eBook

Adding Social Media to an Enterprise Application:

An EA/UML Approach

Karl Schulmeisters

Introduction

This whitepaper differs from Michelle van den Berg's ArchiMate, BPN and UML: an approach to harmonizing the notations in that I will focus on a practical application of using both Enterprise Architecture Principles integrated with UML Modeling to look at best practices for adding a new capability to an existing Enterprise Application.

Specifically we will look at something that is highly relevant to today's business: enhancing an existing Enterprise Application with a Cloud Based Social Media presence.

Inside-Out or Outside-In?

In prior whitepapers I have discussed the pros and cons of an Inside-Out1 vs. Outside-In2 approach to deploying an Enterprise Architecture. The former refers to starting from a technical architecture and working towards business goals and the latter to starting with business goals and moving towards an implementation or deployment.

In our example, we have a clearly defined business goal: to enhance or modernize an existing enterprise application by adding a cloud based social media capability to the application. This is a concrete and verifiable business goal and thus lends itself to an Outside In strategy.

Outside In: The Outside In approach works best when the business goals are clearer than the technology underpinnings or when the business sponsorship is stronger than the technology sponsorship.

Inside Out: In an Inside Out approach the business goals are often less than concrete and derived from the general business strategy at the corporate or divisional level thus lacking the concreteness of verifiable goals.

Gather the Business Requirements

The first step in using an Enterprise Architecture approach to adding new capabilities to an existing Enterprise Application is to gather the specific Business Requirements, Goals and Roadmaps for the existing application. TOGAF[3] (The Open Group Architecture Forum) recommends using the "SMART" approach:

S **Specific** as to what needs to be done in the business

M **Measurable** need clear metrics for success

A **Actionable** clear identification of the problem and the components of the solution

R **Realistic** the solution needs to be realistic

T **Time-bound** there needs to be a clear understanding of the time frame required.

Typically on the business side, this information is stored in various Office compatible documents (Word files, Excel spreadsheets, emails). These can be imported directly into iServer and tagged with the corresponding metadata. The goal here is to clearly articulate

1. **The Problem:** Which in this case could be any one of or all of
    a. We have no Social Media presence and thus are missing branding opportunities

    b. Our customers spend most of their time on Facebook™ and we are seeing less and less usage of our website

    c. We are missing the opportunity to identify new prospects via Social Media

2. **The Environment:** Which aspects of the existing Application need to be changed, what are our resource and time constraints

3. **What are our specific business objectives:** This is somewhat tied to #1 but really refines it by identifying what objectives are going to be met in addressing the problem. For the problems listed above this might look like:

    a. *Objective:* Have a Social Media presence that allows our existing automated brand messaging to be distributed via social media

    b. *Objective:* We want to provide a seamless experience for our customers on Facebook™

    c. *Objective:* We want to be able to use Social Media to Market our products

4. **Human Actors:** This includes the internal stakeholders and users as well as external ones (such as the Social Media sales department as well as the potential customers)

