Global Journal of Enterprise Information System DOI: 10.18311/gjeis/2017/15646

Using ERP Mashups to Improve Business Processes

Tone M. Rognsoy*

IT Consultant - Systems at EWOS Group, Cermaq ASA, Norway; Tone.Marit.Rognsoy@ewos.com

Abstract

Customizing an Enterprise Resource Planning (ERP) system to fit organizational needs is a complex task. Cost, difficulties with maintenance and upgrade, and loss of vendor support are just some of the factors that cause customizations to fail. Enterprise mashups represent a new way of tailoring an ERP environment in close collaboration with users. We have developed two enterprise mashups for a multinational company in the fish feed industry. The paper describes and discusses how organizations may use mashup technology to improve their processes by customizing their ERP systems. Our study suggests that the case organization indeed has a potential for such improvement.

Keywords: Business Process Improvement, ERP, Enterprise Mash Up

Paper Code: 15646; **Originality Test Ratio:** 6%; **Submission Online:** 08-March-2017; **Manuscript Accepted:** 15-March-2017; **Originality Check:** 08-April-2017; **Peer Reviewers Comment:** 05-May-2017; **Double Blind Reviewers Comment:** 22-May- 2017; **Author Revert:** 12-June-2017; **Camera-Ready-Copy:** 25-July-2017

1. Introduction

Enterprise Research Planning (ERP) made its first serious breakthrough in the 1990s. Since then numerous of companies world-wide have invested and implemented ERP package solutions. Off-the-shelf ERP software packages bring best practice business processes into the organization. However, in many cases this leads to a misfit between functionality and existing business processes as every organization has its unique way of doing things. In the end, the organization faces two options: to customize the system to reflect their processes or to adapt to the system[4].

Customization of ERP has been investigated by several authors[12,18,26,27,29,30,34,35] who, for example, discuss different types of customization and weigh their benefits and costs. The various types of costs, monetary and others, have so far gotten most attention. Manpower expenses, maintenance and upgrade difficulties, and loss of vendor support are important factors that are known to make customization efforts fail.

In recent years, ERP vendors have started to include web 2.0 technologies into their products[10,11]. In 2007, SAP introduced Business By Design which – among other things – includes support for mashup development. In 2011, Infor did the same with the introduction of Mashup Designer. In its simplest form, a mashup is a combination of different information sources and services put together in a single, simple, and user-oriented view. The idea is to make customization easy in order to improve user experience and efficiency.

The purpose of this paper is to investigate whether and how mashup technology can facilitate process improvement by customizing ERP systems. We ask: How can mashup technology be used improve ERP-based business processes? What benefits can be reaped from exploiting mashup technology for business process improvement from a user perspective?

To answer these questions, we have developed two mashups for a multinational organization operating in the fish feed industry, using a design science research approach. We have focussed on mashuping as a light-weight alternative to more complex standard ERP-customization approaches. Because mashups belong in the communication layer of the ERP architecture, the underlying structure of the system is not affected by mashup development, which is an advantage if the mashups can become independent of future upgrades. Employees in our case organization were interviewed both before mashup development to identify improvement needs and afterwards to gain insights about the mashups in use. The rest of the paper presents background (Section 2), research method (Section 3), mashup development (Section 4), discussion (Section 5) and conclusions (Section 6).

2. Background

2.1 BPI and ERP

2.1.1 Business Process Improvement (BPI)

is an incremental bottom-up enhancement of existing processes within functional borders (Davenport, 1993 in Shtub and Karni, 2010). The Balanced Scorecard Institute (2014, p. 1) considers BPI a "*focused change in a business process achieved by analyzing*

*Author for correspondence

the AS-IS process(…), then developing a streamlined TO-BE process in which automation may be added to result in a process that is better, faster, and cheaper". Common drivers for BPI are cost, time, quality and flexibility (Davenport & Short, 1990; Shtub & Karni, 2010), where flexibility is the ability to adapt to variations and future needs (Shtub & Karni, 2010). Davenport and Short (1990) considers cost an insufficient motivation in itself because "*excessive attention to cost results in tradeoffs that are usually unacceptable to process stakeholders*"[7]. Shtub and Karni (2010)[36] suggest that a BPI effort should ask: What can be modified? Why should it be modified? How is it modified?

2.1.2 Enterprise Resource Planning (ERP)

is to standardize the business processes of an organization using comprehensive software package solutions so that the organization can bring standard business processes based on best practice to its customers. Davenport et al. (2004)[6] identify three value drivers for realization of ERP value beyond the simple technical implementation of a system: integrating, optimizing and informing. Whereas introducing an ERP package into an organization can improve its business processes by bringing it in line with current best practice, ERP systems can also hamper BPI because they are so complex and closely intertwined with existing ("as-is") processes that the processes become too hard and too costly to change. ERP complexity can be particularly high if the ERP system has been *customized* to the organization.

2.2 ERP Custimization

2.2.1 ERP customization

is called for when there is a gap between the functionality offered by the standardized off-the-shelf ERP package solutions and the needs and requirements of a particular organization (Pollock et al., 2003). An important motivation is to provide a better fit between the system and the organization's business processes. Because ERP packages are generic by design, some customization is always necessary in order to get them up and running (Brehm et al., 2001), not least because some organizations may not be willing to adapt the ERP system's processes (Brehm et al., 2001). Hong and Kim (2002) note that finding the right fit between the ERP system and the business processes is essential for a successful implementation. Light (2005) studied reasons for ERP customization beyond functional misfit.

Customization should not be done without careful consideration because it has downsides like increased cost, maintenance and upgrade difficulties (Hong & Kim, 2002; Pollock et al., 2003), and loss of vendor support (Brehm et al., 2001). At worst, heavy customization can bring down an implementation project (Momoh et al., 2010). Pollock et al. (2003) notes that many vendors attempt to minimize customization among their customers by only releasing new software versions and upgrades that are compatible with the standard system. Light (2001) also emphasizes that every customization decision must be weighed against the amount of required maintenance, both during daily operations and in connection with an ERP upgrade. Rothenberger and Srite (2009) advocate that customization should only be made in rare circumstances, *such as "when a business process cannot be changed without losing a competitive advantage"* (Rothenberger & Srite, 2009, p. 8).

When Gupta (2000) surveyed several companies to assess ERP implementation issues, one of the identified problems concerned customization, namely the decision about whether to customize or not. Customization becomes a problem with major upgrades of ERP systems. A new upgrade often results in changes to the underlying database. This may in turn affect existing application-programming interfaces (API), forcing companies to rewrite code to adjust to the changes in the APIs (Gupta, 2000).

2.3 Types of ERP Customization

ERP systems usually have three layers (Gupta, 2000, Brehm et al., 2001): a *communication layer* that allows for communication with users through a graphical user interface (GUI), an *application layer* that embodies application logic and business rules, and a *database layer* for storage and retrieval of data. Customization can be performed of all three layers.

Brehm et al. (2001) presents a typology of nine tailoring options, ranging from lower-impact light-weight configuration to heavy package-code modifications on various ERP layers. *"Configuration refers to setting parameters in the package to reflect organizational features; modification refers to changing package code to perform unique business processes, often resulting in loss of vendor support"* (Brehm et al., 2001, p. 1). Building on Brehm et al's (2001) typology, Rothenberger and Srite (2009) group customizations into three main types: configuration/selection, system change, and bolt-ons. Whereas configurations have little effect on future upgrades, changes to system code must often be redone because they are overwritten by upgrades. Bolt-ons are third party packages connected to the system via vendor provided APIs. They are primarily used to supplement functionality and will not be affected unless the connecting interface is changed (Rothenberger & Srite, 2009).

An example of configurations/selections are options and parameters. Options allow changing how results appear and were preferred by the respondents in Gupta's (2000) survey over heavier customizations. Davenport et al. (2004) note that, unlike traditional off-the-shelf software, "*(…) ERP packages are generally structured so that both data and many procedures are represented as parameters in tables (…)"* (Brehm et al., 2001, p. 2). By chang-

ing these parameters organizations are allowed some possibilities to ensure a better fit between the system and the business (Brehm et al., 2001; Rothenberger & Srite, 2009). Using parameters, an organization can set the number of factories or sales offices they have and modify the details of how particular processes should operate or a process step be carried out.

