Development of an Anything Relationship Management Prototype for the Smart Factory

Jonathan P. Knoblauch, Rebecca Bulander and Thomas Greiner
Pforzheim University of Applied Sciences, Tiefenbronner Str. 65, 75175 Pforzheim, Germany

Keywords: Anything Relationship Management, Customer Relationship Management, Industrial Internet, Smart Factory, Internet of Things, Services and Data.

Abstract: The Internet of Things, Services and Data (IoTSD) enters into more and more areas of the business, private and public sector. Typical areas are Smart Factory, Smart Home, Smart Grid, Connected Vehicles and Smart City. The area of Smart Factory (also called industrial internet) will be the most important one in the manufacturing sector. For several years There has been another development in information and communications technology (ICT) observable, called Anything Relationship Management (xRM), trying to systematically manage all stakeholders, physical objects and virtual entities of an enterprise through the use of powerful IT platforms. xRM can be used as a cloud management platform for smart industrial production units combined with stakeholder management. In this paper we use xRM to develop a top-down prototype in the Smart Factory environment. The main objective is to demonstrate how xRM could be used in the future Smart Factory. We therefore recreate the structure of an existing machine for mixing liquids as a service on an xRM cloud platform. Furthermore typical data exchange activities between machine and an xRM cloud platform as well as customers and production machine are simulated. The xRM prototype demonstrates why using an xRM cloud platform is helpful for flexile production environments.

1 INTRODUCTION

The Internet of Things, Services and Data enters into more and more areas of the business, private and public sector. Typical areas are Smart Factory Smart Home, Smart Grid, Connected Vehicles and Smart City or newer areas like Smart Farming. Cisco estimates that by 2020, 50 billion devices and objects will be connected to the internet (Evans, 2011). This progress will lead to a huge “Value at Stake” and bring new innovations which are not imaginable today. Cisco believes that the internet of Things and Services will create $14.4 trillion in “Value at Stake” from 2013 to 2022 (Bradley, 2013, p. 1). The area of Smart Factory (also called industrial internet) will be the most important one in the manufacturing sector.

For several years there has been another development in information and communications technology observable, called Anything Relationship Management, trying to systematically manage all stakeholders, physical objects and virtual entities of an enterprise through the use of powerful IT platforms. xRM can be used as a management platform for smart industrial production units combined with stakeholder management.

In this paper we use xRM to develop a top-down prototype in the Smart Factory environment. The main objective is to demonstrate how xRM could be used in the future Smart Factory. We therefore recreate the structure of an existing machine for mixing liquids as a service on an xRM cloud platform. Furthermore typical data exchange activities between machine and xRM cloud platform and customers and machine are simulated. The xRM prototype demonstrates why using an xRM cloud platform is helpful for flexile production environments. The benefit of using xRM is that a flexile cloud platform is provided with the ability to map and create almost any entity, relationship and process in business flieds and to manage them systematically. Next to the field of Smart Factory there are other areas, where to use xRM; just to mention some examples: Business Partner Relationship Management, Emplyee Relationship Management bust also Patient Relationship Management at an hospital or Student Relationship Management at an university.

Knoblauch J., Bulander R. and Greiner T..
Development of an Anything Relationship Management Prototype for the Smart Factory.
DOI: 10.5220/0005444601270134
In *Proceedings of the 5th International Conference on Cloud Computing and Services Science* (CLOSER-2015), pages 127-134
ISBN: 978-989-758-104-5

2 FUNDAMENTALS

2.1 Definition of xRM

The term xRM has already been defined several times in a variety of ways. In a previsous paper we did an extensive literatur review about the term xRM (see Knoblauch and Bulander, 2014). In most definitions xRM is seen as the further stage of Customer Relationship Management (CRM) as well as the realization of the theoretical foundations of relationship management. In addition, xRM includes a technological component (IT system or platform) and a conceptual component (management concept and strategy). In newer definitions xRM is seen as an opportunity to manage objects in the IoTSD (Internet of Things, Services and Data, also called Internet of Everything in an intersectoral way). The following definition covers the main aspects of xRM: “Anything Relationship Management, as an advancement of CRM, is a consistent and holistic concept of Relationship Management between and in-between enterprises, people, physical things and virtual assets. It is based on one or more flexible, modular and scalable IT platforms, which can be focussed on different branches. xRM helps enterprises to capture, coordinate and analyse entities and their relationships as well as processes in the Internet of Everything” (Knoblauch and Bulander, 2014).

