



Red 513RT0471

Nivel de Diseminación: PÚBLICO

D21

Metodología de Integración de Objetos de Aprendizaje Avanzados en Sistemas de E-Learning

Manuel Freire Morán

Baltasar Fernández Manjón

Departamento de Ingeniería del Software e Inteligencia Artificial

Grupo e-UCM, Cátedra Telefónica en Educación Digital y Juegos Serios

Facultad de Informática. Universidad Complutense de Madrid.

España

1.1 Introducción

Este documento describe una metodología de integración de Objetos de Aprendizaje Avanzados (OAAs) en sistemas de e-Learning, centrándose en Juegos Serios (dónde se incluyen también simulaciones educativas y simulaciones con estrategia de juego), que se pueden considerar como un ejemplo de los OAAs más complejos y representativos dentro de este ámbito educativo.

La sección 1 proporciona una introducción breve a las necesidades y opciones que existen a la hora de integrar OAAs y sistemas eLearning. La sección 2 describe las soluciones elegidas en mayor detalle, y describe una arquitectura concreta para llevar esta integración a cabo. Finalmente, la sección 3 resume los resultados y presenta conclusiones.

1.1.1 Marco teórico: LMSs

Existe gran variedad de plataformas eLearning, un término que potencialmente engloba cualquier solución software que de soporte al aprendizaje. En el marco del proyecto RIURE, nos concentraremos en instalaciones institucionales de sistemas LMS (Learning Management System) estándares tales como Moodle o Sakai, frecuentes en las instituciones educativas de nivel terciario de los países representados en el proyecto.

Este documento no cubre los detalles de la integración de OAAs que son comunes con objetos de aprendizaje estándar. Se asume que la integración de OAAs se realizaría en despliegues institucionales de LMSs capaces de acoger OAs estándar (e.g. Moodle es capaz de desplegar OAs empaquetados en Comon Cartridge).

En lugar de describir integraciones en entornos o sistemas concretos, empezaremos por definir una serie de características mínimas que esperamos encontrar en cualquier instalación que use LMSs estándares con propósitos educativos. Estas características son:

- Capacidad de desplegar objetos de aprendizaje estándar (no avanzados) usando estándares como LTI [1], Common Cartridge [2] ó SCORM [3].
- Gestión de identidad institucional, usando “single sign-on” para permitir a los usuarios del LMS y otros sistemas institucionales navegar identificados sin necesidad de proporcionar sus credenciales repetidamente.

1.1.1.1 Despliegue de objetos de aprendizaje en instituciones

Hay varios tipos de despliegues posibles, en función del contenido a desplegar:

- Por tipo de despliegue:
 - Web: acceso sin salir del navegador; por ejemplo, un recursos html5. Presenta ventajas en cuanto a comodidad (no requiere instalación o despliegue específico) e inmediatez (no es necesario descargar todo el recurso antes de empezar a usarlo).
 - Nativo: acceso vía una aplicación nativa a la plataforma. Esta aplicación debe ser descargada antes de poderse ejecutar. Permite un mayor rendimiento y control que un despliegue web, a cambio de un mayor coste de despliegue (y normalmente de desarrollo al necesitar una aplicación nativa para cada plataforma).
- Por localización del recurso:
 - Remoto: el recurso es externo al LMS, y el LMS sólo contiene una ruta en forma de URL
 - Local: el contenido del OA se sirve directamente desde el LMS, que habitualmente impone límites al tamaño del mismo; y en el caso de acceso vía navegador
- Por comunicación con el entorno educativo institucional:
 - Sin comunicación: el OA no recibe ni devuelve información a/del LMS; por ejemplo, un documento ó un conjunto de diapositivas.
 - Comunicación de resultados: el OA devuelve un resultado al LMS al finalizar cada sesión de uso. Por ejemplo, un ejercicio que incluya evaluación podría devolver la nota del alumno.
 - Comunicación bidireccional: el OA recibe información en el momento de lanzarse (por ejemplo, de configuración), y devuelve resultados durante y/o tras la sesión.
- Por grado de persistencia de las sesiones

- No persistentes: el objeto de aprendizaje no tiene concepto de sesión. Por ejemplo, un conjunto de diapositivas.
- Informe de resultados: cada vez que se cierra la sesión, se envía un resultado al LMS. El OA no necesita acceso a la identidad del usuario para hacer este informe, sino que se limita a devolver la puntuación obtenida junto con el identificador temporal asignado por el LMS en el momento de arrancar la actividad.
- Identidad persistente: cada interacción del usuario en una sesión con el OA es registrada, y el sistema puede reconocer a un usuario concreto de sesión en sesión.