In addition to tailoring type, the following tailoring factors may affect the impact of tailoring (Brehm et al., 2001): extensiveness, number of tailoring types, tailoring quality, changes to stored data and its structure, tailoring interdependence, independence of upgrades, and organizational complexity and geographic dispersion.

2.4 Mashups

In music, mashups are created when artists mix and match music to make new songs. In information systems, mashups are likewise created when users or developers mix and match information elements from one or more sources in new ways. Mashuping allows faster development of software solutions, encouraging reuse over building from scratch (Hinchcliffe, 2008)[16]. Service components and data are combined, visualized and aggregated to create new applications (Hoyer et al., 2008)[19]. Aggregation and linking of data can be done in a simple graphical drag-and-drop interface, allowing users with no or limited programming skills to build their own solutions (Liu et al., 2007). Mashups are a typical example of web 2.0 technology, using lightweight programming models enabling the users to become the developer. Koschmider et al. (2009) propose a framework for classification of mashups that asks *what, where, how to* and *from whom* to mash up.

"What to mashup?" categorizes mashups after the type of components that are combined or integrated (Koschmider et al., 2009). Pahlke et al. (2010) distinguish between: presentation, data, functionality and process mashups. A *presentation* mashup – or *information portal* – retrieves and integrates information from different sources with little customization[1], often using predefined widgets in a drag-and-drop mashup tool (Pahlke et al., 2010). A *data* mashup retrieves, processes, and visualizes data from multiple sources (De Vrieze et al. 2009), for example combining maps and business/sales data (Anjomshoaa et al., 2009; De Vrieze et al., 2009[9]; Pahlke et al., 2010). A *functionality* mashup combines and integrates data from different sources through APIs with the purpose of creating a new service (Koschmider et al., 2009; Pahlke et al., 2010). A *process* oriented mashup focusses on user-interface integration, combining data resources, services, and business processes into one common representation[32].

"Where to mashup?" characterises mashups by their location. A server-side mashup uses resources from a server, whereas a client-side mashup uses resources from a client. Most mashups combine the two (Koschmider et al., 2009).

"How to mashup?" categorizes mashups depending on how resources are integrated or combined. Koschmider et al. (2009) distinguishes between *extraction* mashups, which collect and analyse resources from different sources and merge them into a single content page, and *flow* mashups, through which "*the user customizes the resource flow of the Web page combining resources from different sources [...] within the mashup application*" (Koschmider et al., 2009, p. 3).

«For whom to mashup?» considers the intended users of a masup. Koschmider et al. (2009)[24] differentiate between consumer mashups that are intended for public use and enterprise mashups that are used inside the enterprise. An enterprise mashup can tap into corporate IS as well as external sources, for example combining internal customer data with Google Maps. This allows business users to customize their workspace to individual and heterogeneous needs and create situational applications to fit a changing business environment (Hoyer et al., 2008). Hoyer et al. (2008) present enterprise mashups as an extension to Service Oriented Architecture (SOA), which encourages user-centered development through composition of loosely coupled services. SOA provides enhanced efficiency, agility, flexibility and productivity but, unlike mashups, SOA may require extensive investments in both hardware and software. Also, SOA does not support on-the-fly customization, cannot leverage applications without a web-service interface, and usually requires highly trained SOA developers (Liu et al., 2007).

Mashup development encourages users to take action. Hoyer and Stanoevska-Slabeva (2009a) identify user involvement as a success factor in developing mashups for an organization. Organizations need to get their users to invest time in familiarizing themselves with the technology as well as being willing to use it in their daily operational environment (Hoyer & Stanoevska-Slabeva, 2009a). Hoyer and Stanoevska-Slabeva (2009a)[20] suggest that business users tend to focus on their daily activities which are to solve business problems in their departments, not on exploiting the benefits of new technology advances. The work does not stop when the mashup is deployed and taken into use, one must also consider maintenance.

2.5 ERP Mashups

Kurbel & Nowak (2013, p. 295) point to modern configuration and composition options that go beyond the current state-of-the-art of ERP customization, such as web services, mashups and a business adaption catalogue. The aim is to increase flexibility, reduce need for additional programming, and decrease customization costs. ERP mashups range from simple presentation mashups that can be developed with no or limited programming skills (Liu et al., 2007) to complex mashups that build directly on the SDKs of vendors such as Infor and SAP. Infor advertises

that mashups can be used to tailor their ERP solutions to meet individual or company-specific needs. In Infor's eyes this can be accomplished by:

- "*Putting and grouping together information from several sources.*
- *Making new screens using single elements or parts of existing applications screens.*
- Grouping things together with group elements or tabs" (Infor, 2014b, p. 2)[23]

Kurbel and Nowak (2013) used mashups to customize SAP's on-demand ERP solution which, unlike traditional on-premise systems, run on a shared infrastructure in the cloud. Hofmann (2008, p. 86) argues that mashups and other web 2.0 technologies can shorten the life cycle of ERP systems, which currently average to fifteen years, letting "*smart people script services for special purposes" (Hofmann, 2008, p. 88)*[17]. According to Kurbel and Nowak (2013)[25], introduction of ERP mashuping in an organizaion should involve *key users* who specialize in the ERP system and who have broader access rights than the typical user.

An ERP mashup has three architectural layers which together form the mashup stack: the resource, widget, and mashup layers (Hoyer et al., 2008; Hoyer & Stanoevska-Slabeva, 2009b)[21]. The *resource layer* comprises content, data, or application functionality, each building block with its own context and business logic (Liu et al., 2007, p. 4)[28]. Resource interactions are specified in APIs, which provide instructions or standards on how to access a service – create, read, update and delete are common operations. The *widget layer* provides a graphical face to the resources to hide their technical nature. Widgets are connected to resources through their APIs. They represent information and functions that are specific to the application domain, making them configurable and personalizable to individual requirements. The *mashup layer* wires widgets together to support an activity or a process. Wiring is the process of linking the input and output of different widgets (Hoyer et al., 2008).

Grabot et al. (2014) propose a step-by-step method for improving or redefining processes using web 2.0 tools: (1) choose main target for (process) improvement, (2) list the stakeholders involved, (3) model the process "as-is", (4) choose web 2.0 tool, (5) model the process "to-be" using the tool, and (6) analyse the result. The authors tested their method in two case studies. We will use their method in the rest of the paper, because mashuping is an example of a web 2.0 technique.

3. Method

Our research goal is to investigate whether and how mashups can improve ERP-based work processes. To pursue this goal, we have followed the design science research method (Nunamaker Jr and Chen 1990[31], Hevner et al. 2004)[15]. Design science research can be seen as an embodiment of three related cycles: a *relevance cycle, a rigor cycle,* and a *design cycle* (Hevner 2007). The latter is the heart of a design science project, in which the researcher iterates between construction of the artifact, evaluation and feedback until a satisfactory design is achieved. Hevner and Chatterjee (2010) emphasize that there is a difference between conducting design science research and practicing routine design. The difference lies in the creation of knowledge and communication of the contribution. Design science research generates knowledge on how an artifact can be improved, how it is better than existing solutions, and how it can solve the addressed problem more efficiently (Hevner & Chatterjee, 2010)[14].

3.1 Research Process

Our research process have followed Peffers et al. (2007):

Activity 1 – Problem identification and motivation: We have already explained our research goals and research questions.

Activity 2 – Define the objectives for a solution: Our objective was to develop two mashups (artifacts) to improve two ERP-based processes in our case organization, called *FishFeed* in this paper. We wanted to gain experience with mashuping in practical settings and gain insights from the users' points of view.

Activity 3 – Design and develop: Two mashups were developed: a presentation mashup (*Orders By Location*) and a process oriented mashup (*New Customer Order*). The mashups were developed iteratively until we considered them satisfactory, resulting in two iterations for Orders By Location and three for New Customer Order.

Activity 4 – Demonstrate: In the end, the mashups were presented and demonstrated for FishFeed. The presentation included information on the purpose behind the mashup, an overview of involved ERP systems and information about available functionality.

