Application for 2014 University of California Larry L. Sautter Award for Innovation in Information Technology

Project Title: Core Services
Submitted: Jay Ballinger
IT Services, University of California, Davis
Email: jayballinger@ucdavis.edu
Office: 530.554.1373

Diana Cox
IT Services, University of California, Davis
Email: dicox@ucdavis.edu
Office: 530.752.6687

Project Leaders
• Cheryl Lloyd, Chief Risk Officer, Risk Services, UCOP
• Jay Ballinger, UC Systems Architect, IT Services
• Safa Hussain, Director, IT Services

Project Team Members
The Core Services project’s success has been dependent on the support of the Researchers, the Environmental Health & Safety (EH&S), and IT Services communities across all of the UC campuses. As part of the development process at IT Services, numerous staff and faculty provided guidance, feedback, expertise, and insight contributing to the success of the project.

The team members listed below made up the core application development team:
• Andie Cheung, Developer
• Moira Heilmann, Project Coordinator
• Mira Kaloper, Developer
• Serguei Mysko, Developer
• Stefan Tomic, Developer
• Nishant Vincent, Contractor

Summary
Core Services is a middleware solution providing data management and publishing of essential cross-application information including researchers, researcher staff, research students, Environmental Health and Safety staff, location of each room and storage area, and the catalog of chemicals available at each of our campuses. By centrally managing and publishing this core set of data we enable the various applications and business intelligence reports to focus on their unique value while sharing this common set of data.

Project Description
How It Works
When software applications are written without regards to data integration or sharing we refer to that application as being written in a silo. Silo applications often result in the duplication of essential data and that duplication leads to data consistency and reliability problems. For example, if five different applications requested a Principal Investigator to list each room they use and list which chemicals are used in each room, two things happen. First, the PI becomes fatigued
and annoyed at reentering the same information multiple times. Second, because of the fatigue that PI may not be
diligent in keeping each application updated. When the applications have conflicting data sets, reporting and
comparisons become very difficult and can lead to inaccurate results.

Core Services was written to alleviate the problem of redundant information. By using the concept of “enter once, use
often,” data such as the list of rooms and the list of chemicals can be entered and managed once but used multiple
times by each application which subscribes to the centralized data.

Figure 1 illustrates the growing suite of Enterprise Risk Management and Environmental Health & Safety software
applications which subscribe to services published by Core Services. These applications are used by all of the UC
campuses and represent a significant effort in centralizing the essential data. Instead of each application being its own
silo and duplicating the gathering, storing, and management of the same data elements, these shared elements are
available, in real time, to all of the software systems.

By allowing each Core Service client software application to focus on its own unique functionality we realize efficiencies
of effort by solving the central data issue once, versus having to replicate the effort for each campus and for each
application. Imagine that each of the client applications represented in Figure 1 needed to gather building and room
information for each campus. In a non-centralized service environment that effort would equate to 11 project teams
needing to contact the 10 campuses on their own. The benefit of Core Services is that the team was able to engage each
campus just once and then republish that information to all of the subscribing client applications.

In addition to engaging each campus once, Core Services has established automated update routines that will keep the
centrally-published data updated and as accurate as possible. The client applications will always receive the most recent
data on each request to the Core Services features.
Figure 2 illustrates the data flow essential to Core Services. Each UC Campus is the source for the essential people and location information. The people information represents identifiers and attributes about each user of the suite of client applications. The location information provides specific identifiers for every building, floor, and room on every campus. By combining the source identifiers for this data, the campuses and the client applications enjoy the benefits of consistent identifiers. Instead of each application applying its own label to a room, for example, each application can relate its room information to every other application by virtue of using the same source campus identifier.

Core Services also centralizes and publishes the chemical attributes on over 60,000 chemicals obtained from our industry partners. Chemical information such as hazards, safe handling procedures, and required protective equipment has been historically difficult to obtain. By partnering with the suppliers and manufacturers we are able to centralize those attributes and make them available to the client applications.

The data flow represented in Figure 2 also demonstrates the phenomenon of ever growing and improving data. As the campuses provide the user information, and as that user information is combined with the additional attributes in Core Services, the better and more comprehensive the data becomes as the campus’ users continue to use the applications. Each user and each application provides the opportunity for the data to be used, reviewed, corrected, and augmented. In this regard each client application benefits from the efforts of the other client applications.

**Core Services URLs**

Core Services is a middleware solution – meaning that it does not have its own user interface. Rather, Core Services publishes the data used by the suite of client applications. In one sense, the user interface for Core Services can be seen in each of the client applications depicted in Figure 1 – especially when viewing people, laboratory definitions, researcher roles, EH&S roles, and room locations on campus.

**Project Benefits**

This project introduces many benefits which plague smaller projects or projects that cannot effectively share information.

