DOI: 10.15439/2017F309
ISSN 2300-5963 ACSIS, Vol. 13

Towards an Architecture of Electronic Service Market System - requirements analysis and evaluation

Robert Kutera, Wiesława Gryncewicz
Wrocław University of Economics
ul. Komandorska 118-120, 53-345 Wrocław, Poland
Email: {robert.kutera, wieslawa.gryncewicz}@ue.wroc.pl

Tomasz Skalniak
Leipzig University
Ritterstraße 26
04109 Leipzig, Germany
Email: skalniak@wifa.uni-leipzig.de

***Abstract*—The last decade was a time of a very successful growth of Electronic Markets (EMs). EM is an electronically-supported system which is rather focused on selling physical goods. Electronic Service Market (ESM) is a special case of EM, focused on selling services. The raising business of running ESM seems to be the next possible evolution of the classic EM, and in last few years, several ESM projects were started by various organizations. This paper presents the essential Functional and Architectural Requirements needed in ESM system. The way of their transformation into the architecture of real-life application based on STCMBER meta-architecture was presented. The proposed architecture was validated in the series of pilot implementations performed in compliance with Consortium Research method.**

I. Introduction

A. Structure of the Paper

THIS paper is organized as follows: this section describes the basic concepts of ESM and introduces the basic terms used in this publication. In addition, this section describes the origins of this paper and the used research method. The Section II contains the fundamental, functional (FR) and architectural (AR) requirements of ESM. The Section III introduces the meta-architecture used by the authors in various ESM projects, as well a concrete ESM architecture based on it. The Section IV describes how the architecture presented in the previous section was used in different projects and settings. What is more, this section shows how the used architecture is supporting the requirements described in the Section II. Section V contains the summary and proposes the potential next steps.

B. Basics Concepts and Notation

ESM is a raising business of selling services using the new Information Technologies (IT) [1]. ESM is a type of system which can be compared with the classic EM. In EM the main actors are: consumers, sellers and intermediaries. In ESM context, similar actors can be defined, but the sellers sell services, so we could name them also as "service providers". Service provider is an owner of the business which offers services. The person who is performing the ordered service is called a "service performer" (usually it is an employee of the service provider). Service performer is a new, additional actor, whose presence is distinctive in case of ESM - this role doesn't appear in classic EM. To sell the service, the service consumer (buyer) and the service performer (seller) have to meet. The meeting of the two sides of the transaction can be called simply as "service appointment". The process of coordinating the appointment of the stakeholders of the transaction, and the time needed to execute the service are the distinctive factors of ESM (when compared to the classic EM). Other fundamental, distinctive and unique feature of ESM is the m:n model of the offered service types. There are many electronic stores of the food delivery services, the hotel booking services, the transportation services, the home cleaning services, and other types of services. All similar solutions, offering a specific type of services, are based on the 1:n model - 1 type of services is available for many people. The n:m model means that in one system, different types of services are available to the same group of people. In this context, ESM can be understood as a business innovation: all the possible services available using one account, one password, and the same user interface. In the context of this paper, the "service" term means a "Bricks-and-Mortar" service: a service offered usually by SME and rather addressing the needs of individuals, instead of the needs of other companies [1]. The Bricks-and-Mortar services require to make an appointment between the service performer and service consumer, such a service can't be offered electronically.

What seems to be important to underline, is that the business of ESM means not only running a web store of services. ESM can also be used as an internal tool for enterprise companies or big corporations, or in the context of offering and coordinating public services to public consumers. What's more, in some contexts of use, the transactions done using ESM don't have to be money-based. Sometimes the services can be bought using the internal, artificial currencies, or even can be offered for free, or in the barter-based process. Similar characteristic of the various potential contexts of use can be observed in the classic EM. Described fundamental concepts of the ESM show that the ESM is an evolution of the classic EM. However ESM has some distinctive characteristics which together with some classic EM characteristics create a new class of enterprise systems which requires own architectural models and a new definition of crucial requirements.