Activity 5 – Evaluate: While the primary purpose of our research was to explore mashuping in a practical setting, we also wanted to gain insights from the users' points of view. We therefore interviewed the users about the mashups we developed. The interviews were structured according to the Technology Acceptance Model (TAM) (Davis 1993, Venkatesh and Davis 2000), which identifies user acceptance as a key in determining the success or failure of a IS system. TAM evaluates an IS's functionality, usability and fit with the organization through user acceptance (Davis 1993)[8]. TAM has proven to be robust across settings, populations and technologies (Venkatesh, 2000; Venkatesh & Davis, 2000).

Activity 6 – Communicate: The results of our research have been communicated in a report and in this paper.

3.2 Case Organization

Our case organization was the Norwegian customer order department, called *OrderDept* in this paper, of a multinational

company (*FishFeed*) in the fish-feed industry. The company has around 900 employees, but OrderDept was small, having four order consultants and a manager. In addition to receiving phone orders, FishFeed allows its customers to register orders online and by email.

FishFeed uses the ERP system *Infor M3*, which FishFeed had recently upgraded (from version 5.2 to version 10.1) in 2012 and 2013. The focus of the upgrade project was purely technical in the sense that no new functionality was introduced in the organization. The new version 10.1 of Inform M3 nevertheless offered new possibilities, among other things for mashup development, through a built-in tool called *Mashup Designer* and improved search possibilities through *Infor Enterprise Search (IES)*. Mashup Designer offered the possibility to customize the organization's ERP environment through mashup development.

In the spring of 2013 the OrderDept took an initiative to look at the possibility of utilizing these new features to ease their work processes. In February 2013, a departmental meeting took place, where the first author was also present. One meeting topic was how they could take use of the new functionality available. Representatives from the IT department were invited to present possible new functionality, such as personalization of views, creation of mashups, and offering Google-like search within the ERP system. In the end, they decided they could use a mashup for showing a customer's outstanding orders by delivery location.

The purpose and functionality of OrdersByLocation, the presentation mashup, was thus already chosen when our work started. The purpose and functionality of NewCustomerOrder, the process oriented mashup, was chosen by us and FishFeed in cooperation: we conducted semi-structured interviews to identify processes that could benefit from a mashup, finding the need for a mashup that would simplify registration of new orders. The interviews were then used for describing the "as-is" process along with wishes for the "to-be" process. Both mashups were developed in the ERP environment of FishFeed.

3.3 Data Collection

We used three qualitative data collection methods were used: observations, interviews and documents.

3.3.1 Observations

Observations were used before mashup development started, when the first author spent a week in one of the offices of the customer order department (*OrderDept*) of *FishFeed*. The purpose was to get an understanding of the order handlers daily work practices.

3.3.2 Interviews

Interviews were used before development to identify process improvements and afterwards to gain insights about the mashups. Before design and development, we conducted two semi-structured interviews to identify cumbersome work processes within FishFeed. We used focus groups to gain an understanding of what each participant was thinking and to let them know what the others were thinking (Bryman, 2012)[5]. Because OrderDept was small and divided into two offices in different parts of the county, each interview had only two participants.

After design and development, we did four semi-structured interviews to gain insights about our mashups. We again held two separate presentations and did two group interviews, with the department manager present at both presentations. Both mashups were made available for OrderDept. Interviews were conducted shortly after the presentation in an attempt to capture individual opinions about the new solutions. Other case-study techniques – such as interviews and observations – were also adapted for data collection and analysis, as case studies are particularly suited for dealing with knowledge captured from practitioners and in areas with little priori knowledge (Benbasat et al. [3].

Both rounds of interviews followed an interview guide we had developed. The purpose of the guide was to ensure that all desired topics or issues were addressed during the interview. The interview guides are available in parts A and B of the Electronic Appendix to this paper ((the final version will have an URL here)). The interviews were organised according to TAMs thee main variables: *perceived ease of use*, *perceived usefulness* and *intention to use*. A presentation of the artifacts were done before the interviews. The participants were the same as those who participated in the initial interview round. All six interviews were recorded and transcribed, producing 1461 lines and 52 pages of text.

3.3.3 Documents

Documents were used along with interviews prior to development to identify process improvements by collecting written work instructions and educational materials from FishFeed. In total, we collected 1240 lines and 72 pages of text.

4. Mashup Development

4.1 Development Tools

Mashup Designer (Figure 1) was used to develop both mashups. It is a built-in tool that enables end users and developers to create mashups directly in the ERP environment. Mashup Designer allows easy creation of mashups through a drag-and-drop interface without modifying systems code. Both preview and deployment of the mashup become instantly available. "*With mashups a [Infor] user can essentially turn an ERP system into a*

pile of Lego bricks that may be used to build new structures" (Infor, 2014b, p. 1).

When creating a mashup, all the included components are positioned relative to each other in a grid. The main components for showing information retrieved from the ERP system are

- *Detail Panel that shows an individual record;*
- *List Panel* that shows a list of records returned as search results;
- *MI Panel* that shows a panel for a transaction executed through an Infor M3 API call;
- *MI List Panel* that shows a list of records returned by an Infor M3 API call.

Hence, whereas DetailPanel and ListPanel offer generic easy-to-use ways of viewing ERP data, MIPanels and MIListPanels let a mashup communicate with the ERP system's database or external sources using APIs, in order to fetch, add, update and delete data in user-specific ways. Standard components like buttons, textboxes, combo boxes, and splitters etc. are also available using the drag-and-drop interface.

Advanced users can also create and edit mashups using eXtensible Application Markup Language (XAML). XAML is an XML-based markup language from Microsoft that allows developers to add more features to their mashup. The XAML code can be edited directly in Mashup Designer, next to the drag-and drop panels. Infor provides its customers with access to the software development kit (SDK), although this possibility was not used by FishFeed.

4.2 Development Process

Both mashups were developed in the ERP environment of FishFeed, where the first author had previous experience as a consultant. We organised the development process according to the six activities that we have already described (Peffers et al. 2007)[33]. For both mashups, activities 2-5 were iterated, two times for OrdersByLocation and three times for NewCustomerOrder. Each iteration interleaved these four activities with the method proposed by Grabot et al. (2014) for introducing web 2.0 tools to business processes, which we also described earlier.

We proceed to present one of the mashups, OrdersByLocation, in fuller detail. Due to limited space, the other mashup, NewCustomerOrder, is described in the Electronic Appendix ((the final version will have an URL here)) associated with this paper. In agreement with FishFeed, information about customers, products and sales prices has been obscured, along with all screenshots of the mashups we show, because they use real data from the organizations' ERP system.