5. **Computer Actors:** What systems and applications are affected

6. **Roles and Responsibilities** for the above actors

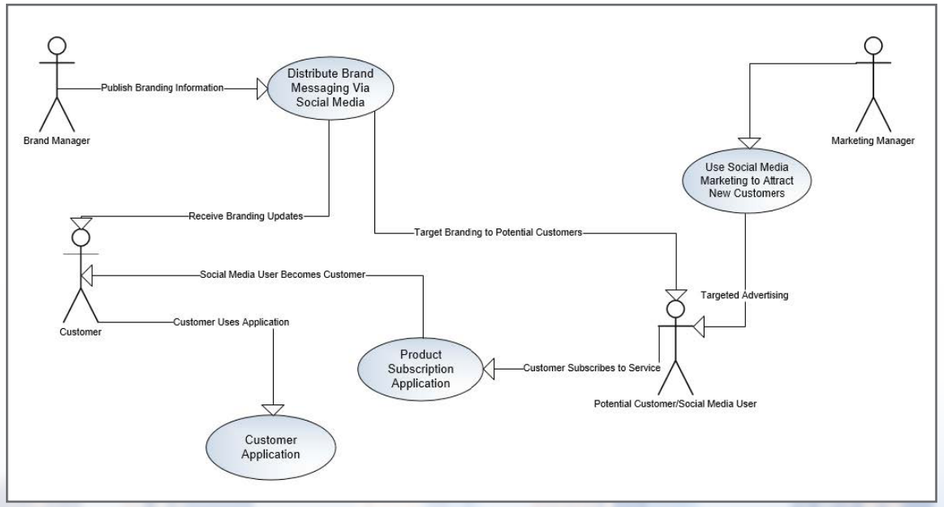

Figure 1- UML Use Case Diagram

Identify Enterprise Application Actors and Processes

So in the above section of gathering business requirements we have identified the actors and the processes they are involved in. Here we will use the UML Use Case Diagram to document the Actors and the processes they interact with.

In this diagram we have the actors:

- Brand Manager
- Marketing Manager
- Social Media User/ Potential Customer
- Customer

And the processes we have in play are:

- Marketing Manager uses Social Media Marketing to attract new customers
- Brand Manager Distributes Brand Messaging via Social Media
- Potential Customer subscribes to product via Subscription Application
- Customer Uses Application

Now of course these are high-level processes but what comes out of this view is that we have identified two subsystems that we need to build a Social Media front end to:

1. The Product Subscription Application
2. The Customer Application

And we additionally have two Use Cases to flesh out into more detailed specifications:

- The Social Media Marketing Use Case
- The Brand Messaging Use Case.

Map Enterprise Process with Social Media Processes

So now that we have identified some of the processes involved, let's drill down into the next step. In this case we are going to take our hypothetically existing subscription process and add a social media front end to it.

The first thing we will do is to create a UML Activity diagram for this interaction. The Activity Diagram lets us map the steps and processes involved in a new customer subscribing to our application service.

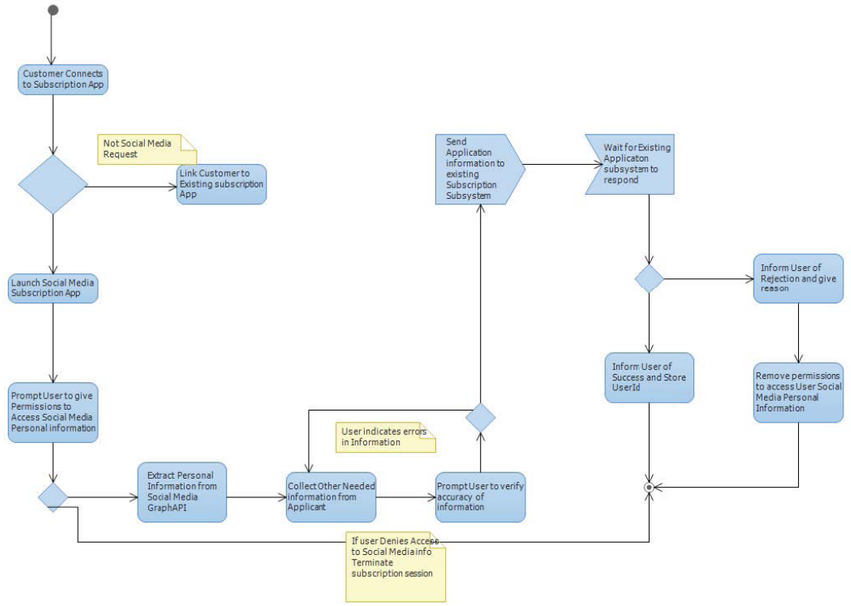

Figure 2 - UML Activity Diagram

This is of course a very simplified case but it shows the interaction flow between the new Social Media subscription application and the existing subscription system. Note how we add a prompt to extract data from the Social Media Graph API as well as other personal information. We also hand off the evaluation of the actual subscription request to the existing subsystem.

The activity diagram is the basis for our coding efforts and would be stored within the Orbus iServer as documentation of the process flows for the new functionality. The Use Case diagram from the previous section would be stored under the Business Goals since Use Cases are usually business process related.