2.2 xRM Cloud Platforms

In this section a brief overview about xRM platforms is given. An xRM platform as an extensible foundation provides core functionalities which can be used by multiple modules; each module can interact with each other through defined interfaces (Tiwana et al., 2010). Most xRM platforms use Cloud Computing technologies based on the fundamentals of SOA. xRM platforms provide high-value functions that improve the relationship structures and also any related applications. Business logic and associated user interfaces are based on the defined model. Also the creation of business objects as well as their networking is supported at a conceptual level.

The architecture of an xRM platform can be divided into three layers. The management layer in xRM describes the conceptual approach to manage the n:n interaction, coordination and collaboration between all entities. The middleware layer interconnects people, enterprises, virtual assets and smart objects to create virtual organizations and cross-company business processes. This layer has to be implemented as a highly efficient and dynamical platform with the capability of interoperability. The back-end layer integrates various systems in a homogeneous system landscape. Besides ERP and SCM systems this layer also has to integrate intelligent physical things like Cyber Physical Systems (CPS) or virtual things such as cloud computing services.

xRM platforms should be highly connected and integrated in multiple ways, also across business operations and domain boundaries. The provisioning of electively networked, cooperating, and human-interactive systems will be an essential component in the adoption of such solutions in the future.

In our research project we did also an extensive xRM market analysis over 26 software suppliers which offer CRM platforms with xRM functionnalities. We want just mention some global players out of this analysis to underline the importance of this topic: Microsoft with Dynamics CRM 4.0, Salesforce with their Salesforce1 Plattform or Selligent GmbH with CRM & Interactive-Marketing-Suite (see Knoblauch and Bulander, 2014).

2.3 Advantages through xRM

By using xRM platforms and appropriate management concepts a range of advantages for enterprises are given. In the following the most important ones are summarized.

xRM as platform-as-a-Service (PaaS) provides a cloud-based software development environment for xRM applications (Britsch et al., 2012, p. 86). Such a platform has a flexible and scalable infrastructure as well as the ability of interoperability. Thus, the use of well-defined communication models and communication protocols is necessary (Günthner and Hompel, 2010, p. 79). The software development environment includes components like a repository or has debug functions and the ability to install plug-ins.

Another advantage of xRM is the existence of a configurable framework. Such a framework provides an implementation of important application services like access management or administration functions and a first area of application (typically CRM).

One benefit of xRM is the possibility to build Point-and-Click apps and to customize them easily out of the box. This is one of the core principles that xRM brings along and therefore allows apps to be built fast and easily without the need for extensive implementation skills.

Furthermore, the data models of xRM platforms don't have a fixed schema but a flexible and extensible one. This means, that xRM platforms can hold any data model and can generate or extend the data model without much programming knowledge.

xRM enables the mapping of any kind of entity (stakeholder, virtual asset or physical object) in an application. This allows comprehensive business requirements to be fulfilled on one IT platform. The next level of xRM is integrating smart objects or shared virtual objects through the Internet Things, Services and Data.

Many xRM platforms follow the service orientation paradigm and are built on a service-oriented architecture (SOA). This allows the platform consumer to be served with service orientated capabilities like immediate availability and well-defined behavior of servicers or service composition.

With xRM company-wide and system-wide workflows can be established more easily, since one or more interoperable platforms or well-defined communication standards are in place. This leads to less workflow disruptions and a faster cycle time as well as a more consistent management of workflows and business processes.

Finally any graphical user interface (GUI) of an xRM application can be customized by the user. Depending on user preferences and access restrictions one and the same xRM application can have a completely different GUI.

2.4 xRM, IoTSD and Smart Factory

The Internet of Things, Services and Data is "a dynamic global network infrastructure with self-configuring capabilities based on standards and interoperable communication protocols where physical and virtual "things" [like services] have identities, physical attributes and virtual personalities and use intelligent interfaces and are seamlessly integrated into the information network" (Martinez, 2012, p. 3). In the Internet of Things, Services and Data xRM can be used to define clear relationship structures and to link real and virtual entities dynamically with the right context (Britsch et al., 2012, p. 87). Furthermore, xRM enables the stakeholders of an organization to be connected to enterprise services, virtual assets and physical things. In a nutshell, xRM brings people, things, services and data together on a business platform that allows the systematic management of all relevant business objects.

xRM must also be considered in the context of the Smart Factory. According to acatech (2013) there are three overarching aspects for implementing the Smart Factory (in Germany called "Industry 4.0"): A horizontal integration through value networks, a holistic integration of engineering across the entire value chain and a vertical integration along networked manufacturing systems (acatech, 2013, pp. 20). By using xRM concepts and platforms it will be possible to build powerful solutions across the vertical integration by reconfiguring whole manufacturing systems over an xRM user interface with regards to business use cases. xRM platforms will help to make connections between multiple companies as well as stakeholders in the horizontal integration of interoperable, to share business context and to extend value networks.