This paper describes both functional and architectural requirements of ESM, so the authors discuss both the business-oriented (functional requirements) and technical details (architectural requirements) of ESM. Architectural requirements are the technical requirements which come from the functional requirements. The architectural requirements are the requirements which are analysed in the process of creating the software architecture (SA) [2], [3], [4]. Based on the architectural requirements, the design decisions are made, and a complete set of taken design decisions form a concrete SA. That's why, to propose the model of SA dedicated for ESM, there is a need of defining the crucial architectural requirements of such a system. On the classic electronic markets different types of requirements are already known and analysed. Also some requirements were defined for ESM [5]. This paper's research input is the analysis of the functional and architectural requirements of ESM more deeply, their synthesis and the indication of how ESM architecture is realizing them in real-life use cases.

The next section will introduce the research method used for identification and evaluation of the developed research artefacts.

C. Origins and the Research Method

The reason behind the idea of gathering the most important ESM requirements, and proposing the concept of ESM architecture was to do the next step in the process of building the theory about the ESM. In addition, one of the key theoretical aspects of this paper, was to emphasize the role of the appointment coordination as a key process in the age of service economy [6]. The requirements for the ESM system were researched by using a Consortium Research method [7], [8] which is a design-oriented research method based on the "design science" approach [9], [10]. The main goal and reason of why to use the design-oriented methods is to engage in the research process different stakeholder groups with different points of views and to analyze the research object from different perspectives. This approach is often used in the Information Systems (IS) research projects. The case-oriented research methods (including design-oriented) often produce "solution proposals" as the reserch results. Solution proposals are valid research artefacts of the Consortium Research, and are very often used while researching SA models [11]. It is important to note that the design-science approach is not based on the deductive reasoning, but rather on the natural language deduction by taking into consideration existing models and theories [9]. That is why the process of design-oriented reasearch is always iterative. The iterative nature of the design science enables to validate the research artefact in different contexts and different environments, making it better and better with every iteration. The design science process consists of Analysis, Design, Evaluation and Diffusion (for a detailed explanation please see [7], [8]). This process used in a specific context and under concrete conditions can be named as the Consortium Research method. In the Consortium Research method the research team consists of practitioners and researchers, university workers as well as people from real businesses. They work closely together to find a solution (as a research artefact) which solves a real-life problem. The minimum research duration is 2 years. To ensure that the artefact will be usable in real-life context, the research has to be partly finance by the partner companies (practitioners). The results of the research have to be disseminated in both the researchers and practitioners communities, and the research results have to accessible in a public domain. The research artefacts have to be tested in a business context of the partner companies. The research artefacts don't have to be theoretical - the practical solution is acceptable. [8] Only in rare cases the constructed and validated reasearch artefacts can be formally verified. Rather the artefacts are rejected or accepted by experts, who can judge if the verification was correct or not [9]. In the case of this paper, the research artefact (ESM architecture model) was tested, judged and verified by the partner companies and social organizations who was engaged in the research process, and who used the designed ESM architecture model in real-life cases.

The ESM system built during the research, its requirements and architecture were researched by the Consortium Research team. The research was started few years ago, and the partial results were tested in different contexts and projects in the last 10 years. The developed artefacts, including the discovered appointment coordination (AC) processes, ESM SA, software delivery concepts, social impact of ESM and related-platforms, UI-specific issues related with cross-platform applications, and other topics, were researched thanks to the long and deep engagement of different researchers and practitioners [12], [13], [14], [15], [1], [16], [17], [18], [19]. Right now, for more than 2 years, the team includes 3 universities from 3 different countries (Switzerland, Germany and Poland), 2 different companies (Switzerland), 3 social organisations (one from Switzerland and two from Germany) and more than 20 researchers and practitioners engaged.

In the next section the requirements gathered during many performed iterations will be presented in a synthetic form and marked with proper symbols (FRx, ARx, where x is the subsequent number) for their better identification within the paper.