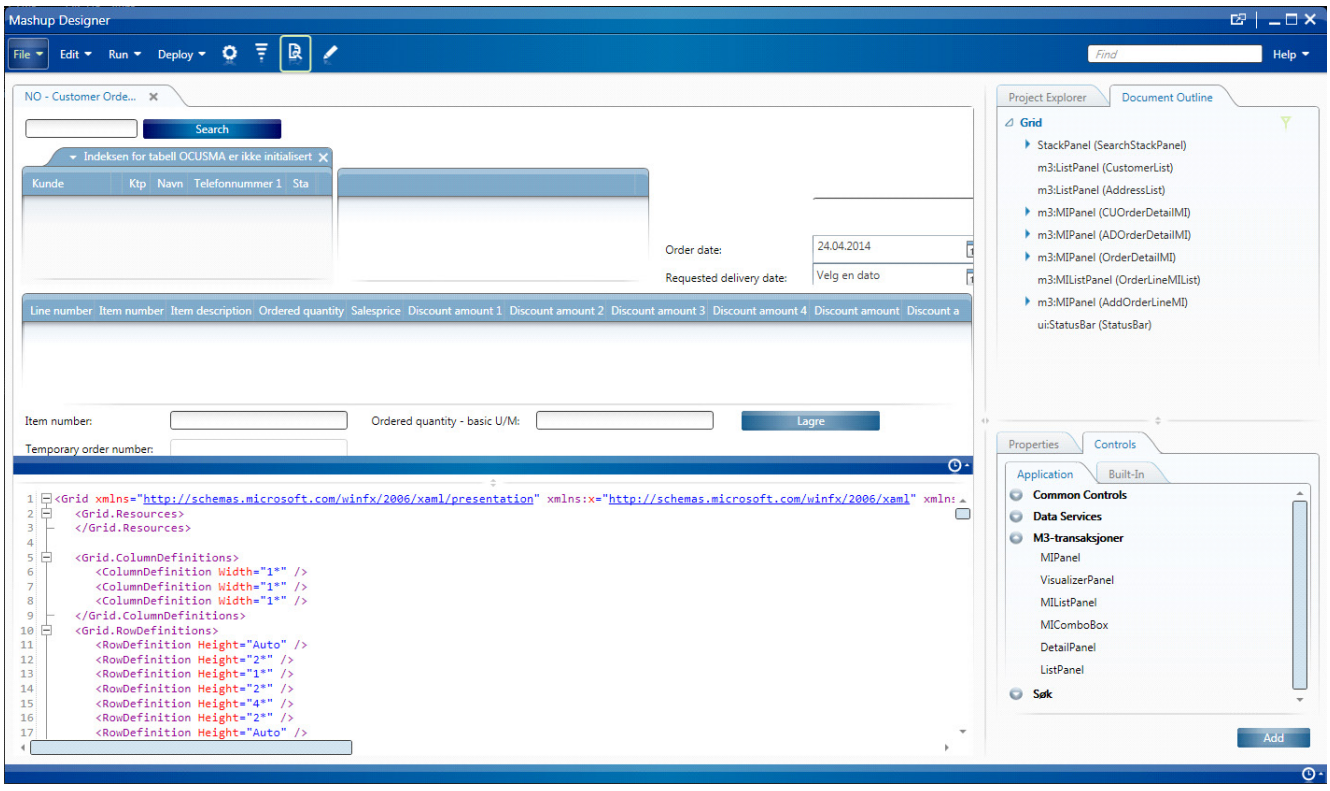

Figure 1. Screenshot of Mashup Designer.

4.3 Development of the OrdersByLocation-mashup

4.3.1 APIs in use

This mashup used two transactions offered by one of the APIs offers by Infor M3 (called *CRS610MI*):

4.3.2 GetAddress

This transaction retrieves customer address data from the ERP system. Its input parameters are *customer number, address type, address number. Its output parameters are: company, division, customer number, customer name, address type, address number, customer address 1, customer address 2, customer address 3, customer address 4, postal code, via address, EAN location code, area / state, run, departure, delivery method, term of delivery, terms text, telephone number, facsimile number, your reference, user, address, valid media, country, VAT registration number, place-replaced by EDE2, route, route departure, unloading zone, marketing id in movex SMS, geographic code, tax code, harbor or airport, place, city.*

4.3.3 ChgAddress

This transaction is used to modify customer address data. Required input parameters are *customer number, address type, customer address, country.*

4.3.4 First Iteration

Step 1 - Problem identification and motivation: OrderDept had already concluded that IT should assist in the development of a mashup. The primary purpose was to aid the staff in getting the whole picture when a customer calls. The mashup should list all orders specified by the customers' delivery addresses, fixing a hole in the original process that does not allow sorting per delivery address (only per customer). A secondary purpose was to introduce the organization to mashuping as it would hopefully show that mashups could improve current businesses by ameliorating problems with the existing ERP system.

Step 2 - Define the objectives for a solution: The objective was to develop a mashup that aids the order staff when a customer calls to ask for a status on his or her orders. As FishFeed also focusses on getting the customers to place orders online, the mashup should offer a quick way of showing whether or not the customer has been using the online solution. Table 1 describes the "as-is" and "to-be" processes for OrdersByLocation.

Step 3 - Design and develop: The mashup is built from three main components, as shown in Figure 2. When the mashup is started MICustomerList (a MIListPanel widget) lists all the customers in the customer registry by customer number and name. A MIListPanel was used rather than a generic ListPanel because we wanted more control over which fields to show for each customer, by using the *GetAddress* operation from the CRS610MI API.

The user must then locate the correct *customer* from the MICustomerList. A search option is implemented taking the customer number as input, displaying the results in the MICustomerList. When a customer has been selected, the AddressList (a ListPanel widget) is populated, showing all available *delivery addresses*. When a delivery address for the customer has been chosen, two things happen:

- The AddressDetailList (a DetailPanel widget) shows information about the customer address, i.e., name, address, zip code and country. It uses the two API transactions GetAddress and ChgAddress that we have described already.
- The list panels within the TabControl widget are populated with order lines. In Figure 2, ListORSL33 (another ListPanel widget) is shown, but not yet populated with data (it will later show the status of all orderlines where the product has been ordred and forwarded to production for picking).The purpose behind the various ListPanels in the TabControl

Table 1. "As-is" and "to-be" scenarios for the Orders By Location mashup

«As-is» process:
The process includes two main stakeholders: the customer and the order consultant (or user). When a customer contacts the order office by phone or email, he may ask for information about his orders, for example an overview of all outstanding orders (orders that are yet to be delivered) or specifics about a particular order he received yesterday. The customer provides the necessary information to the staff at the order office including information about the name of his company. The order consultant then looks up the customer number in the customer registry and opens a program holding information about all order lines. A search is performed using the customer number as input. If the status of the order is known, e.g., not delivered or invoiced, status is given as input to the search query. The program returns a list of all the order lines for that specific customer, which the order consultant has to go through to locate the correct order.
Problem/opportunity:
The list of orders can be very long in cases where customers have many delivery addresses ordering items continually.
«To-be» process:
The improved process allows the user to search for order lines per delivery address, limiting the search results. The improved process is as follows: select the customer, select the delivery address, sort by status and report back to the customer.

is to sort order lines in different ways. The user opens the appropriate tab to minimize the order lines in the list, e.g., if he or she is looking for all orders that are not yet delivered to the customer "Orderstatus [0 TO 33]" (ListORSL33) would be a good place to start. (OrderType=NW1 indicates orders placed by a customer using the online service.)

Step 4 – Demonstrate: The mashup was implemented in FishFeed's ERP environment for the order staff to test. A small presentation was held to explain functionality.

Step 5 – Evaluate: The order staff requested the ability to search by customer number or name to locate a customer, to update address details, and to improve the sorting tabs.

4.3.5 Second Iteration

Step 1 - Problem identification and motivation: The feedback provided during our first evaluation identified the problems for and thus motivated the second iteration. Possible search terms should include customer names to enhance usability, eliminating the need to open the customer registry to identify the customer's number in advance. Also, both the content and the order of the tabs should be altered, creating a clearer and more logical sorting.

Step 2 - Define the objectives for a solution: The objectives this time were to redesign the search function to handle searches using customer names in addition to numbers. Also, the tabs were to be arranged in a more logical order.

Step 3 - Design and develop: The search function was redesigned to leverage the new search service built into Infor M3: *Infor Enterprise Search (IES)*. This service had already been used to populate the list panels in the tab control with order lines based on status or order type (see the XAML code in Figure 3, in which TargetEventName="Search" calls on IES when TargetName="ListCustomers" is clicked).

The MIListPanel widget used for the MICustomerList was replaced by a regular ListPanel widget, which was initialized to list only active Norwegian customers (identified by customer

