Architecting Intelligent Content Management Software as a Service for Cuba's enterprises

Adrian Hernández-Méndez[*], Nastaran Matthes[+], and Florian Matthes[#]

[*] Department of Biomedical Engineering, Faculty of Electrical Engineering, Instituto Superior Politécnico José A Echeverría, Marianao, 19390 La Habana. Cuba

[+] InfoAsset AG. 80538 Munich. Germany

[#] Software Engineering for Business Information Systems, Informatics Faculty, Technische Universität München, 85748 Munich. Germany

Abstract

The complexities to manage the rising amount of content and the software development processes in Cuban enterprises demand new solutions to improve the Cuban economic model. We present a software architecture to manage *intelligent content* with an integrated software platform using cloud technologies. The cloud's service is Intelligent Content Management Software as a Service (ICMSaaS). In this paper, we argue that the development process and the facilities of cloud technologies make ICMSaaS a promising software development solution for Cuban enterprises.

Keywords

ICMSaaS, SaaS, Intelligent Content, Enterprise Software, Economies of Scale

Introduction

A modern enterprise has to deal with large amounts of content generated from all of its processes. This content can be, but is not limited to, documents, media and emails, as well as metadata. Finding the right content in faster ways guarantees an accelerated decision-making, so content management processes become one of the great challenges of companies in XXI century. A recent approach to manage enterprise content is to enrich it with metadata and context which is called *intelligent content* by (Rockley & Cooper, 2012).

In Cuba there are several causes of failure to obtain intelligent content at the highest levels of enterprise's management even though the improvement in the Cuban economic model demands for solutions in such cases. One of the most important causes is related to the heterogeneity of existing software solutions deployed in Cuban enterprises. The publication of Decree-Law No. 281 "INFORMATION SYSTEMS OF THE GOVERNMENT" caused enterprises in the 90s to develop their own solutions individually. Figure 1 shows the growing of software solutions presented in the International Workshop

ICT in Organizations Management of the main Informatics Conference in Cuba during the years 2009, 2011, and 2013.

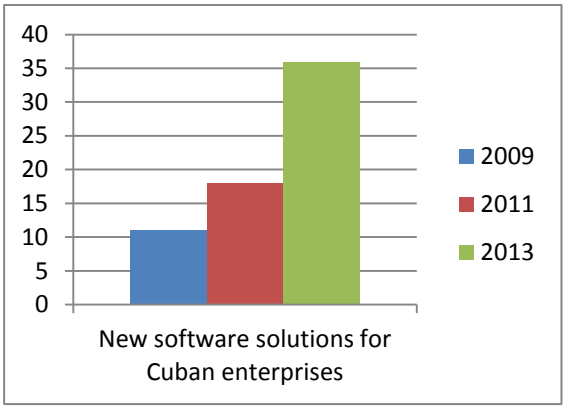

Figure 1. Papers presentenced at the International Workshop ICT in Organizations Management.

Another cause is related to the development process. It was centered in the developer side and what the developer team knows and understands of the business process. When the process is centered in the developer side the final solution is less likely to adapt to any changes related to the developer side (e.g. when a software's developer leaves the company) or to the business process.

However, these limitations have grown in parallel with the development of cloud technologies that provide a centralized technological solution, Intelligent Content Management Software as a Service (ICMSaaS), which allows maximizing the efficiency of shared resources. Each subordinate company manages its content when they need it by using a common architecture that facilitates the management of intelligent content. This paper proposes an architecture for ICMSaaS that guarantees robustness in the development of enterprise software solutions for Cuban enterprises.

Intelligent Content

Intelligent content (IC) has two main properties: It is structurally rich and semantically categorized. These properties are sufficient for making IC automatically discoverable, reusable, reconfigurable, and adaptable (Rockley & Cooper, 2012).

Our model of IC has three components: content types, relations between content types and relations between similar instances.

A content type (CT) is the specialization of an IC. The properties which make a CT structurally rich are its fields and its relations to other content types. Furthermore, the content types are semantically categorized by tags into hierarchies.

Cloud Technologies

The cloud computing model provides on-demand services over shared computer resources (Mell & Grance, 2011). There are three basic cloud service models that are organized from the level of the technology required by the customer. The first is closer to the hardware, the customer manages the operating system and the service is known as Infrastructure as a Service (IaaS). In the second case the customer gets the operating system, the programming environment and tools for developing application, the service's name is Platform as a Service (PaaS).

Finally, the customers get the applications they need (on-demand) without the need to install requirements of the platform or operating system. That service is called Software as a Service (SaaS).

The Intelligent Content Management Software as a Service (ICMSaaS) is a special case of SaaS, where the intelligent content is the main element managed on-demand by the customers.