II. Requirements Analysis

A. Functional Requirements

On electronic service markets, both parties must agree to conclude a contract. There are many requirements that are significant to them. The most important ones are described below. Some of them (e.g. FR1, FR4, FR5 - see below) are inherited from the classic EM systems, and even if they are not unique for the ESM, they play a significant role, and can't be omitted.

First of all, while surfing the Internet people are using different tools: desktop computers (PCs), laptops, tablets, smartphones and other mobile devices. That is why applications for electronic services have to work on various platforms (FR1), not only web platforms, but also mobile ones. Users expect to have access to the system from diverse places and with different devices. The interface should ensure the similar positive user experience on each platform (similar navigation and content logic, easiness of use). Also the process of coordinating the place, time and choosing the actors needed to realize a given bought service is crucial (FR2). Selling a service means, that the service consumer and the service performer need to meet at some point. In addition, in some cases to fully utilize the ordered service, the service has to be served few times, e.g. some massages might need to be repeated few times. The place of the actual execution of the service has to be coordinated too. For example the gardener might offer his services at the client location, at the same time the hairdresser might offer own services only in the hairdressing salon.

The service performers, providers and the service consumers should be able to manage their calendars with the view of all the past and upcoming appointments. Such a view is very important when the negotiation of the date and place of a concrete appointment takes place. What is worth to note, is that some parts of the different AC processes are the same. For example the process triggered by the service consumer, or started by the service provider, can achieve a state of a date and time of the appointment already agreed by both sides of the transaction. From such a state it doesn't matter who started the process, so both the consumer-triggered and provider-triggered coordination processes can have shared, common parts.

Of course managing all the appointments means not only having them all on one view, but also being able to filter them based on different conditions. For example, some appointments which are not yet coordinated should be possible to be filtered. Also, some upcoming appointments might require the action of a specific actor in the coordination process. For example the data proposed by the service performer might not be suitable for the service consumer. So the service consumer should be able to propose an alternative date. Then, the service performer should be able to accept the data, or to propose a different one (or a set of alternative dates). Such a state of the appointment in the AC process requires the performer's action - the process of coordination will not proceed until he/she does the action. To handle the appointments, both sides of AC process should be able to filter the appointments which require their actions to proceed.

What is more, it is very important for both sides to have the possibility of changing the time of an appointment. Nowadays people are very busy, have a lot of additional jobs and duties which can cause unexpected changes in plans. Both, the service provider (and his performers) and the service consumer must be flexible, they must enable changes of previously established appointments.

Very important functional requirement, directly connected with previously mentioned one, is the support for manual and automatic dispatching of appointments to proper performers (FR3). It is very important for providers to have as many automated processes as possible in each possible case. Such an automation of dispatching process can spare their time spent for performing simple repeatable day-to-day operations of looking for free time slots of particular perfomers at the particular time. In both scenarios of automatic and manual decision making, the calendar with already agreed appointments and the free timeslots is needed. In addition, the service provider or the manager responsible for manual dispatching the services requested by the consumers, might see all the companies past and up-coming appointments in one calendar view. Such a functionality might be needed in different use cases, e.g.: while preparing the internal company event (when all the employees have to be available). Dispatching the performer might be done automatically, based on the personal calendars of the performers, or even directly by the performers - the one, who wants to perform the requested service, performs it. The processes of matching the service consumers and providers (needed services) might be very complex. For example, in some cases, the service consumer might want to just select the needed type of the service he is looking for. After selecting the desired type of service, the service provider (and his specific service) could be automatically matched with the consumer (using different algorithms) or could be chosen using different method, e.g. "First-come First-served" (FCFS) method.

Another important requirement is the mutual confirmation of service delivery (FR4). The customer should have the opportunity to express an opinion about the service he has used and about the provider. Also the service provider (or performer) should be able to rate his customers. In that way both sides of the transaction become less anonymous and the overall service market - more transparent. When buying a service, customers don't want to rely only on its description and expect to get familiar with the opinions of other customers. They can give credibility to the service provider. On the other hand, the service performer will gladly perform a service for the customer that has positive opinions. For this purpose, a system of stars or points can be used.