La comunicación con el entorno educativo institucional y la persistencia de sesiones van mano a mano: no es posible implementar un OA con identidad persistente a no ser que se disponga de un mecanismo de integración que permita al contenido conocer la identidad de sus usuarios.

En general, siempre es posible desplegar un OA que soporta un grado alto de comunicación y persistencia a un nivel inferior, a costa de perder parte de su funcionalidad. Por ejemplo, es posible desplegar un examen interactivo e ignorar su resultado (y todavía podría tener sentido como ejercicio de práctica o autoevaluación).

1.1.1.2 Gestión de identidad institucional y LMSs

En cualquier institución educativa, es deseable unificar la gestión de identidad institucional, de forma que ésta esté disponible de forma transparente en todos los sistemas (LMS, biblioteca, correo web, etcétera) que usan sus integrantes. Las ventajas de esta gestión unificada, también denominada “single sign-on” (SSO, por sus siglas en inglés), son múltiples:

- Gestión: Facilita el alta, modificación y gestión general de las identidades de los integrantes de la comunidad institucional, ya que los cambios sólo deben ser efectuados en un único sistema. La alternativa sería mantener sincronizados una multitud de sistemas distintos.
- Seguridad: Disminuye la superficie de vulnerabilidad de la institución a ataques ó fallos informáticos; ya que basta con proteger muy bien el único componente crítico (el sistema de gestión de identidad) para proteger todos los servicios que dependen de esta identidad. La alternativa, con varias copias de las credenciales distribuidas por los respectivos sistemas, es que cualquier fuga de información en uno de ellos comprometería al resto.
- Comodidad: Los usuarios no tienen que recordar varias contraseñas ni ser conscientes de los cambios que se realizan en la red. En un sistema SSO, una vez dentro del sistema, los usuarios no tienen porqué ser conscientes de los distintos sistemas que les están proporcionando cada servicio; de forma que es posible añadir o modificar sistemas adicionales sin impacto alguno para los usuarios.

Algunos sistemas de gestión de identidad almacenan más información que otros. Para este trabajo, nos concentraremos en los siguientes datos:

- Autenticación básica: el sistema permite determinar qué nombre de usuario interno (y único en el sistema) corresponde a unas credenciales dadas.
- Roles: el sistema permite determinar la categoría de usuario; por ejemplo, “personal docente”, “estudiante”, ó “personal de servicio”.
- Roles específicos: el sistema proporciona roles específicos; por ejemplo, puede determinar en qué clases enseña un profesor dado; o si un usuario es ayudante del profesor de un grupo

de prácticas determinado; o si un alumno está enrolado en un grupo concreto en una materia dada en el curso actual.

Nótese que esta información no tiene por qué estar toda en el mismo sistema físico. Por ejemplo, una institución dada podría usar LDAP para autenticación y roles básicos, y Moodle para los roles específicos.

Es conveniente en este punto clarificar la distinción entre autenticación y autorización:

- **Autenticación:** permite verificar la identidad de un usuario, dadas sus credenciales de autenticación. Por ejemplo, nombre de usuario y contraseña, la autenticación nos permite asegurar que el usuario es conocido por ese nombre en el sistema, y es quien dice ser.
- **Autorización:** permite verificar si un usuario previamente autenticado tiene o no permisos para realizar una acción dada.

Ambas son necesarias en el contexto de acceso a un LMs: por ejemplo, antes de permitir a un profesor alterar contenidos de un curso, es importante verificar que el profesor es quien dice ser, y que efectivamente tiene permisos para hacer cambios en el correspondiente curso.

1.1.1.3 Estándares para interoperar con sistemas de gestión de identidad

Cabe distinguir dos estándares para gestión de identidad institucional. El primero, SAML [4] (“Security Assertion Markup Language”, traducible como lenguaje de marcado para aserciones de seguridad) permite, en su faceta básica, negociar autenticación, pero no proporciona información acerca de roles. El segundo, BLTI, forma parte de LTI [5] (“Learning Tools Interoperability”, un estándar para la interoperabilidad de herramientas de eLearning desarrollado por el consorcio internacional IMS), y permite comunicar tanto autenticación como autorización.