A Model-based ICMSaaS Architecture

Figure 2 visualizes the proposed architecture of ICMSaaS. The ICMSaaS architecture was developed on top of the SaaS architecture proposed by Intel (Spence, Devoys & Chadal, 2009). It assumes that in the future software solutions for Cuban enterprises will be built on an integrated software platform, which shields software engineers, technical domain experts, business domain experts, and end users from details and changes in the infrastructure layer (databases, file systems, search engines and other middleware) and the implementation of the generic functions of information systems (persistent data management, querying, event handling, integrity checking, message encoding, marshaling, user interface management).

A reference architecture for each business solution identifies the models and modeling languages to be used and constrains their mutual dependencies so that the actual architecture of an information system built in the environment conforms to the reference architecture of the integration platform.

The data (and metadata) model is the foundation for all other models. It is defined using a given data modeling language with concepts like content types, content templates, prototypes, attributes, relationships, roles or inheritance.

The software model builds on the data model and specifies the business logic of the system using a domain-specific language which provides concepts like queries, functions, procedures, rules, or constraints.

The process model depends on the other two models and specifies the interaction between the information system and other systems and users using a domain-specific language with concepts like events, messages, processes, actors or ports.

Finally, the interaction model specifies the details of the interaction between users using a domain-specific language with concepts like viewpoints, views, controllers, or view states.

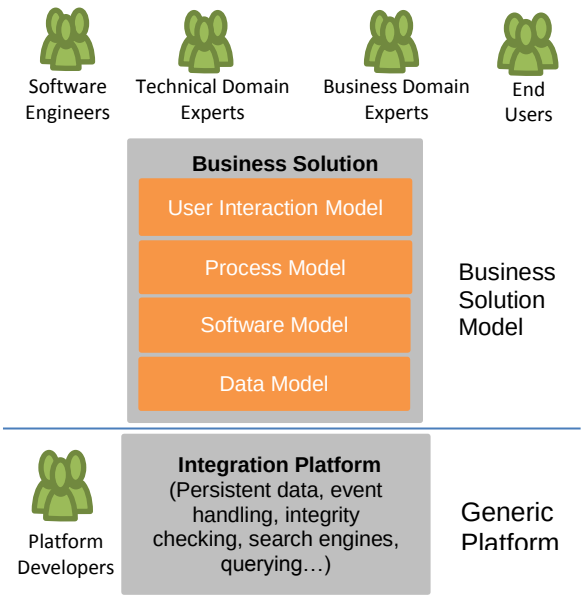

Figure 2. Model-based implementation of a ICMSaaS business solution

Enterprise Scenario Examples

To illustrate the implementation of our model of intelligent content, we use a simplification of a Customer Relation Management (CRM) system and a Task Management (TM) System.

The process starts with the definition of the data model: the content types, fields, and theirs relations.

Content types and fields:

Content Type	Fields	Field Type	Constraint
Task	Title	Text	Exactly one value
	Status	Enumeration	Exactly one Value
	Due Date	Date	At most one value
	Creator	Reference to User	Exactly one value
	Reminder	Enumeration	At most one value
	Assigned To	Reference to User	Any number of values
	Related To	Reference to Company, Opportunity, or Contact	At most one value
	Description	Text	Any number of value
Company	Company Name	Text	Exactly one value
	Location	Text	At most one value
	Address	Text	Any number of value
	Phone	Text Pattern	Any number of value
	Fax	Text Pattern	Any number of value
	E-Mail	Text Pattern	Any number of value
	Web	Text Pattern	At most one value
	Category	Enumeration	Any number of value
	Status	Enumeration	At most one value
Contact	First Name	Text	Exactly one value
	Last Name	Text	Exactly one value
	Company	Reference To Company	At least one value
	Position	Text	Any number of value
	Address	Text	Any number of Value
	Phone	Text Pattern	Any number of value
	Fax	Text Pattern	Any number of value
	E-mail	Text Pattern	Any number of value
	Category	Enumeration	Any number of value
Opportunity	Project Name	Text	At most one value
	Client Company	Reference To Company	At most one value
	Main Contact	Reference To Contact	At most one value
	Volume	Number	Exactly one value
	Start	Date	At most one value

	Finish	Date	At most one value
	Invoice Date	Date	At most one value
	Status	Enumeration	At most one value
	Feeling	Enumeration	At most one value

The automatic analysis of the content types' relations makes them discoverable.

Relations between content types:

Content type	Field	Related Content Type
Task	Related To	Company
		Opportunity
		Contact
Company	Parent Company	Company
Contact	Company	Company
	Assistant	Contact
Opportunity	Client Company	Company
	Main Contact	Contact

A default user interface supports the definition of metadata with associated create, edit and delete actions.