Campus System of Record – Unlike many applications which create their own surrogate keys or apply local identifiers to key data elements, Core Services uses the campus’ system of record and republishes the campus’ identifiers for people
and for locations. Core Services is careful to help scope the information by campus – preventing apparent duplicate keys – so that each subscribing client application can use the identifiers with confidence that they are unique, and that they are relatable and usable by the campus and by related applications. For example, if applications A, B, and C all subscribe to Core Services and if they all need to understand campus room information, a Business Intelligence report (another feature of Core Services) can be generated relating the specifics of how a particular room is used in those applications. This can only work if applications A, B, and C use the same identifiers for the same rooms – which is what Core Services provides.

Efficiencies of Scale – Core Services saves time for every project team implementing a new subscribing client application. Rather than spend time engaging a campus to glean information that was used in another application, new client applications can simply subscribe to Core Services and immediately enjoy the benefits of a comprehensive data set of campus people, people roles, and location information for their intended campus. With Core Services’ automated update features, the client applications can also trust that the Core Services data is the most recent and accurate possible. This removes a heavy burden for each team and allows them to focus on the development efforts of their new application.

Data Sharing – The current suite of ERM and EH&S software applications benefit the researchers on our campuses. In recognition of their time and efforts we strive to make the applications as intuitive and as easy to use as possible. We can accomplish this by the “enter once, use often” technique of asking for data just once. Asking a Principal Investigator to define which rooms comprise their lab once, as an example, we can use that data set of PI and rooms and share that information amongst the related client applications which need that same data. And, when the PI modifies their data set to include additional Lab Workers or rooms, that data is immediately shared with the related client applications removing the need to reenter that information more than once.

Technology Neutral – Core Services is implemented using SOAP web services and web-based applications. This allows client applications written in different technologies to subscribe to and use the data available via the web services. The common protocols and representations of http, https, xml, soap, and json allows the flexibility for the client applications to use their own tool set when integrating the Core Services features within their applications.

Campus Collaboration – One of the greatest side benefits of this project is the building of relationships with the people on each campus. This project represents a trail-blazing effort to consolidate people and location information from every campus with the expressed purpose of allowing those campus’ users to use that same information. By having the Core Services development team perform the campus engagement to establish the various data feeds, we have been able to meet friends and colleagues from the all of the campuses. These relationships pay dividends when resolving issues and looking for even more opportunities to share and collaborate on common projects.

Technology Innovation – The suite of ERM and EH&S applications have been recognized for their innovations, usefulness, and performance. Core Services shares in their successes by providing the underlying common data elements in support of the applications’ features and functionality. This innovation of relying on Core Services is unprecedented at this level of supporting applications written for and used by every campus and by such a large user population on each campus. Those client applications would not be as successful without the benefits of Core Services.

Ease of Integration – Because Core Services is published using standard SOAP messaging, a client library is available for use by the subscribing client applications. The client library is merely a convenience for the project teams which makes the integration with Core Services trivially easy. For example, the function of listing the buildings on a campus is as simple as using the client library and writing just a couple of lines of code. The XML translation and the network communication is already taken care of for the application team saving them the complexity of having to write it
themselves from scratch.

Technology Used

The magic of Core Services is not rooted in which technology used, but in how the technology is used. These elements are commonly available and used in other application suites, but the unique combination of these items used to create Core Services is what makes it special. The design of Core Services was envisioned before the technology was chosen, but these technology pieces certainly helped to make the vision a reality

- Java
- Kuali Rice
- Custom web services written in JAX-WS
- Java XML marshalling using JAX-B
- Apache Tomcat
- Linux application servers
- Mule ESB for proxy services allowing for absolute URL control
- Various monitoring tools to test the service availability
- Various logging and log analysis tools
- SoapUI for raw web service testing
- Various flavors of a Java integrated development environments (IDE)
- wsimport to help generate client library
- JUnit for unit and integration testing
- Jenkins performs continuous building/testing
- JIRA for Agile board management
- JIRA for issue/task management across the multiple JIRA projects comprising the facets of Core Services
- Confluence for team collaboration and documentation
- Bitbucket for source control

The technology stack is common to many Java-based projects, but the employment of these tools lends itself well to the development of the solution. For example, the code for Core Services is divided into six different projects representing the following facets: core web service, core web service client, wrapper web service, wrapper web service client, profile management, administration management. Each of these code repositories is continually monitored for changes and automatically compiled and tested when any code change is detected. This continuous integration and testing helps to build reliability in the services by having tests performed at a level much higher than anticipated in the production environment. In other words, our select of development tools and production hosting tools are equally important in the success of Core Services.

Project Timeline

The Core Services project team embraced the Agile methodology for development. While the overall picture and goal for the set of features needed in Core Services was known, the division of duties and the order in which the service was developed allowed for short iterations and frequent testing. This agile process afforded the team the ability to release small bits of functionality at a steady pace. The client application teams were able to consume these tidbits of functionality as they developed their applications and test them for their acceptance. Having the constant feedback of the other teams helped to result in a high quality product.

Core Services has been in continual development for more than a year. As each Agile sprint is accomplished, the velocity of the team increases by virtue of the work performed in previous sprints. In other words, with the underlying data management already completed, the team is able to focus on advanced data relationships, adaptive caching strategies, and innovative threading techniques which keeps Core Services responsive and ahead of the curve on requested features and functionality.