1.1.2 Marco teórico: OAAs

Los Objetos de Aprendizaje Avanzados se definen, en el marco del proyecto RIURE, como aquellos que presentan un nivel de interactividad considerablemente superior al de los OAs “estándares”. Por ejemplo, los juegos educativos, las simulaciones, ó los videos interactivos serían tres ejemplos de OAAs.

El resto de este documento va a centrarse en un tipo concreto de OAAs: los juegos serios (Serious Games; ó SGs por sus siglas en inglés). Las diferencias entre SGs y simulaciones educativas son más narrativas que técnicas: es posible implementar un SG sobre una simulación añadiendo una capa narrativa (nosotros incluso hablamos de simulaciones con estrategia de juego en esos casos). Por ejemplo, dada una simulación de un laboratorio de hematología, se podría crear un SG para enseñar a determinar el hematocrito [6]. Un vídeo interactivo también se puede ver, en este sentido, como un caso particular de juego: un juego con gráficos realistas debería ser indistinguible de un vídeo interactivo que ofreciera los mismos puntos de decisión. Por tanto, en el resto del presente documento, se considerará que los juegos serios son OAAs prototípicos.

La alta interactividad de los OAAs tiene dos consecuencias:

- Generan gran cantidad de información de interacción, que puede **analizarse** para entender qué ha sucedido en el transcurso una sesión concreta, y para evaluar el conocimiento adquirido.
- Pueden beneficiarse de **comunicación bidireccional**, usando información proporcionada por el entorno (por ejemplo, un LMS) para presentar el contenido de forma adaptada al usuario; y devolviendo resultados al final de la actividad.
- Asumiendo acceso a **identidad persistente**, pueden asociar la información de interacción recogida a la identidad de un usuario dado, permitiendo al usuario retomar el juego en el punto en el que lo dejó, y al profesor ver su progreso sesión a sesión.

1.1.2.1 Game Learning Analytics

El término Game Learning Analytics se usa para referirse al uso de técnicas de análisis de aprendizaje (Learning Analytics, ó LA por sus siglas en inglés) a juegos educativos [7]. Debido a su alto grado de interactividad, los juegos generan un flujo constante de información sobre las acciones del jugador que es susceptible de ser capturado con distintos propósitos. Este flujo puede ser analizado para

- Evaluar el conocimiento adquirido
- Detectar puntos donde los jugadores se atascan, ya sea debidos a fallos en el diseño del OAA ó a dificultades con los conceptos necesarios. En ambos casos, detectar el problema es el primer paso para permitir solucionarlo, ya sea mejorando el OAA ó con explicaciones suplementarias por parte del profesor.
- Comparar el progreso de alumnos y grupos

Un sistema de GLA requiere los siguientes ingredientes [7]:

- Instrumentación: de los OAAs, de forma que envíen información de interacción a un servidor. Llamaremos a esta información “trazas”, ya que, en conjunto, permiten trazar las acciones que toma un usuario a lo largo del OAA.
- Colección y almacenaje de trazas: las trazas deben ser recolectadas y almacenadas para estar disponibles para la fase de análisis.
- Análisis: a medida que se recolectan, las trazas de bajo nivel (“usuario X tomó acción Y en punto Z”) son analizadas para por ejemplo, generar información de progreso o completación (“... y por tanto, el conocimiento de X sobre el concepto C se incrementó en 1: ha completado un 55% del tema T”)
- Presentación: muestra la información de los análisis a los profesores responsables; y, a cada usuario, su propia información. En inglés es frecuente el uso del término “Dashboard” (panel o cuadro de mando con gráficas), que adoptaremos para referirnos a este tipo de visualizaciones.