An interactive data view that can be deployed in a widget container allows the user to interact in a browser with the data. Figure 3 displays a detail page for an opportunity that includes all relations to tasks and other opportunities:

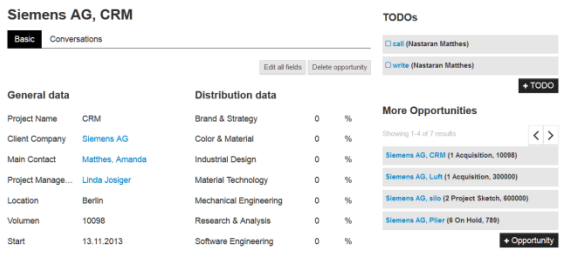

Figure 3. Opportunity detail page

Software engineers and technical domain experts are able to create and customize the pages using tools like changing the order of data fields on page or definition a new template using typed based template engine as defined by (Reschenhofer 2013).

Interfaces can also contain views which are designed by technical domain experts. Figure 4 shows a list of opportunities with status open:

Figure 4. List of open opportunities

Default tables allows end users to customize the views according to their needs (columns, filters, sorting).

Using a domain-specific language like MxL (Model-based expression Language, Reschenhofer 2013) software engineers or technical domain experts define summary reports and queries as well as graphical representation like charts. Figure 5 shows the definition of a sample CRM dashboard with MxL.

Figure 5. Simple CRM Dashboard with MxL

Finally ad-hoc workflows can be designed using reusable built-in component like

such as “Action”. Using precedence relations among actions and their assignment to data types, software engineers can define control flows for (e.g. for the opportunity follow-up process) based on a case handling paradigm (Motahari-Nezahd 2012).

Example of a Commercial Cloud Application Platform

Tricia is a Java-based enterprise 2.0 Wiki system initially developed by the SEBIS chair of the Technical University of Munich and commercialized in 2012 by infoAsset AG located in Munich. Tricia can be used as a platform to implement dynamic data-intensive enterprise web applications and social software solutions as a service in the cloud (Büchner, Matthes & Neubert, 2010).

The Tricia platform enables the secondary design (Germonprez et al. 2011) of business applications and their dynamic integration together. Secondary designers could be business domain experts who shape the business application in the context of their use without being aware of the platform.

The platform provides a set of enterprise 2.0 functions like notification, recommendation, tagging, version history, wiki, blog which can be customized by secondary designers and end users.

Tricia-based business applications are already available on the German Telekom Business Marketplace as a cloud service (SaaS). Tricia-PIM is a collaborative multi project management tool which allows users to manage all project information like tasks, meetings and files in one place. Further business applications like Tricia CRM, Tricia Product Life Cycle Management and Tricia Scrum which are developed based on the same platform are available as service on the Business Marketplace as well.

The customer can order these applications through the web and immediately start using without any of the traditional software installation challenges. They can use different applications like Tricia-CRM and Tricia-PIM on the same account which allows them to orchestrate their enterprise business processes on one integrated platform and have a central integrated access point to their business content.

Using a central technology for various business applications maximizes the efficiency of shared resources and minimizing the information retrieval time at the same time.

Figure 6 shows the integration of Tricia business applications based on the common Tricia platform.

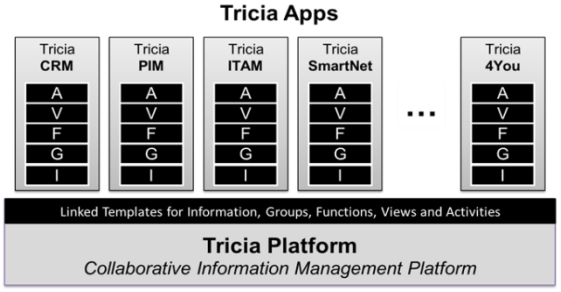

Figure 6. Changing and adding new apps to orchestrate enterprise business processes

Conclusions

The development a of integrated software platform with a clear separation of the roles of software engineers, technical domain experts, business domain experts, and end users from the generic functions of information systems and the facilities of cloud services architectures make the ICMSaaS a robust development software solution for Cuban enterprises.

However providing a secure and scalable cloud platform requires new tools and technologies to meet the challenges of creating and running integrated business solutions in the cloud.

References

Rockley, A. & Cooper, C. (2012). Managing enterprise content : a unified content strategy. Berkeley, CA: New Riders.

Mell,P. & Grance,T.(2011). The NIST Definition of Cloud Computing. NIST Special Publication 80-145. Gaithersburg, MD: National Institute of Standards and Technology.

Spence C., Devoys J. & Chadal, S (2009). Architecting Software as a Service for the Enterprise. IT@Intel White Paper. Intel Information Technology.

Büchner T., Matthes F, & Neubert C. (2010) Data Model Driven Implementation of Web Cooperation Systems with Tricia. 3rd International conference on Objects and Databases.

Reschenhofer T. (2013) Design and prototypical implementation of a model-based structure for the definition of calculation of Enterprise Architecture key Performance Indicators, Master Thesis

Motahari-Nezhad, H. R., Bartolini, c., Graupner, S. and Spence, S. (2012) Adapive case management in the social enterprise. In: Service-Oriented Computing (pp. 550.557). Springer Berlin Heidelberg

Germonprez M, Hovorka D, Collopy F (2007) A theory of tailorable technology design. J. AIS, 8(6): 351-367