In times of social media expansion an important requirement is also the recommendation possibility (FR5). Satisfied customers could be the best ambassadors of the service and could share the information about particular offer among people in their closest environment. An interesting information may even become part of an advertising campaign for a particular service provider. This phenomenon is described in the literature as viral marketing [20]. An intriguing idea can inspire more and more new audiences to further spread the information and create opinions. In social media it is also very popular to comment and assign "Likes" to interesting offers. These are examples of market stimulation by those who have not used the service, but whose opinion may be of interest to potential buyers.

In addition, some of the users of ESM would like to use a different calendar system then the internal calendar view offered by ESM. There is a big variety of the electronic calendars on the market, many users already do use some of them.

Switching to a new electronic calendar, or using few calendars to maintain all the appointments is very uncomfortable. That's why one of the most important functional requirements is the need of being able to see ESM appointments in different calendars (FR6). Calendars integration is a difficult problem, because there are different implementations on the market, and different implementations use different standards of live synchronization of the content. On the other hand, the need of calendar synchronization is one of the most important issues when it comes to lowering ESM entry barriers.

Service providers, who deliver services on electronic markets, should adopt flexible pricing policy (FR7). They should offer different discounts, promotions and special offers for particular user groups. The possibility of price negotiation is also very appreciated (e.g. I have a large house, so I expect lower window cleaning rates). In addition, services can be reckon up on the basis of duration (e.g. lawn mowing time), number of completed units of service (e.g. number of massages) or level of complexity (eg type of hydraulic failure). Moreover, service settlements should be available in different currencies. Due to the high involvement of people in all kinds of non-profit organizations, the possibility of providing services free of charge should also be considered.

AC processes consist of many possible stages and require a cooperation of both sides in determining the place and time of the appointment. That is why both sides have to be informed about any changes in the process to stay up-to-date all the time. If any action was performed, there is always a side of the transaction, which has to wait until the second side will react. That's why, it's important in ESM, to offer the functionality of notifications about changing the state of the appointment to shorten the waiting time as much as possible (FR8). In case of any action required, such notifications should motivate users to act immediately.

The list of crucial functional requirements described in this section is basic but they seem to be universal for many electronic service markets. That is why this list was chosen as a basis for determining the list of architectural requirements for univeral ESM system.

B. Architectural Requirements

The analysis of the functional requirements of ESM, described in the last section, leads to discover a set of very concrete architectural requirements. The first architectural requirement can be defined as multi-platform character of end-user applications with a strict separation of its layers (AR1) and is related directly to the requirement FR1. Support of different end-user applications (on various hardware platforms) and using/extending the native platform elements (e.g. native platform calendars), leads to the separation of the application logic code and the business logic code. In a result the SA must consist of separate layers with a very detailed inter-layer communication protocol. It defines basic rules of how the layers are dependent on each other, and what kind of communication is allowed in which direction. For example, the communication between the layers, and modules written in different technologies (e.g. Android/iOS application and responsive web site developed in PHP or Python) might be done using the REST architectural style [21]. What is more, using layer-based approach forces the strong authorization policy for ensuring an access to particular layers only for allowed requestors to avoid any security breaches and unauthorized data leaks. For that purpose, OAuth protocol could be used [22].