Estos pasos son comunes a cualquier sistema de analítica; su uso en GLA afecta, sobre todo, al mayor flujo de información (los juegos son altamente interactivos, y generan más trazas a para analizar que otros tipos de contenido) y añade complejidad a la hora de presentar la información, ya que hay más tipos de consumidores para las visualizaciones, cada uno con intereses específicos:

- Los profesores quieren entender qué sucede con sus alumnos

- Los alumnos deben tener acceso sólo a sus datos; y posiblemente, datos agregados de sus compañeros.
- Los desarrolladores de juegos estarán interesados en saber qué partes del juego funcionan mejor y peor a nivel de interacción; pero, por cuestiones de privacidad, no deben tener acceso a las identidades de los jugadores.

Además de los aspectos puramente de acceso por supuesto hay que tener en cuenta todas las implicaciones éticas y legales respecto a la captura de los datos y a su explotación posterior. Y en mas medida cuando se trabaja con menores de edad.

1.1.2.2 Estandarización en GLA: Caliper y xAPI/TinCan

El uso de analíticas de aprendizaje (LA) es tan prometedor que hay importantes esfuerzos de estandarización para permitir la interoperabilidad de datos recopilados de distintos orígenes de LA. Cabe destacar los estándares IMS Caliper y ADL xAPI. El estándar xAPI (previamente denominado TinCan) [8], basado en aseveraciones (“statements”) de la forma

<sujeito> <verbo> <objeto> [<contexto>]

usando un vocabulario controlado, y almacenadas en almacenes de registros de aprendizaje (“Learning Record Stores”, ó LRSs por sus siglas en inglés).

Por su parte, el estándar Caliper [9], aunque proporciona funcionalidad bastante similar a xAPI, presenta una mayor dificultad a la hora de su uso ya que no es completamente abierto y es mas complejo formalizar vocabularios específicos, tal y como resultaría deseable para describir interacción con OAAs. Como además goza de menor adopción que xAPI, nos centraremos en el uso de este último.

Existe soporte para LA con xAPI en los principales proveedores de LMSs¹, incluyendo Moodle². Al adoptar xAPI, los usuarios y proveedores de LMSs ó sistemas de LA ganan interoperabilidad y evitan depender de soluciones particulares, que podrían en un futuro ser discontinuadas (se pasa de un problema de integración punto a punto, que es dependiente de versiones concretas en los dos extremos, a un problema de compatibilidad con un estándar). Esto es especialmente importante para usuarios de LA, ya que no poder mover los datos de un sistema de LA a otro implicaría posible pérdida de datos, y una dependencia muy grande del proveedor de LA elegido.

El grupo e-ucm de la Universidad Complutense, en colaboración con los desarrolladores de xAPI, ha desarrollado un perfil específico de xAPI (estos perfiles también se denominan “recetas”) para cubrir las necesidades específicas de los juegos serios [10].

¹Adopción de xAPI en herramientas para eLearning: <http://tincanapi.com/adopters/>

² Plugin de Moodle para xAPI: https://moodle.org/plugins/mod_tincanlaunch

1.1.3 Requisitos de OAAs

Es posible desplegar OAAs sin hacer uso de GLA; pero esto desperdicia gran parte de su potencial impacto educativo. Esta sección explora las necesidades de integración para sacar el máximo provecho de contenidos OAAs en despliegues institucionales de LMS.

Consideremos el tipo de OAA más ambicioso posible: un juego serio que reciba información del LMS en el que se ejecuta, y a medida que un usuario va jugando, captura y mantiene un registro de las acciones del mismo, que es analizado en tiempo real para permitir al profesor de la clase entender cómo está transcurriendo la sesión, y, si lo desea, intervenir en la misma.

Desde el punto de vista de las secciones anteriores, esta integración requeriría:

- **Comunicación bidireccional:** el juego necesita tener información de los usuarios antes de comenzar; y para permitir acceso a analítica de aprendizaje, es necesario poder comunicar de alguna forma los resultados (y todos los demás datos sobre el transcurso de la sesión) al sistema.
- Si bien es posible solicitar datos mediante un formulario, la fiabilidad de los mismos y la experiencia de usuario será mucho mejor si éstos se obtienen sin necesidad de solicitarlos.
- Si bien es posible mantener un sistema de analítica completamente aislado del LMS, consideramos que en este caso la integración es insuficiente: este aislamiento implicaría que los datos de la analítica no están disponibles para el LMS (sino que tienen que ser importados de forma manual por el profesor responsable).
- **Persistencia integrada:** los usuarios no deberían perder su progreso e identidad en caso de desconexión temporal (modo semiconectado muy habitual en dispositivos móviles como tabletas); ya que, en general, no es de esperar que los usuarios jueguen todo el juego “de una sentada”; y al contrario, queremos permitir que entren y salgan cuantas veces necesiten hasta acabarlo.
- **Roles específicos:** para no requerir credenciales suplementarias de alumnos ni profesores (por no mencionar los demás roles que puede requerir un juego serio: desarrolladores, expertos en el dominio del juego, administradores, ...). El acceso a roles específicos permitiría a alumnos y profesores de cada grupo entrar en juegos desplegados como OAAs (y en el caso de profesores, administrarlos y acceder a la analítica); y preservaría la privacidad de cara a terceros.