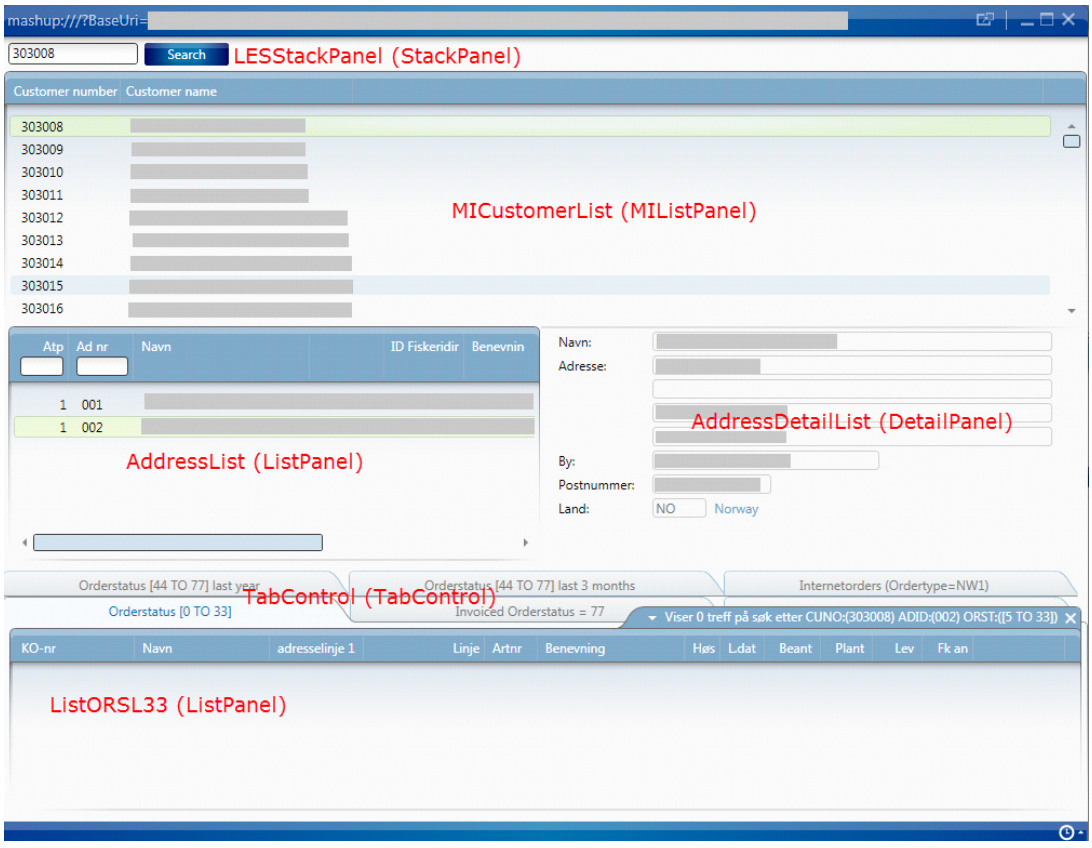

Figure 2. First iteration - Orders by Location.

```
<mashup:Event TargetName="ListCustomers" SourceEventName="Click" TargetEventName="Search">
   <mashup:Parameter TargetKey="CUNM" Value="{Binding ElementName=CUNO_TextBox, Path=Text, StringFormat='\{0\} STAT:20 AND CUNO:[300000 TO 399999]'}" />
   <mashup:Parameter TargetKey="OR CUNO" Value="{Binding ElementName=CUNO_TextBox, Path=Text, StringFormat='\{0\} STAT:20'}" />
   <mashup:Parameter TargetKey="OR ALCU" Value="{Binding ElementName=CUNO_TextBox, Path=Text, StringFormat='\{0\} STAT:20 AND CUNO:[300000 TO 399999]'}" />
</mashup:Event>
```

Figure 3. Search for customer (XAML code).

numbers between 300000 and 399999) who are allows to order feed (identified by status code 20). The MIPanel let us customize the list to show only information about the customer number and name, whereas the regular ListPanel would have shown all the fields stored in the ERP system, such as information on customer number, customer type, name, phone number and status.

The DetailPanel widget behind the AddressDetailList was replaced with a MIPanel widget to allow the user to change information about the delivery address. We created five new tabs (Figure 4):

- "Order status [0 TO 33]" to list all order lines that are registered into the system but not forwarded to production for picking.
- "Order status [44 TO 69]" to list all order lines that are ready for picking, on their way to the customer, or delivered but not invoiced.
- "Order status [77 TO 79] last 3 months" to list all invoiced order lines for the last three months.
- "Order status [77 TO 79] last year" to list all invoiced order lines for the last year.
- "Internet orders [Order type=NW1]" to lists all order lines from orders placed online.

A menu was also included in the top right corner to provide shortcuts to the customer registry and new order registration

Step 4 – Demonstrate: Two presentations were held for members of OrderDept. The manager sat in on both presentations. The functionality and thoughts behind the solution was examined. The participants were allowed to comment on functionality and bring forward their initial reaction. The revised mashup was deployed in FishFeeds ERP environment.

Step 5 – Evaluate: This time, the order staff's main concern about functionality was that the mashup did not sort the order lines based on shipping date. The lack of sorting was due to a confirmed bug in FishFeed's version of Mashup Designer, scheduled to be fixed in the next version. The insights gained in this step will be discussed in the next section.

5. Discussion

This section will reflect on our development work and on the Mashup Designer tool, before we review the insights gained from the users, and interpret and position our findings relative to other work.

5.1 Development Challenges and Mashup Designer as a Development Tool

From the start, Mashup Designer appeared as a simple and effective development tool for creating mashups. It was easy to create a mashup by drag and drop of ERP widgets and to enable users to view information in new ways. The graphical drag-and-drop development interface was useful because it allowed development with minimal programming skills. The embedded XAML editor nevertheless turned out to be one of the strengths of Mashup Designer. A graphical interface makes development easy for users with no or only limited programming skills, whereas the more advanced and demanding users are able to leverage the full power of Mashup Designer using XAML and scripts.

OrdersByLocation was primarily developed using drag and drop. NewCustomerOrder was more complex, as data were entered into the ERP system's underlying database. This requires that the correct data is passed to the database in the correct format. At the same time the interface needs to be simple for the user. For example, while the API require the date to be in the format YYYYMMDD, the users might find it more favorable to be presented with a calendar where they can pick the date. The API transactions are added using drag and drop, but are edited using XAML. The changes done using XAML and the problems encountered with using the APIs took up a lot of the time during the development phase. Implementing functions that were not included in the graphical interface or working with API transactions turned out to be more difficult than using the drag-and-drop interface, so that creating user friendly mashup turned out to be time consuming work, such as displaying sufficient information to the users.

We thus encountered challenges in combing the two development approaches: the graphical drag-and-drop approach and XAML scripting. While the approaches worked fine when used separately, changes made using drag and drop seemed to overwrite changes done directly in the XAML code for the same widget. E.g. if a field was added or removed from a detail panel so that the panel needed to be regenerated, the panel was stripped for all changes that had been added using XAML. As a result it was easy to end up writing the XAML code for function that might have been added twice as quickly using the graphical approach. However, it would have been better if the graphical interface required less detailed knowledge about the fields and tables in the underlying database.

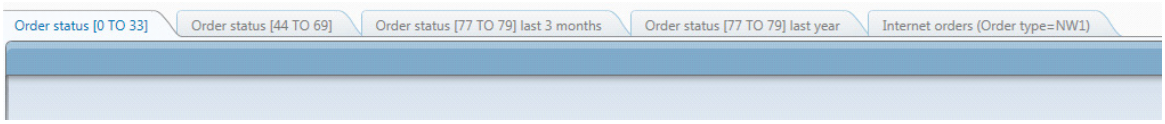

Figure 4. Tabs for sorting order lines.

A big issue during development concerned API documentation. A Infor M3 tool called MITest was used to test the APIs in advance. The tool provides an overview of all the available input and output field as well as which fields/input is mandatory. During development of OrdersByLocation, we discovered that the documentation of mandatory fields was incorrect.

Bugs in the system were the main challenge in the development of OrdersByLocation. Two bugs were discovered. The first prevented the list panels from displaying the chosen view. This bug was reported to the vendor in November2014 and a fix was received in March 2014. The second bug affected the same list as it prevented sorting of rows (order lines). This bug was also reported in January 2014, but no fix was available for the Mashup Designer version that is installed at FishFeed.

Finally lack of experience with the tool and XAML proved to be a challenge during development. This became a problem mainly because there is very little documentation available written or online. The first author attended two courses on Mashup Designer (introduction and advanced) provided by the vendor, in addition to a meeting in the Norwegian M3 user group. Some useful blogs were also located online. As a result of the small amount of documentation there was some communication with a consultant during the development phase. A consultant provided some general tips and tricks, but was not directly involved in the development of the two mashups.

5.2 Insights Gained from the Users

The two mashups – both the OrdersByLocation mashup presented in the previous section and the NewCustomerOrder mashup described in the Electronic Appendix ((the final version will have an URL here)) – were also used in practice by employees in FishFeed. We first held two presentations, one in each order office (using video conference in the most remote office), where both mashups were presented. Afterwards, we conducted four individual interviews with the employees working in each office, all of them working in OrderDept with order registration and processing. The interviews were organised according to TAM's perceived ease of use, perceived usefulness and intention to use.

