, R., Zhu, H., Sun, H., Tang, X.: Dynamic cascades for face detection. In: 11th IEEE International Conference on Computer Vision, pp. 1–8 (2007)
 18. Guo, G., Li, S.Z., Chan, K.: Face recognition by support vector machines. In: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 196–201 (2000)
 19. Barba-Guamán, L., Quezada-Sarmiento, P., Calderon-Cordoba, C., Ordoñez López, J.: Detection of the characters from the license plate by cascade classifiers method. In: Future Technologies Conference (FTC), pp. 1–6 (2016)
 20. Aguilar, J., Valdiviezo, P., Riofrío, G.: A general framework for intelligent recommender systems. *Appl. Comput. Inf.* 1–14 (2016)
 21. Aguilar, J., Valdiviezo, P., Riofrío, G.: A fuzzy cognitive map like recommender system of learning resources. In: IEEE World Congress on Computational Intelligence (WCCI), pp. 1539–1546 (2016)
 22. Chamba-Eras, L., Arruarte, A., Elorriaga, J.A.: Modelo de confianza para comunidades virtuales de aprendizaje. In: XVI Congreso Internacional de Informática Educativa 2011 (TISE), Santiago de Chile (2011)
 23. Chamba-Eras, L., Aguilar, J.: Design of an augmented reality component from the theory of agents for smart classrooms. *Revista IEEE América Latina* **14**(8), 3826–3837 (2016)
 24. Salamó, M., Rodríguez, I., López Sánchez, M., Puig, A., Balocco, S., Taulé, M., Rodríguez, J.: GradeForeseeer: recurso docente para la predicción de notas del alumnado de informática. In: XXI Jornadas de la Enseñanza Universitaria de la Informática, Universitat Oberta La Salle, pp. 89–96 (2015)