1.1.4 Infraestructura de soporte a OAAs

Para sacar el máximo provecho a OAAs, es necesaria una infraestructura de soporte que no está incluida de forma estándar en los LMSs. En particular, los OAAs necesitan

- **Dashboards** para mostrar informes interactivos de los resultados de la analítica
- Gestión de **roles de soporte**, para poder restringir la presentación de datos de analítica a los profesores de los respectivos grupos (y, dentro de esos grupos, permitir a cada estudiante ver sólo sus propios resultados y un agregado de los vecinos). Es posible, y altamente deseable, incorporar roles adicionales, como por ejemplo el de “desarrollador de juegos”, que permitiría acceder a versiones anonimizadas de las analíticas de un juego particular a los desarrolladores del mismo, a fin de realizar mejoras cuando se detectan problemas.

- Un **LRS** (“Learning Record Store”, o sistema de registros de aprendizaje) que permita almacenar interacciones a largo plazo y re-analizarlas si fuera preciso, así como compartir esta información con otros sistemas de LA ó exportarla.
- **Análisis** en tiempo real o en diferido para poblar el *dashboard* y mostrar a sus usuarios información más útil y aplicable que la proporcionada por las trazas almacenadas en el LRS.
- **Interfaz de configuración** que permita a los desarrolladores de juegos y profesores configurar los análisis y la forma en que se mostrarán sus resultados. Si bien una mayoría de profesores preferirán no entrar en esta interfaz y usar sólo los análisis por defecto, es posible sacar mucho más rendimiento del sistema de analítica si al menos los desarrolladores de juegos invierten tiempo en asociar acciones del juego con conocimientos adquiridos.

1.2 Integración de OAAs en LMSs

Teniendo en cuenta los requisitos y opciones expuestos en la sección anterior, esta sección presenta las recomendaciones del proyecto a la hora de integrar OAAs en instalaciones institucionales de LMSs. El primer subapartado describe la arquitectura elegida para proporcionar la infraestructura de soporte a LMSs; y el segundo subapartado describe cómo se integra esta infraestructura con una instalación institucional de LMS.

1.2.1 Arquitectura de analítica RAGE

El proyecto EU H2020 RAGE (“Realising an Applied Gaming Eco-system”, traducible como “implementando un ecosistema para juegos aplicados”), en el que participan los autores de este documento y el grupo e-ucm, busca crear un ecosistema de componentes que facilite el desarrollo de juegos serios; que, aplicados a la educación, y como se argumenta en secciones anteriores, pueden verse como OAAs prototípicos.

En particular, varios de los componentes de RAGE están pensados para, en combinación, proporcionar toda la infraestructura de soporte que requiere un juego serio, tal y como se muestra en la Figura 1.