The general response was that the two mashups appeared to improve the corresponding ERP work processes and that the participants liked them both, despite certain concerns – mainly about functionality. Applying an iterative development process to the mashup development proved to be a strategic approach. As the users were unsure of both the term and the potential usage, looking at the mashups as they were developed helped them to come up with ideas, wishes and requirements.

5.2.1 Perceived Usefulness

Nevertheless, there was some discussion regarding functionality both during the presentations and in the interviews. The participants suggested that some additional functionality would make the mashups more user-friendly. In the end, there are two alternatives for eliminating the concerns about functionality: training and experience or implementation of additional functionality. The functionality discussion is mainly applicable for NewCustomerOrder, as the function missing in OrdersByLocation is due to a bug in the system that is fixed in a new release.

The participants agreed in general that the two mashups were useful. For example, the order consultants might now be able to complete the process while talking to the customer on the phone. ***OrdersByLocation*** improved the efficiency of order handling by providing search/sort functionality.

NewCustomerOrder improved the registration process by eliminating noise. But there were also some concerns about job relevance and output quality. After the second iteration, the participants were concerned whether the mashup was useful because it was not possible to change the shipping cost or show sales prices and fees. A third iteration therefore implemented a button that updated the order line list with sales prices after the order was confirmed. Ideally the sales prices should have been visible as soon as a line was added to an order, but the pricelists are not available for the API list transaction before the order is confirmed. The problem of shipping costs and fees were not addressed in the third iteration. Using the business logic, the ERP system handles this automatically when an order is placed (provided that the system has all the necessary data available). Although the final mashup provided sufficient functions to register an order, the order consultants argued that it should also be possible to edit the information.

Both mashups were found useful in the sense that they seemed to increase the department's understanding of mashup technology and its potential. When the idea of mashuping and the Mashup Designer tool was first introduced to the employees, they had great difficulties seeing how the technology could be exploited. After a year of talking about mashups and seeing the technology in use the situation has changed. During the interviews some new ideas were proposed. For example, when seeing the order registration mashup in action, one of the employees realized that the same mashup layout could be used to create a mashup for confirming internet orders: when receiving online orders, an employee at OrderDept has to go through the order and confirm it before it is sent to the production planer. The steps are more or less the same as for registering a new order, except that the data are already registered and just needs validation.

5.2.2 Perceived ease of use

Both mashups were perceived as easy to use. They were clear and understandable in the sense that none of the participants had any problems understanding how they worked or how to use them.

Neither mashup was perceived as overcrowded with information as the lists and panels were filled out only when they required attention. One of the participants said "*You simply move your eyes from one view to another*".

OrdersByLocation was considered easy to use. It customized the original process by allowing the users to search per location. The biggest concern was the lack of sorting of order lines, which were sorted by status into different tabs, but which were not sorted within each tab. Unfortunately, the only way to fix this would be an upgrade, which was not an alternative for FishFeed at the moment.

NewCustomerOrder was appreciated for gathering all the information and input fields are into a single view/screen, building on reuse and combination of data. The participants liked that all the relevant data/input fields were presented in the same view. The single view had a positive impact on the users, as they believed that it was easier to get a full overview and there register an order, when everything was mashed together.

Customization of the order registration process was done by consolidating four API transactions: add order head, add order line, add batch text and confirm. Although this new registration form provided a simplified registration process, it also posed a challenge due to the combination of transactions. The use of different API transactions caused a save action for the order head, order lines and text. As one of the participants pointed out, this was not a familiar action for them. As the order line and order text registration relied on a temporary order number provided as output by the order head API, it was crucial to save the order head information. If the order head was not saved this would lead to errors during the registration process. The same applied for the lines and text. If the save button was forgotten, the line or text would not be added to the database.

We developed and implemented NewCustomerOrder to improve registration of customer orders. The current ERP system handles it through the same interface that the order consultants use for updating and removing orders and for crediting. The mashup only takes a segment of this functionality, namely the registration process, and tries to improve it by presenting a registration form fitted to the organization's needs. It is important that the users understand the boundaries of each mashup to find it easy to use. It is vital that the order consultants understand which orders that can be registered using the mashup, and which that still needs to be entered using the ERP system.

Although both mashups were thus perceived positively, there were some discussions regarding functionality both during the presentations and in the interviews. For the OrdersByLocation mashup, the missing functionality was due to a bug in the system that was fixed in a later release.

For the NewCustomerOrder mashup, no update or delete transactions were yet implemented. If a user entered the wrong delivery date or quantity it was not possible to correct it in the mashup, forcing the user to open the original ERP system. This process was a bit clumsy. Another obstacle was the combo boxes. It proved difficult to create a mashup basing the design on the use of combo boxes, as they limited the possibility for presenting sufficient information to the user. For example, listing delivery methods in a combo box resulted in a list showing the code "M8J" for the different options, rather than the name "M/S Olympic". In the example, the delivery method is a boat. For this reason the initial form design for the first iteration was abandoned in the creation of NewCustomerOrder.

In sum, perceived ease of use is best illustrated in the words of OrderDept's manager "*it is almost like I can begin to register orders*". The statement is a good example of how the process has been simplified. To draw advantages from mashup customization, like offering new functions and increasing the efficiency of employees, it is however crucial that the users find the mashup easy to use.

5.2.3 Intention to use

In TAM, perceived ease of use and perceived usefulness are together considered to determine one's behavior towards using a system again in the future (Venkatesh, 2000)[37,39]. Collecting data about intention to use was therefore not a priority during data collection. We nevertheless posed one question about future use: whether the participants would use the mashups the next time they had the corresponding tasks at hand. All of the participants expressed willingness to use the mashups again.

One of the strengths of OrdersByLocation comes from Infor Enterprise Search (IES), which is another built-in tool from Infor that allows for Google-like searches in M3 data (Infor, 2014a)[22]. The process of fetching, listing and sorting order lines in the mashup is dependent on IES search queries. Although IES does all the hard work, the power lies in the combination with mashups.

However, NewCustomerOrder received some criticism for lack of functionality as already mentioned in the discussion of perceived ease of use. One of the participants stated that "*it is always the easier solution that will be used*", and if a mashup does not fulfill all their requirements, the old way might be the easiest. OrderDept consist of four highly experienced order consultants who all have worked for the company between six and eighteen years. This long experience may play a role in how they perceive the mashups – especially NewCustomerOrder, which supports a task the perform several times every day and which has not changed significantly for the last fifteen years. ERP is a slow changing technology with a fifteen years long average life cycle. Screenshots of the ERP interface for registering orders from 1999 and 2014 shows that only the screen colors had changed. Nevertheless, this particular task was chosen by the consultants,

and they agreed that it was time to make changes to ease registration of orders.

Alongside the presentation of a third version of TAM, Venkatesh and Bala (2008) suggest a set of interventions intended to enhance user acceptance by affecting perceived ease of use and perceived usefulness. Interventions are split into two categories: pre-implementation and post-implementation. Pre-implementation interventions include design characteristics, user partition, management support, and incentive alignment (Venkatesh & Bala, 2008)[38]. These interventions aims at minimizing initial resistance to a new system and to provide a realistic preview of the system allow users to create an accurate perception regarding functionality and usefulness.

5.3 Meaning of Findings and Relation to other Research

ERP customization is a complex topic. While customization potentially brings great benefits, there are many dangers. Mashuping has the potential to present a change in how ERP systems are customized, and we are already starting to see vendors including mashup tools as part of their ERP solutions. Using mashuping, it is possible to develop views/screens that pull together information from multiple sources.

In addition to the possibility for quick and simple development processes, the main advantage of mashups in the area of ERP customization lies in its low impact on future upgrade initiatives. Mashups operates mainly on the communication layer in the ERP architecture. Any communication with the database layer is performed using APIs. This provides mashups with an advantage. As mashups have no effect on the system code, upgrades can be carried out without paying much attention to the organizations mashups. Nevertheless, it is important to remember that mashups may need to be rewritten whenever an upgrade changes any of the connecting APIs.