3 REQUIREMENTS ENGINEERING

The definition of objectives are essential in the requirements engineering process of software development. We want to highlight the main objectives of the xRM prototype, demonstrate the added value, describe the software development items and explain restrictions in this chapter.

3.1 Achievable Objectives

Objective 1: The proposed xRM prototype should map and link customers, suppliers, employees, business partners and the industrial production units within the xRM application.

Objective 2a: The proposed xRM prototype should simulate an industrial production unit for mixing liquids as a service.

Objective 2b: The service of mixing liquids is explained as an example for an industrial production unit.

Objective 3: The xRM prototype sensors of the industrial production unit should receive fictive sensor values and save them within the entity.

Objective 4: A saved sales order can also be saved as an XML file that could be sent to an industrial production unit for further processing.

Objective 5: A business process of the sales order via the mixed liquid as a service should be demonstrated.

3.2 Advantages and Added Value of the Prototype

Different advantages and added values are shown through the implementation of the corresponding xRM prototype. The main advantage of the xRM prototype is to demonstrate how xRM can be used in the Smart Factory as a platform for relationship management and value network design. The xRM prototype refers to a use case of process engineering. Tt is not useful to think in objects (e. g. with RFID-Tags or barcodes) in process engineering, but rather to think in the category of sales orders and their items as well as production services. The xRM prototype elucidates why this is necessary and useful in the Smart Factory. The elements of the industrial production unit of the Smart Factory are called Cyber Physical Production Systems (CPPS), since they are CPS for production.

The tracing of ingredients, products, batches etc. can be carried out with the xRM prototype. Furthermore monitoring and maintenance of the production machine is enabled by receiving important key figure values like temperature, power consumption, number of revolutions or plant utilization in real-time. This in turn enables machines and their components to be checked remotely through specific xRM GUIs. Key figures can also be used to alert if values are out of range. Moreover an improved accounting and reporting by using actual material and production plant consumption is possible. A stronger relationship between customers und customer needs is given by thinking in services. The customer becomes the producer of his product with the xRM prototype. Customers can choose their production services and start their production process over the cloud.

3.3 Development Items

In this section we want to give an overview of the chosen development items for the xRM prototype. There are three basic development items that are described in the following section.

The objective of the first development item is to model an industrial production unit with all existing components on an xRM platform. The industrial production unit is used to mix two different liquids. Furthermore customers, suppliers, employees, business partners and ingredients are also modeled. Therefore, a suitable entity relationship model (ERM) is needed. After the specification of the ERM this can be used to build the logic on an xRM platform via Point-and-Click-Customization. This is the primary development item of the xRM prototype.

In the second development item the objective is to transfer a sales order with corresponding order items into an XML file that can be sent to an industrial production unit. Besides information about the customer information about the product and the industrial production unit also has to be saved in the created XML file. The idea behind this development item is that information saved as XML can easily be merged in a data exchange format like PLCopen XML. We also want to backtrace the effort that is needed to implement such a function with this development item.

In the last development item we want to implement a simulation of the real-time data exchange of the industrial production unit and the xRM platform. Data out of the machine sensors is saved in the related xRM entity of the xRM platform. Furthermore, each new sensor value is saved within the entity. The current sensor value is always set to the main sensor value attribute. Older sensor values are saved in an XML file.

3.4 Restrictions of the Prototype

The xRM prototype has some restrictions that have to be mentioned. Even though the xRM prototype simulates an existing industrial production unit, there is no direct communication linkage for now. The xRM prototype follows the top-down approach since a working smart industrial production unit is unavailable. Additionally, the data sent to the xRM prototype is randomly generated data. Finally, the generated XML files of the sales order are not sent directly to the industrial production unit, they are saved in a storage location for further processing.