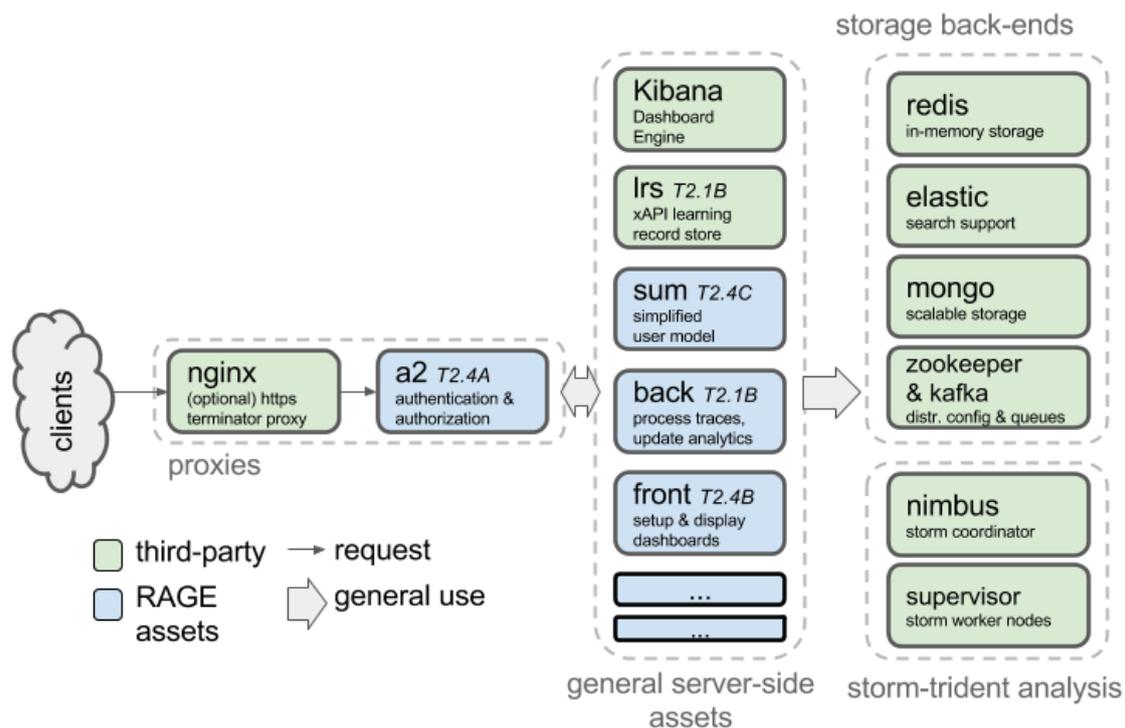


Figura 1: Infraestructura de soporte proporcionada por los componentes de analítica de RAGE.

•

La Figura 1 muestra la arquitectura de servidor que da soporte a un juego serio en RAGE. Las cajas en azul han sido desarrolladas para el proyecto RAGE; mientras que aquellas en verde han sido adaptadas de software libre, y proporcionan servicios estándar tales como servidor web (Nginx) almacenamiento (Redis, Elasticsearch, Mongo) o infraestructura de análisis (Apache Storm). Teniendo en cuenta el apartado 1.1.4 sobre infraestructura de soporte, cabe destacar:

- Para proveer tanto *dashboards* como una interfaz de **configuración para los análisis**, el componente “front” (Tarea 2.4B de RAGE), descrito en [11]; que delega en Kibana³ para proporcionar las visualizaciones en tiempo semi-real.
- Para realizar **análisis en tiempo real**, desplegamos los análisis en una instalación Apache Storm⁴. Del despliegue y la configuración y comunicación de los análisis se encarga el módulo “back” (Tarea 2.1B, descrita en [12]).
- Para el apartado de **gestión de identidad**, usamos el módulo “a2” (Tarea 2.4A, también descrito en [12]).

³ Kibana, un *dashboard* interactivo para web: <https://www.elastic.co/products/kibana>

⁴ Apache Storm, un sistema de análisis altamente escalable y resistente a interrupciones: <http://storm.apache.org/>

- Como **LRS**, hemos elegido el software abierto preexistente Apereo OpenLRS⁵ por su facilidad de integración.

Teniendo en cuenta la complejidad técnica de despliegue debido al alto número de servicios empleados por la infraestructura RAGE, ésta hace uso del sistema de contenedores Docker, que permite realizar despliegues de todos los servicios como si todos ellos fuesen un único servicio integrado, y define un único punto de acceso (ver flecha “clientes” en la Figura 1). Es posible descargar la última versión de la infraestructura de análisis⁶, e incluso participar en el desarrollo de la misma, gracias a la naturaleza de código abierto del proyecto.

1.2.2 Acceso al sistema institucional de gestión de identidad

El módulo “a2” de RAGE proporciona control de autenticación y autorización para todos los módulos de RAGE (y de hecho su nombre proviene de acortar las iniciales de Authentication & Authorization).