From the Activity Diagram we move on to the Sequence Diagram.

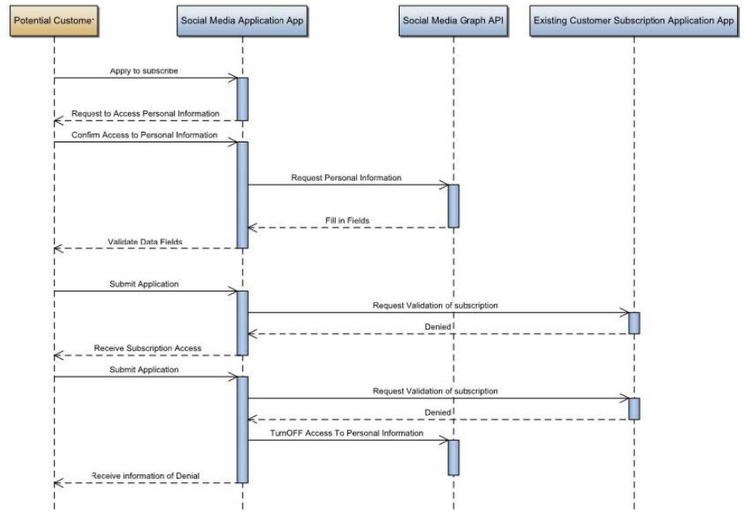

Figure 3 - UML Sequence Diagram

The UML Sequence Diagram is a diagram documenting the message and control flows between the various applications and subsystems. In this case we have the Social Media platform itself, the new Social Media Application App and the existing Enterprise Subscription App. Again this s a fairly simple sequence but in a more complex environment, this sort of documentation is invaluable in identifying what the messaging flows are.

Within the Enterprise Architecture process this would be stored as part of the System Interaction documentation.

Now because there is no code behind this example I have not included the UML Class Diagram. The Class Diagram enables the creation of the object classes that will be used in the code itself. But herein begin the difficulties. Using the Class Diagram can be very helpful in sorting out the object interfaces, the Enumerations, Aggregations, Inheritance and such between the various class definitions. Many UML modeling tools go a step further and allow actual code generation from the UML models.

The problem with doing code generation from the models is that there is not a 100% unambiguous mapping of UML attributes to all modern object oriented languages. In particular there is a fair amount of difference between how Microsoft Visual Studio handles UML and how tools like IBM's Eclipse handle them.

Thus my recommendation is to use UML based class definitions to document the general behavior of the class interfaces to existing systems. However to document the implementation of the actual code for your existing classes will work best if left within the development environment itself and the support systems that are optimized to handle them.

Summary

To summarize:

- **UML Models** – are useful for documenting both functional business cases and processes as well as lower level technical details of your Enterprise Architecture
- The Models can often be stored directly within the Orbus iServer as documents that support the Architecture
- While UML allows for documentation of Class level and even code generation – this is best left for tooling that is specifically built for code development, analysis and management

About Karl Schulmeisters

Karl Schulmeisters is Technology Advisor for the Carver Global Health Group where he provides expert leadership in Business Process Architecture, Enterprise Architecture and Cloud Computing. Karl's current emphasis is on the impact and integration of disruptive technologies into traditional enterprise IT organizations: Cloud, Mobility, Consumerized IT, Machine Learning/Big Data and Social Media.

Karl Schulmeisters is an internationally recognized speaker – his most recent speaking engagement was at the Congress on the Future of Engineering Software in St. Petersburg Russia. He welcomes your comments at *karl.schulmeisters@cg-hg.com*

Orbus Software UK
London

Orbus Software US
New York

Orbus Software AUS
Sydney

Orbus Software RSA
Johannesburg

enquiries@orbussoftware.com | **www.orbussoftware.com**

Seattle Software Ltd. Victoria House, 50-58 Victoria Road, Farnborough, Hampshire, GU14 7PG. T/A Orbus Software. Registered in England and Wales 5196435