4 IMPLEMENTATION

4.1 xRM Software (SugarCRM)

The xRM platform SugarCRM was chosen to implement the xRM prototype. In its basics SugarCRM is a Customer Relationship Management application that was founded in 2004 as an open source project for Silicon Valley companies. Today there are different product editions of SugarCRM existing. These are a Community Edition (open source), that is licensed under the GNU General Public License and several fee-based software editions. The SugarCRM platform is written in the programming language PHP. SugarCRM has evolved into an xRM platform over the last years, fulfilling xRM principles like the existence of a configurable framework, a plugin installation module and a Point-and-Click functionality. The reason why SugarCRM was chosen is listed below.

1. The platform in its community edition is open source and therefore free in use.
2. SugarCRM offers a big community and there have been lots of installations.
3. The platform can be installed as on-premise software or used as an on-demand service.
4. SugarCRM allows the software developer to easily access and modify the code as well as the database.
5. The platform offers good functions to build Point-and-Click applications.
6. SugarCRM has a consistently good usability.

The Sugar Community Edition 6.5.17 was the chosen version for developing the xRM prototype.

4.2 Data Model and Business Logic of the Prototype

A corresponding data model is needed first for implementing the structure of a smart machine for mixing liquids. This data model includes the following entities.

In figure 1, the data model of the xRM prototype is visualized. In order to reduce complexity not all attributes of the entities are shown.

Table 1: Entities of the xRM prototype – Part 1.

Entity	Description
Customer	A customer is an external stakeholder who wants to buy a certain amount of mixed liquid.
Sales Order	A customer places a sales order to buy mixed liquid.
Order Item	Each sales order has one or more order items that describe what the customer wants to have mixed, with which mixing ratio, how much and in which volume per filling.
Product/ Ingredient	A mixed liquid is made up out of at least two ingredients/products. Therefore, an order item always includes at least two ingredients.
Supplier	The ingredients are delivered by a supplier.
CPPS-Service	A CPPS-Service is a virtual entity. Thus, mixing liquids is defined as a service.
CPPS-Module	The physical modules of a CPPS-Service are named CPPS-Module. A CPPS-Module is a distinguishable part of a machine that is responsible for a specific task in the production process.

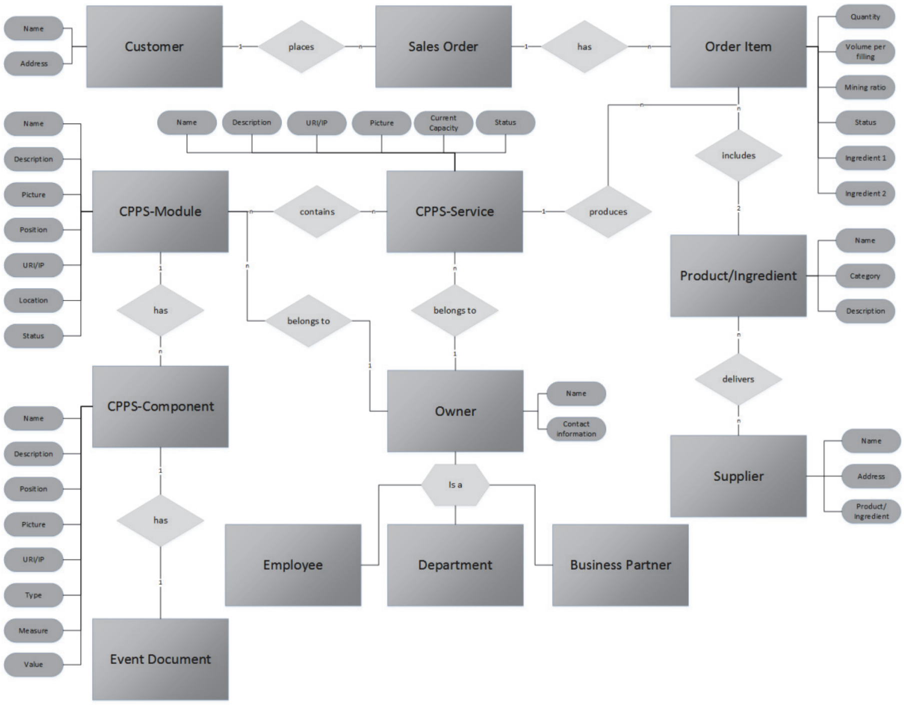

Figure 1: Data model of the xRM prototype.

Table 2: Entities of the xRM prototype – Part 2.