Desde diciembre de 2016, a2 soporta tanto BLTI como SAML⁷, gracias a una arquitectura de complementos (plugins) de autenticación. Antes de iniciar una instalación de la plataforma de analítica, el administrador debe especificar qué tipo de integración desea realizar entre la gestión de identidad institucional y la plataforma. Para ello, el administrador debe activar y/o configurar una o más de las plantillas proporcionadas, a elegir entre:

- El plugin LTI, que permite realizar autenticación y autorización (=obtención de roles) con una instalación LTI institucional.
- El plugin SAML, que realiza sólo autenticación y requiere la existencia de infraestructura SAML
- Un plugin desarrollado desde cero, pero partiendo de cualquiera de las estrategias de gestión de identidad proporcionadas por la librería estándar passport.js (más de 300 en el momento de escribir este documento⁸).

1.3 Integración de OAAs en LMSs

Una vez instalada la plataforma de soporte a OAAs, es posible integrar éstos en un LMS de primera línea sin perder ninguna funcionalidad.

⁵ Apereo OpenLRS, con soporte para xAPI: <https://www.apereo.org/projects/openlrs>

⁶ Infraestructura de análisis de RAGE: <https://github.com/e-ucm/rage-analytics>

⁷ Soporte para BLTI y SAML en a2: <https://github.com/e-ucm/rage-analytics/wiki/Login-Plugins>

⁸ Librería de gestión de identidad Passport.js: <http://passportjs.org/docs>

1.3.1 Integración: cambios en el OAA

Recordando la sección 1.1.1.1 sobre tipos de despliegue, el OAA puede consistir bien en un módulo web pensado para ejecutarse en el navegador del usuario mientras envía información de analítica a su infraestructura de soporte; o en una aplicación nativa que se ejecuta en el dispositivo del usuario, enviando la misma información. En ambos casos, el OAA no requiere cambio alguno para ser integrado en un LMS.

Para comunicarse con la infraestructura de soporte RAGE, un OAA requiere sólo dos parámetros:

- La URL y puerto donde está escuchando el servidor de recolección de trazas (llamado “back”, en la Figura 1).
- El código identificativo del juego al que se está jugando.

Asumiendo una correcta integración con el sistema de identidad institucional, la infraestructura enriquecerá automáticamente las trazas enviadas por el juego con la información de identidad y roles del usuario-jugador.

Para facilitar la creación y envío de trazas SG-xAPI correctas (ver sección 1.1.2.2), y también dentro del marco del proyecto RAGE, el grupo del autor ha desarrollado múltiples librerías de código que facilitan esta comunicación, pensadas para integrarse directamente en el código del OAA. Están disponibles online como código abierto⁹, y descritas en [13].

1.3.2 Integración: cambios en el LMS

No hay que efectuar cambio alguno en el LMS, ya que toda la infraestructura de soporte es externa al mismo.

En cualquier momento, el profesor podrá consultar el progreso de sus alumnos a través del cuadro de mando (dashboard); y de nuevo, asumiendo que la integración sea correcta, no necesitará suministrar información adicional al sistema, salvo para elegir la clase, de entre aquellas en las que es profesor, para la cual quiere consultar información.

1.3.3 Integración incompleta

De haberse optado por una integración incompleta con el sistema de gestión de identidad, tanto los alumnos jugadores como los profesores tendrán que gestionar credenciales de forma separada: unas para usar su LMS habitual, y otras para interactuar con los OAAs y su infraestructura de soporte. Es también posible, para los alumnos, prescindir de autenticación alguna (mediante sesiones anónimas), aunque en este caso no se podría usar el juego serio para

⁹ Librerías cliente para integración en el propio OAA: <https://github.com/e-ucm/unity-tracker> (Unity), <https://github.com/e-ucm/ClientSideTrackerAsset> (C#), <https://github.com/e-ucm/rage-analytics-tracker> (Java)

propósitos de evaluación, ya que no dispondría de la correspondencia entre identidades anónimas y reales.

1.4 Conclusiones

Este documento ha descrito una metodología de integración de Objetos de Aprendizaje Avanzados (OAAs) en sistemas de e-Learning, centrándose en Juegos Serios. El documento va más allá de la integración con un LMS, y describe una estrategia que abarca también la infraestructura institucional en la que se encuentra este LMS.