6. Conclusion

6.1 Summary

We have investigated whether and how mashup technology can be used to improve ERP-based processes. Hopefully, our study contributes to ERP customization and enterprise mashups. For example, it shows one way to build mashups and suggests how practitioners can use mashup technology in their ERP systems.

The "NO – OrdersByLocation" mashup allowed FishFeed to add functionality that was missing from the original ERP-search process. IES was used to facilitate search functionality that sorted order lines per location. As a results orders can now be found with less scrolling up and down looking for the correct line, which hopefully saves the order consultants some time.

The NewCustomerOrder mashup let FishFeed simplify and fit the ERP system's customer order registration process to its unique organizational needs. The result was a single-screen registration process that gathered all the necessary information and input fields.

The participants reported that the two mashup were easy to use, clear and understandable. They appreciated having a customer order registration process where all the data could be entered into a single form on the screen. OrdersByLocation was perceived useful for searching for a customer order line and for determining whether a customer uses the available online order registration service.

The participants were more skeptical to NewCustomerOrder, although they thought it would be quicker than the original process. Their main concern was functionality. Both mashups were equipped with sufficient functions to perform the corresponding task. However, the experienced user group were used to additional functionality and wished for an update transaction to be included into the mashup for registering customer orders.

All participants were positive about using the mashups again. The mashups had potential to improve the tasks of registering orders and searching for order lines. The consultants said the new mashups would allow them to finish the tasks while talking to the customer on the phone. The findings suggest that FishFeed, the department manager and her employees seemed positive towards using mashup technology as a tool for process improvement. At the end of the study, the participating employees all saw potential for new mashups that would improve existing work processes.

6.2 Research Method

As is usual in design science research (Hevner 2007)[13], the mashups were developed iteratively, where each iteration was made up by five main steps: Problem identification and motivation, define the objectives for a solution, design and develop, demonstrate, and evaluate. If the result of an iteration was deficient, we collected feedback from employees of FishFeed to identify and investigate the problems and used them as objectives for a new iteration.

Because mashup technology was new to the organization and the employees, making it hard for them to see its full potential, focus group interviews turned out to be a good choice of data collection method, although we would have liked more participants.

TAM served as a suitable instrument for gaining insights about the mashups, inviting the participants to share their thoughts about perceived ease of use, perceived usefulness and their intention to use the artifacts. It would have been interesting to be able to spend more time with the participants, investigating how their perceptions of usefulness and ease of use changed over time. Furthermore, it would be interesting to collect quantitative

measures on task performance using the mashups and compare them with the original ERP-based tasks.

6.3 Further Work

The findings from our study of two mashups are of course hard to generalize. Further case studies are called for, in larger departments with more participants and time for more thorough interviews. We would also like to see further work on:

- Quantitative measures: we would like to measure how our mashups improve work processes. It would be interesting to see if the mashups reduce task times. Each year, FishFeed registers more than 5000 customer orders in their ERP system. Saving 30 seconds on each registration would free up more than six working days per year.
- Generalization; we need to study mashup-based customization in different departments, organizations and industries in order to generalize our findings.
- Different stakeholders: we would like to study how different types of users and other stakeholders groups react to mashups for ERP customization, e.g., how a user with a technical background differs from a less technical one, or how a new employee differs from an experienced one.
- User mashuping: we would like to let key users develop mashups themselves – or let them lead mashup development assisted by mashup specialists.

7. References

1. Anjomshoaa A, Bader G, Tjoa A. Exploiting mashup architecture in business use cases. Paper presented at the Proceedings of the 2009 International Conference on Network-Based Information Systems (NbiS 2009). 2009.
2. Balanced Scorecard Institute. Definitions. Retrieved 19.05.2014, Available from: https://balancedscorecard.org/Definitions/tabid/145/Default.aspx .
3. Benbasat I, Goldstein DK, Mead M. The case research strategy in studies of information systems. MIS quarterly. 1987; 11(3). https://doi.org/10.2307/248684
4. Brehm L, Heinzl A, Markus ML. Tailoring ERP systems: a spectrum of choices and their implications. Proceedings of the 34th Annual Hawaii International Conference on Paper presented at the System Sciences. 2001. https://doi.org/10.1109/HICSS.2001.927130
5. Bryman A. Social Research Methods (4th ed.). United States: Oxford University Press. 2012. PMid:22398310
6. Davenport TH, Harris JG, Cantrell S. Enterprise systems and ongoing process change. Business Process Management Journal. 2004; 10(1):16–26. https://doi.org/10.1108/14637150410518301
7. Davenport TH, Short JE. The new industrial engineering: information technology and business process redesign. Sloan management review. 1990; 31(4).
8. Davis FD. User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. International Journal of Man-Machine Studies. 1993; 38(3):475–87. https://doi.org/10.1006/imms.1993.1022
9. De Vrieze P, Lai X, Bouguettaya A, Jian Y, Jinjun C. Process-Oriented Enterprise Mashups. Workshops at the Paper presented at the Grid and Pervasive Computing Conference, 2009. GPC '09. 2009; 2009 May 4-8. https://doi.org/10.1109/GPC.2009.20
10. Grabot B, Houe R, Lauroua F, Mayere A. Introducing 2.0 Functionalities in an ERP Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services: Springer. 2013; 104–11.
11. Grabot B, Mayere A, Lauroua F, Houe R. ERP 2.0, what for and how? Computers in Industry. 2014. Doi: http://dx.doi.org/10.1016/j.compind.2014.02.017
12. Gupta A. Enterprise resource planning: the emerging organizational value systems. Industrial Management & Data Systems. 2000; 100(3):114–8. https://doi.org/10.1108/02635570010286131
13. Hevner A. A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems. 2007; 19(2).
14. Hevner A, Chatterjee S. Design Science Research in Information Systems Design Research in Information Systems: Springer US. 2010; 22:9–22
15. Hevner A, March S, Park J, Ram S. Design Science in Information Systems Research. MIS quarterly, 2004; 75(105).
16. Hinchcliffe D. An executive guide to mashups in the enterprise. 2008.
17. Hofmann P. ERP is Dead, Long Live ERP. Internet Computing, IEEE. 2008; 12(4):84–8. Doi: 10.1109/MIC.2008.78 https://doi.org/10.1109/MIC.2008.78
18. Hong K-K, Kim Y-G. The critical success factors for ERP implementation: an organizational fit perspective. Information & Management. 2002; 40(1):25–40. https://doi.org/10.1016/S0378-7206(01)00134-3
19. Hoyer V, Stanoesvka-Slabeva K, Janner T, Schroth C. Enterprise Mashups: Design Principles towards the Long Tail of User Needs. 2008. SCC '08. IEEE International Conference on Paper presented at the Services Computing. 2008 Jul 7-11.
20. Hoyer V, Stanoevska-Slabeva K. The Changing Role of IT Departments in Enterprise Mashup Environments. In G. Feuerlicht & W. Lamersdorf (Eds.), Service-Oriented Computing – ICSOC 2008 Workshops : Springer Berlin Heidelberg. 2009a; 5472:148–54. https://doi.org/10.1007/978-3-642-01247-1_14
21. Hoyer V, Stanoevska-Slabeva K. Design Principles of Enterprise Mashups. Paper presented at the Wissens management. 2009b.
22. Infor. Infor Enterprise Search. Retrieved 09.05.2014. 2014a. Available from: http://www.infor.com/solutions/technology/user-experience/enterprise-search/
23. Infor. Lawson S3 Mashup Designer [Brochure]. Retrieved 01.02.2014. 2014b. Available from: http://www.infor.com/content/brochures/lawsons3mashupdesigner.pdf/
24. Koschmider A, Torres V, Pelechano V. Elucidating the mashup hype: Definition, challenges, methodical guide and tools for mashups. Paper presented at the Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web at WWW. 2009.
25. Kurbel K, Nowak D. Customization of On-Demand ERP Software Using SAP Business ByDesign as an Example. In F. Piazolo & M. Felderer (Eds.), Innovation and Future of Enterprise Information