CPPS-Component	A CPPS-Module is built up out of smaller components called CPPS-Components. For example these elements can be seen as the sensors and the actuator of the module.
Event Document	In a CPPS-Component different events are triggered form inside or outside. These events get saved in an entity named Event Document which is linked to the corresponding entity.
Owner	Each CPPS-Service and CPPS-Module has an Owner who is responsible for maintenance, order processing etc. The Owner can be an employee of the organization, a department of the organization or a business partner.

An xRM platform enables an application to be built out of this data model with the Point-and-Click functionality. In SugarCRM this is carried out in the function “Module Builder”. This data model was transferred one-to-one on the xRM platform. A customer places a sales order with order items. Each order item has two ingredients (liquid for mixing) that are delivered by a supplier. In addition, each order item gets a mixing service allocated. This mixing service (CPPS-Service) is responsible for the production of the mixed liquid. A CPPS-Service is constructed out of CPPS-Modules that can be seen as parts of an industrial production unit. A CPPS-Module in turn has various sensors and actuators (CPPS-Components).

CPPS-Components also have an Event-Document that records activities. Furthermore CPPS-Components and CPPS-Services have an Owner who is responsible for production and predictive maintenance.

The next step is to fill the entities with content. Here we focus on the content of the industrial production unit for liquid mixing. Figure 2 shows a liquid mixing service we have defined with five CPPS-Modules that have sensors and actuators.

4.3 Sales Orders via XML

In future customers will configure and produce their own products in the Smart Factory over the cloud. We simulated what such a business process could look like with the xRM prototype (see section 4.5). The customer selects which liquids he wants to have mixed, defines the mixing ratio and the boxing (bottles, barrels etc.) over the xRM cloud interface. Additionally, the customer also chooses a CPPS-Service which will produce his mixed liquid. After the order item is saved the production can be triggered by sending the data of the item to a CPPS-Service.

We implemented a function on the SugarCRM platform in the main menu of the sales order interface that allows the creation of an XML file that contains all relevant data of an order item. This can be done by defining own PHP-classes that are extensions of the class DOMElement.

The XML file can be sent via HTTP-POST to an existing cloud server of the industrial production unit. Communication via SOAP web services is also feasible. In the following, there is a short example what content in the XML file might look like.

This generated example contains information about an order item that wants to have the liquids cola and soda mixed in five bottles with the mixing ration 90/10.

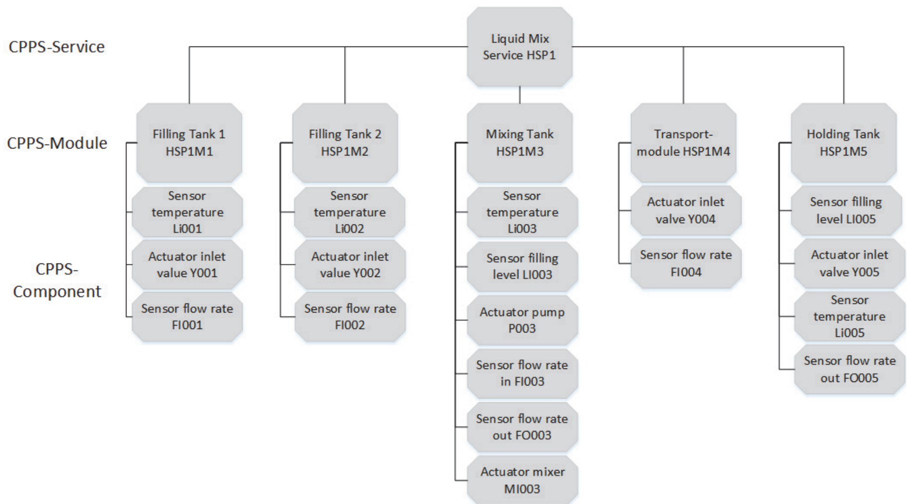

Figure 2: Elements of the liquid mixing service.