Los puntos principales de esta metodología son:

- Integración con el sistema de gestión de identidad institucional, ya sea éste interno o externo al LMS, para proporcionar seguridad y facilitar el uso de los servicios de soporte de los OAAs, incluyendo el LRS y, sobre todo, los “dashboards” (visualizaciones interactivas) de analítica. Esta integración, hace uso de los estándares BLTI y SAML, soportados por un amplio número de instalaciones.
- Uso del perfil de aplicación o receta SG-xAPI como formato de almacenamiento de los datos enviados por el OAA al sistema. (propuesta y parcialmente desarrollada por el grupo e-ucm de la Universidad Complutense en colaboración con ADL) para proporcionar resistencia a cambios tecnológicos: aunque varíen los sistemas de LMS, LRS y LA utilizados, los datos recopilados seguirán siendo interpretables.
- Uso de la infraestructura RAGE para proporcionar todos los componentes necesarios para realizar esta integración. Esta infraestructura, desarrollada por el grupo de investigación de los autores, está disponible de forma abierta y gratuita, y puede ser libremente instalada y mejorada por cualquier particular ó institución.

Bibliografía

- [1] IMS Global Consortium, “IMS Learning Tools Interoperability, Version 1.1 Final Specification,” 2012. [Online]. Available: <http://www.imsglobal.org/lti/>.
- [2] IMS global learning Consortium, “IMS common cartridge specification,” 2009. [Online]. Available: <http://www.imsglobal.org/cc/index.html>.
- [3] ADL, “Advanced Distributed Learning Sharable Content Object Reference Model (ADL-SCORM) 2004 v4,” 2009. [Online]. Available: <http://www.adlnet.gov/scorm/>.
- [4] J. Hughes and E. Maler, “Security assertion markup language (saml) v2. 0 technical overview,” *Work. Draft sstc-saml-tech-overview-2.0-draft ...*, 2005.
- [5] G. McFall and L. Neumann, “IMS learning tools interoperability basic LTI implementation guide v1.0 Final,” 2010. [Online]. Available: <https://www.imsglobal.org/specs/ltiv1p0/implementation-guide>.
- [6] P. Moreno-Ger, J. Torrente, J. Bustamante, C. Fernández-Galaz, B. Fernández-Manjón, and M. D. Comas-Rengifo, “Application of a low-cost web-based simulation to improve students’ practical skills in medical education,” *Int. J. Med. Inform.*, vol. 79, no. 6, pp. 459–67, Jun. 2010.
- [7] M. Freire, Á. Serrano-Laguna, B. M. Iglesias, I. Martínez-Ortiz, P. Moreno-Ger, and B. Fernández-Manjón, “Game Learning Analytics: Learning Analytics for Serious Games,” in *Learning, Design, and Technology*, Cham: Springer International Publishing, 2016, pp. 1–29.
- [8] ADL, “Experience API,” 2014. [Online]. Available: <https://github.com/adlnet/xAPI-Spec/blob/master/xAPI.md>. [Accessed: 04-Feb-2016].
- [9] IMS Global Learning Consortium, “Caliper Analytics,” 2015. [Online]. Available: <https://www.imsglobal.org/caliper/caliperv1p0/ims-caliper-analytics-implementation-guide>.
- [10] Á. Serrano-Laguna, I. Martínez-Ortiz, J. Haag, D. Regan, A. Johnson, and B. Fernández-Manjón, “Applying standards to systematize learning analytics in serious games,” *Comput. Stand. Interfaces*, vol. 50, pp. 116–123, 2017.
- [11] M. Freire, I. M. Ortiz, and B. F. Manjon, “D2. 5-First bundle of dashboard components,” 2016. [Online]. Available: <http://dspace.learningnetworks.org/handle/1820/7020>. [Accessed: 24-Jan-2017].
- [12] M. Freire, A. Nussbaumer, M. Dascalu, and P. Hollins, “D2. 1-First bundle of server-side components,” 2016. [Online]. Available: <http://dspace.ou.nl/handle/1820/6919>. [Accessed: 24-Jan-2017].

- [13] M. Freire, K. Bahreini, D. Vassileva, B. Bontchev, and J. Dias, "D2. 3 First bundle of client-side components," 2016. [Online]. Available: <http://dspace.ou.nl/handle/1820/6909>. [Accessed: 24-Jan-2017].