Systems: Springer Berlin Heidelberg. 2013; 4:289–97. https://doi.org/10.1007/978-3-642-37021-2_21
26. Light B. The maintenance implications of the customization of ERP software. Journal of Software Maintenance and Evolution: Research and Practice. 2001. 13(6):415–29. Doi: 10.1002/smr.240 https://doi.org/10.1002/smr.240
27. Light B. Going beyond 'misfit' as a reason for ERP package customisation. Computers in Industry. 2005; 56(6):606–19. Doi: http://dx.doi.org/10.1016/j.compind.2005.02.008
28. Liu X, Hui Y, Sun W, Liang H. Towards service composition based on mashup. 2007 IEEE Congress on Paper presented at the Services. 2007.
29. Momoh A, Roy R, Shehab E. Challenges in enterprise resource planning implementation: state-of-the-art. Business Process Management Journal. 2010; 16(4):537–65. https://doi.org/10.1108/14637151011065919
30. Nazemi E, Tarokh MJ, Djavanshir GR. ERP: a literature survey. The International Journal of Advanced Manufacturing Technology. 2012; 61(9-12):999–1018. https://doi.org/10.1007/s00170-011-3756-x
31. Nunamaker JF Jr, Chen M. Systems development in information systems research. Proceedings of the Twenty-Third Annual Hawaii International Conference on Paper presented at the System Sciences. 1990. https://doi.org/10.1109/HICSS.1990.205401
32. Pahlke DWII, Beck R, Wolf DWIM. Enterprise mashup systems as platform for situational applications. Business & Information Systems Engineering. 2010; 2(5):305–15. https://doi.org/10.1007/s12599-010-0121-9
33. Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. Journal of management information systems. 2007; 24(3):45–77. https://doi.org/10.2753/MIS0742-1222240302
34. Pollock N, Williams R, Procter R. Fitting standard software packages to non-standard organizations: the 'biography'of an enterprise-wide system. Technology Analysis & Strategic Management. 2003; 15(3):317–32. https://doi.org/10.1080/0953732031000160150 4
35. Rothenberger MA, Srite M. An investigation of customization in ERP system implementations. IEEE Transactions on Engineering Management. 2009; 56(4):663–76. https://doi.org/10.1109/TEM.2009.2028319
36. Shtub A, Karni R. Business Process Improvement ERP: Springer US. 2010; 217–54.
37. Venkatesh V. Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. Information systems research. 2000; 11(4):342–65. https://doi.org/10.1287/isre.11.4.342.11872
38. Venkatesh V, Bala H. Technology acceptance model 3 and a research agenda on interventions. Decision sciences. 2008; 39(2):273–315. https://doi.org/10.1111/j.1540-5915.2008.00192.x
39. Venkatesh V, Davis FD. A theoretical extension of the technology acceptance model: four longitudinal field studies. Management science. 2000; 46(2):186–204. https://doi.org/10.1287/mnsc.46.2.186.11926

Annexure-I

Using ERP Mashups to Improve Business Processes

ORIGINALITY REPORT

6%

SIMILARITY INDEX

PRIMARY SOURCES

1	etds.lib.ncku.edu.tw Internet	25 words — < 1%
2	is2.lse.ac.uk Internet	25 words — < 1%
3	Hevner, Alan. "Design Science Research", Computing Handbook Third Edition, 2014. Crossref	20 words — < 1%
4	bura.brunel.ac.uk Internet	18 words — < 1%
5	brage.bibsys.no Internet	18 words — < 1%
6	domino.fov.uni-mb.si Internet	17 words — < 1%
7	aisel.aisnet.org Internet	17 words — < 1%
8	herkules.oulu.fi Internet	17 words — < 1%
9	Ford, Merryl, Marlien Herselman, and Adele Botha. "Building an Mlearning Research Framework through Design Science Research", Communications in Computer and Information Science, 2014. Crossref	15 words — < 1%
10	blog.csharplearners.com Internet	14 words — < 1%
11	Lafifa, . "Restructuring and Expanding Technology Acceptance Model Structural Equation Model and Bayesian Approach", American Journal of Applied Sciences, 2012. Crossref	13 words — < 1%
12	www.gelog.etsmtl.ca Internet	13 words — < 1%
13	cora.ucc.ie Internet	13 words — < 1%
14	Grabot, Bernard, Anne Mayere, Fabien Lauroua, and Raymond Houe. "ERP 2.0, what for and how?", Computers in Industry, 2014. Crossref	13 words — < 1%
15	Lecture Notes in Business Information Processing, 2009. Crossref	13 words — < 1%
16	Journal of Enterprise Information Management, Volume 25, Issue 5 (2012-09-22) Publications	13 words — < 1%
17	www.evolve.mb.ca Internet	12 words — < 1%
18	Transforming Government: People, Process and Policy, Volume 7, Issue 1 (2013-03-09) Publications	11 words — < 1%

19	eprints.eemcs.utwente.nl Internet	11 words — < 1%
20	Marlies Steenbergen. "The Design of Focus Area Maturity Models", Lecture Notes in Computer Science, 2010 Crossref	10 words — < 1%
21	espace.curtin.edu.au Internet	10 words — < 1%
22	globaltechins.com Internet	10 words — < 1%
23	openarchive.cbs.dk Internet	10 words — < 1%
24	www.cek.ef.uni-lj.si Internet	9 words — < 1%
25	www.ejise.com Internet	9 words — < 1%
26	Journal of Enterprise Information Management, Volume 25, Issue 4 (2012-07-07) Publications	9 words — < 1%
27	Lecture Notes in Information Systems and Organisation, 2013. Crossref	9 words — < 1%
28	www.iceg.net Internet	9 words — < 1%
29	www.osoe-project.org Internet	9 words — < 1%
30	bura.doi.org Internet	9 words — < 1%
31	eprints.aston.ac.uk Internet	9 words — < 1%
32	vuir.vu.edu.au Internet	9 words — < 1%
33	staff.scm.uws.edu.au Internet	9 words — < 1%
34	Rodriguez-Lopez, C., A. Ulla, and R. Garrido. "The first extensive search for sdO pulsators", Monthly Notices of the Royal Astronomical Society, 2007. Crossref	9 words — < 1%
35	www.todayscience.org Internet	8 words — < 1%
36	www.tojet.net Internet	8 words — < 1%
37	doria.fi Internet	8 words — < 1%
38	eprints.qut.edu.au Internet	8 words — < 1%
39	repository.up.ac.za Internet	8 words — < 1%
40	Lecture Notes in Business Information Processing, 2015. Crossref	8 words — < 1%
41	hufee.meraka.org.za Internet	8 words — < 1%
42	Ronel Smith, Marita Turpin. "Chapter 29 Design Science Research and Activity Theory in ICT4D: Developing a Socially Relevant ICT Platform for Elderly Women in Remote Rural South Africa", Springer Nature, 2017 Crossref	8 words — < 1%
43	repositorium.sdum.uminho.pt Internet	8 words — < 1%
44	opus.bath.ac.uk Internet	8 words — < 1%
45	boris.unibe.ch Internet	8 words — < 1%
46	Lecture Notes in Computer Science, 2014. Crossref	8 words — < 1%
47	www.vvenkatesh.com Internet	8 words — < 1%
48	files.eric.ed.gov Internet	8 words — < 1%
49	dspace.cc.tut.fi Internet	8 words — < 1%
50	projects.business.uq.edu.au Internet	8 words — < 1%
51	Information Technology & People, Volume 27, Issue 3 (2014-09-16) Publications	6 words — < 1%
52	Internet Research, Volume 24, Issue 2 (2014-03-28) Publications	6 words — < 1%
53	M.L. Markus. "Tailoring ERP systems: a spectrum of choices and their implications", Proceedings of the 34th Annual Hawaii International Conference on System Sciences HICSS-01, 2001 Crossref	4 words — < 1%
54	ce.uoregon.edu Internet	4 words — < 1%

EXCLUDE QUOTES ON EXCLUDE MATCHES OFF
EXCLUDE BIBLIOGRAPHY ON

Source: http://www.ithenticate.com/

Citation:

Tone M. Rognsoy

"Using ERP Mashups to Improve Business Processes",

Global Journal of Enterprise Information System. Volume-9, Issue-3, July-September, 2017. (http://informaticsjournals.com/index.php/gjeis)

DOI: 10.18311/gjeis/2017/15646

Conflict of Interest:

Author of a Paper had no conflict neither financially nor academically.