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<cppsorder machineid="378909e4-c90f-54bd-fc07-54203c4aa3ee"
        machineadress="http://192.168.101.54">
  <orderitem itemid="5a493058-82ab-816a-ebb0-542bee73ce51">
    <ratio>b90d10</ratio>
    <ingredient1>Cola</ingredient1>
    <ingredient2>Soda</ingredient2>
    <quantity>5</quantity>
    <unit>Bottle</unit>
  </orderitem>
</cppsorder>
```

Figure 3: Example of an XML file for data exchange.

4.4 Real-time Data Exchange via Web Services

The exchange of data is necessary to trace the production process of the mixing liquid service. In SugarCRM data can be sent directly to the platform entities (SugarCRM calls them "Modules") via web services. We implemented a PHP script for the xRM prototype that sends data to all of the CPPS-Components of a CPPS-Service. This PHP script simulates how a mixing liquid production process would actually send data to the SugarCRM platform. Thus, it enables employees to monitor the production process in real-time. Besides monitoring sensor values, finished process steps and status can also be visualized through the platform interface. The following figure illustrates the interface to monitor values of CPPS-Components.

Production process values

(1 - 10 of 26)

Name	Service Modul	Sensor value
.Actuator inlet value Y001	Filling Tank 1 HSP1M1	58.000
.Actuator inlet value Y002	Filling Tank 2 HSP1M2	0.000
.Actuator pump P003	Mixing Tank HSP1M3	600.000
.Sensor filling level LI003	Mixing Tank HSP1M3	10.000
.Sensor flow rate FI001	Filling Tank 1 HSP1M1	125.000
.Sensor flow rate FI002	Filling Tank 2 HSP1M2	150.000
.Sensor flow rate in FI003	Mixing Tank HSP1M3	0.000
.Sensor flow rate out FI003	Mixing Tank HSP1M3	5.000
.Sensor temperatrue	Filling Tank 2 HSP1M2	88.000
.Sensor temperatrue Li001	Filling Tank 1 HSP1M1	54.000

Figure 4: Interface to monitor production process values.

To evaluate past data every new sensor value can also be saved in the entity Event Document that has a relationship to the corresponding CPPS-Component. This event document is linked to an XML file which saves sensor values and time stamps. Hence, it is possible to evaluate past data by analysis tools. Through this approach we want to emphasize the importance of saving data in the entity it belongs to and not to save unstructured data somewhere else.

4.5 Implemented Business Scenario

To illustrate the big picture of the implemented business scenario a corresponding business process is shown in the Business Process Management Notation (BPMN) in figure 5. Thereby, only the important process steps were depicted.

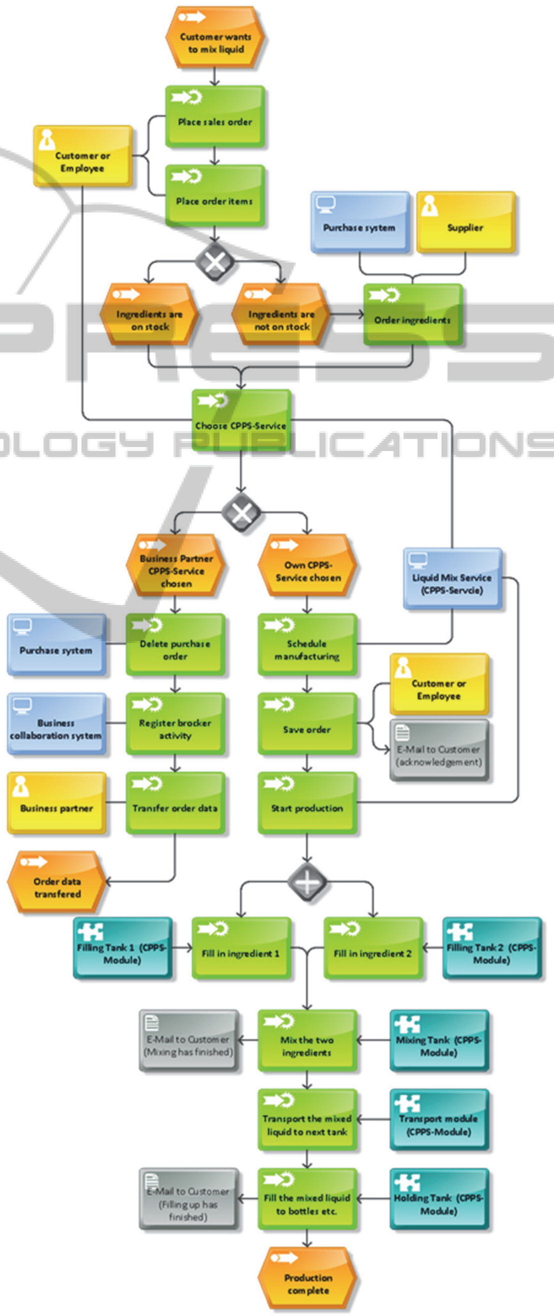

Figure 5: Business process of the xRM prototype.

The business process starts with the event

"Customer wants to mix liquid". While placing the sales order and the order items it is verified if necessary ingredients are in stock. If not, they are ordered from a supplier through a purchase system. Afterwards a CPPS-Service for mixing liquids as a Service is chosen by the customer or by the employee (if customer doesn't have the skills or permissions). Depending on the chosen CPPS-Service the business process on the one hand is transferred to a business partner (if its own organization cannot accept the sales order) and on the other hand further processed in its own organization. If the sales order is transferred to a business partner this is registered and will lead to a brokerage for the organization.

The further process steps inits own organization are the scheduling for the manufacturing, the final saving of sales order with an acknowledgment via email and the start of the manufacturing process when possible or desired. The two filling tanks are first filled with the chosen ingredients of the order item in the manufacturing process. The next step is to mix these ingredients in the mixing tank and to send notification to the customer when finished. In the third step the mix liquid gets pumped to the final holding tank. The last step is to fill the mixed liquid out of the holding tank in chosen volumes per filling (bottles etc.) of the order item and to send another email to the customer when completed. After the manufacturing process is finish the mixed liquid is prepared for shipping to the customer.

5 CONCLUSIONS

Organizations are confronted with a rapidly changing environment today in which relationship management is more important than ever. By using xRM concepts and xRM platforms an approach is given to handle the increasing complexity. In future, production services will also be able to automatically allocate their sales orders among their related industrial production units.

We predict that industrial production units will independently configure themselves according to the relations in the xRM. As an example the liquid mixing service we have shown could have a third filling tank added on the xRM platform. This would create a task in the Smart Factory that ends up by adding such a tank to the industrial production unit and connecting it to the other modules. Vice versa, adding a new tank cloud also automatically creates the relationship in xRM.

Regardless of the data flow direction, xRM platforms and their relationship networks will become more and more important in the future.

ACKNOWLEDGEMENTS

The authors would like to thank the research program of Karl Steinbuch of the MFG Innovation Agency for ICT and Media for the financial support of the research project "Ma-x-RM – Management concept of Anything Relationship Management".

REFERENCES

acatech, 2013. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*. URL http://www.forschungsunion.de/pdf/industrie_4_0_final_report.pdf, (accessed 10/10/2014).

Bradley, J., Barbier, J., Handle, D., 2013. *Embracing the Internet of Everything To Capture Your Share of $14.4 Trillion*. URL: http://www.cisco.com/web/about/ac79/docs/innov/IoE_Economy.pdf, (accessed 12/05/2014).

Britsch, J., Schacht, S., Mädche, A., 2012. *Anything Relationship Management*. In: Business & Information Systems Engineering : BISE (4:2), pp. 85-87.

Evans, D., 2011. *The Internet of Things. How the Next Evolution of the Internet Is Changing Everything*, Cisco. URL: https://www.cisco.com /web/about/ac79/docs/innov/IoT_IBSG_0411F INA L.pdf (accessed: 09/03/2014).

Günter, B., Helm, S., 2006. *Kundenwert. Grundlagen – Innovative Konzepte – Praktische Umsetzungen*. Wiesbaden: Gabler Verlag.

Günthner, W., Hompel, M. (2010): *Internet der Dinge in der Intralogistik*. Berlin, Heidelberg: Springer-Verlag (VDI-Buch).

Knoblauch, J. P.; Bulander, R., 2014. *Literature Review and an Analysis of the State of the Market of Anything Relationship Management (xRM) – xRM as an Extension of Customer Relationship Management*. In: Proceedings of 11th International Conference on E-Business and Telecommunications (ICE-B), INSTICC, Wien, Austria, 28–30 August, 2014, pp. 236–244.

Martinez, C., 2012. *Objective ICT-2013.1.4 - A reliable, smart and secure Inter- net of Things for Smart Cities*. URL: http://www.oko-ist.cz/calls/ncp-infoday_12-06-19/Obj_1_4.pdf (accessed: 10/12/2014).

Mertic, J., 2009. *The Definitive Guide to SugarCRM - Better Business Applications*. USA: Apress.

Tiwana, A., Konsynski, B., Bush, A. A. 2010. *Platform evolution: coevolution of platform architecture, governance, and environmental dynamics*. Information Systems Research 21 (4), pp. 675–687.