Another very important requirement is to have the specialized AC services, with the very different AC processes available inside (AR2). It is based on a set of functional requirements: FR2, FR3, FR4. ESM should be usable and understandable for users so that the coordination processes should be similar to each other with unified interfaces and common set of possible actions. The processes should be adapted to the functional requirements, including the manual or automatic appointment dispatching (FR3), flexible pricing policy (FR7) and post-sale phase supporting with two-sided confirmation of performed service (FR4). The processes require automation to some extent so the respective business logic should allow for performing tasks not only on user request (action pinned to the specific button) but also catching some application events and handle them automatically with taking into account the dedicated algorithm. An example could be the already mentioned feature of auto-dispatching appointments to performers who defined their availability in theirs individual calendars (FR3). It also requires to design complex data structures and business entities for appointment, service or user in a way that allows for handling all process tasks. In order to do that it is necessary to equip the SA of ESM with the Business Process Management (BPM) system. The possibility of defining new sub-processes based on the existing processes, having one set of the available actions between the processes, changing the flow of the process without the need of a deep code refactoring and re-engineering the back-end parts of the system, seems to be the most crucial part of ESM system. On the basis of AR2 different design decisions can be made: different BPM engines can be used, different technologies might be needed to address this functionality and to fit into the overall SA. For example, if JAVA technologies are used, maybe the good idea would be to use the Activiti framework [23] as the BPM engine.

The next architectural requirement for ESM is the need of synchronization of different user calendars for the use within AC processes as well as within company management ones (AR3). It seems to be the natural consquence of the FR6 requirement. In order to do that, some native implementations and extensions to the commonly used calendar systems should be made. Consequently, the SA of ESM might contain modules dedicated to different end-user devices (some popular calendars are only available on a very specific hardware platforms).

Functional requirement of flexible pricing policy (FR7) leads to another significant architectural requirement of having the user groups entity which allows for addressing offers to specific public or private groups of users (AR4). What is more, many organizations on the electronic service market would expect to have the multitenancy-ready ESM which allows for not only addressing specific offers, but also providing the tenant-specific interface of the application. For this purpose, new processes for managment of those entities should be designed as well as those entities should be taken into account in main processes of ACS, as it has been mentioned above.

The functional requirement of a fast and convenient notification system (FR8) leads to another architectural requirement. There should be the dedicated component to handle the notifications (AR5), not only for catching events and sending proper messages, but also queuing and postponing the notifications if they are being tried to send at the inappropriate time (e.g. at night). Depending on the specific context, sometimes the notifications engine can work only using the Internet - the notifications can be implemented on the end-user devices using the native platform notifications system (e.g. Android native notifications system), sometimes the SMS engine might be required.

By taking an advantage of efficient notification system the recommendation tool (AR6) could be also designed for satisfying the requirement FR5. The special notification, which will be sent on user request to another person, even if he/she is not the actual ESM user. What is more, the data structures should be designed in a way to allow different kinds of promoting the services within the ESM catalog.

This Section introduced main architectural requirements, and potential design decisions which would have to be taken in real-life use cases. Designing ESM SA is a very complex process of gathering a lot of functional requirements, and based on the aggregated set of such requirements, the architectural requirements have to be defined. Then, based on the defined architectural requirements, the design decisions have to be made. What is more, the strict non-functional requirements should be defined to enable quick, safe and convenient performance of ESM. The whole process of software design is very complex, and it is not a goal of this publication, to address all the problems of such a process.

In the next section the Authors will present the concept of the meta-architecture developed by the Consortium which allows to build on top of it concrete ESMs architectures, based on the modern enterprise design patterns.

III. ESM Architecture

A. STCBMER Meta-Architecture

The meta-architecture is an abstract software model which can be used to build on top of it a concrete SA [24]. STCBMER (Smart Client-Template-Controller-Bean-Mediator-Entity-Resource) is such a component-based [25], [11] meta-architecture [14] designed to support building modern, layered enterprise software systems. STCBMER was comparatively evaluated with other component-based, layered meta-architectures in previous works of the authors [12], [13]. STCBMER is following various architectural principles [13] which are simple pieces of practical architectural knowledge [4], captured by SA researchers and practitioners, and used to build different kinds of software models. Due to the limitations of the paper it is not possible to describe all of them [13], but just as an example: the DDP (Downward Dependency Principle) principle states that the architecture model is top-down, and objects of higher layers depend on objects in lower layers. Based on the principles, STCBMER constructs a component-model which is a graph of several main nodes attached to three layers:

1) Business Logic Layer
    a) Resource
    b) Entity
    c) Mediator
2) Application Logic Layer
    a) Bean
    b) Controller
    c) Template
3) Smart-Client Logic Layer

The base component is Resource - it's main goal is to communicate with the data sources. Entity is component which includes all the data definitions. The Mediator is a main Business Logic component, it contains all the business logic. Mediator manipulates on the Entity objects but is also responsible for storing them in the data sources and caching. What's more, Mediator prepares the REST API available for higher layer - Application Logic. Bean is a component from the Appication Logic layer, it is similar to the Entity, but is never stored in a database - Bean classes rather live in the memory. Controller is a heart of the Application Logic - it consists of server-side application functionalities. Controller has various internal sub-elements responsible for: consuming the Business Layer API, defining the application logic pieces (application processes), defining the REST API for the higher layers. To define the application logic, Controller has to manipulate on data - Bean objects. Both Controller and Bean object are used by the Template to prepare the results of the REST web services of the Application Logic layer. Depending on the needs, Template can prepare UI elements (e.g. HTML, JavaScript, CSS), data (e.g. XML or JSON) and other types of results. The Smart-Client Layer is a layer which contains elements needed to prepare the web application run by the web browser or other end-point environment. It's internal components are usually based on one of classic MVC variants and are usually implemented in the same technology, usually JavaScript. Smart-Client connects to the Application Logic via REST API and complements the logic delivered by the Application Logic layer. Thanks to the Smart-Client Layer, the application is faster and more responsive, because some parts of the processing are performed directly on the end users machines. Smart-Client Layer enables also to cache some parts of the application in the end users devices what allows for using the system in the off-line mode.

B. STCBMER Concrete Implementation

The ESM system (Amiona) prepared by the Consortium (see Section I-C) researching in the field of ESM, was based on the STCBMER meta-architecture. The presented ESM

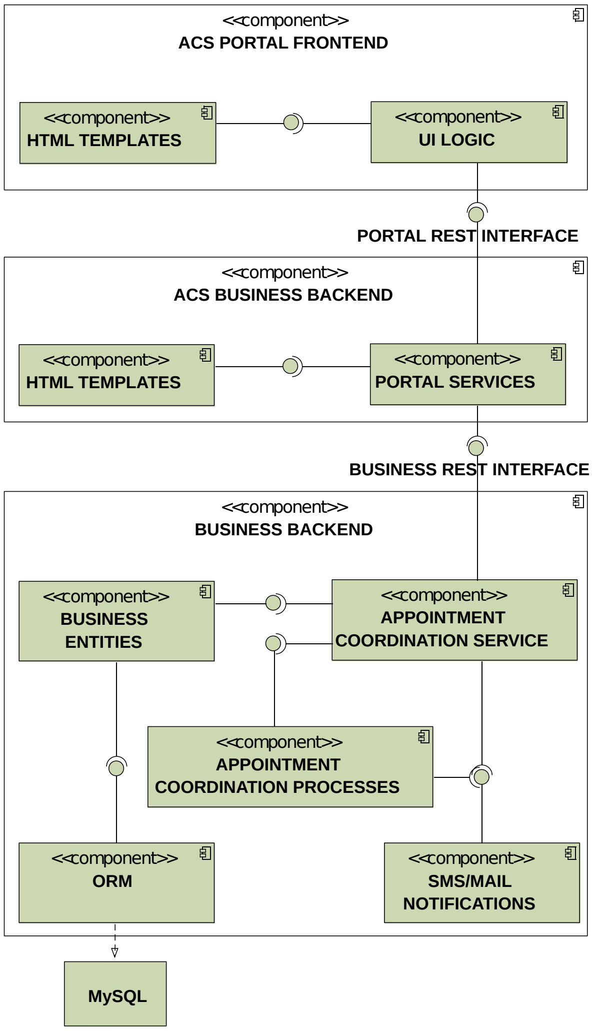

Fig. 1. ESM architecture

architecture (see Fig. 1) consists of 4 layers:

1) Business Backend
2) ACS Business Backend
3) ACS Portal Frontend

The Business Backend is based on the STCBMER Business Logic Layer. The Business Entities and ORM can be mapped as the Entity sub-elements, The Appointment Coordination Service can be mapped as the Mediator. Appointment Coordination Service contains all the crucial ESM elements: implementations of various AC processes as well as the notification engine. In the presented ESM architecture the Resource responsibilities are delivered by the ORM subcomponent (because of the concrete framework used in the implementation).

The Portal Services can be mapped as the Controller, while the HTML Templates can be understood as Template.

The ACS Portal Frontend is based on the Smart-Client Layer concept. UI Logic is containing all the client-side logic (including data definitions), while the HTML Templates produce the pieces of HTML to build the UI.

It is worth noticing that both Appointment Coordination Service and Portal Services are both equipped with the REST interfaces, and could be consumed by other, third-party system as well. The data source used in the presented ESM architecture is MySQL, the basic frameworks and libraries used in this architecture implementation are: SQL-Alchemy (ORM, Entity), Pyramid (Appointment Coordination Service, Appointment Coordination Processes, Portal Services), Mako (ACS Business Backend: HTML Templates), Angular.js (ACS Portal Frontend), Bootstrap (HTML Templates).

To read more about how frameworks can fit into STCBMER, and other software models, see [14].

The web application for ESM, implemented on the basis of this architecture, was used and deployed for various market conditions. Those use cases gradually delivered new requirements in compliance with the principles of chosen research method. Thanks to it the application was evaluated by real customers and the current state of the application architecture is the joint effect of gathering feedback from various use cases. The next section will provide information how the proposed architecture meets the special requirements in different community settings (rural, small town and urban).

IV. Evaluation

A. *ESM in small town setting*

The Consortium (details in Section I-C) organized three pilot implementations of Amiona ESM in three different cities in Northern Germany (urban setting), Southern Germany (rural setting) and Switzerland (small town setting). These 3 pilots addressed different target groups, focused on different occupational modules, and had to deal with different local systems. Thereby, the final solution is flexible enough to adapt to varying situations, and, consequently, better suitable for a roll-out to different regions. The ESM architecture model was positively evaluated by the experts (see Section I-C) on the basis on their experiences, required business knowledge, and test results. Due to the limitations of the paper, it is not possible to show the details of the pilot-specific systems in a graphical form. Therefore, the pilot descriptions contain only the most important remarks, and some references to the presented requirements (see Section II).

This pilot implementation (Sankt Gallen, Switzerland) is the main foundation for the shape of proposed ESM. The general objective of the pilot implementation is to improve the quality of life of elderly people from small town by facilitating their access to all kinds of brick and mortar services. Their independence is associated with self-attaining extremely broad spectrum of everyday life activities. Consequently, they have an increasing need for services of specialists in specific fields. The main purpose of the ESM in that case is to eliminate the existing problems related to service orders, like fragmentation of suppliers, the complicated process of ordering and negotiating the terms of performance and to reduct to the maximum of the complexity of these processes[26]. The requirements taken into account in this case are: FR1, FR2, FR6, FR8 (full covarage), FR3, FR4, FR5, FR7 (partial coverage). They affect all architectural components of ESM.

The proposed architecture meets these requirements well. First of all, the AC processes are very intuitive (on the basis of AR2) and supported by simple and concrete notifications at each stage (on the basis of AR5). They consist of small tasks which are organized in a specific way for each separate process, like a standard process (consumer-triggered), a provider-triggered process, a direct booking process (request on specific date with automatic acceptance of appointment) or an ad hoc process (when at least one side of appointment isn't identified in the system). All those processes are supported by Appointment Coordination Processes component and respective web services from Appointment Coordination Service component. However the processes are different in general, they use the same tasks in a different order and the user is navigated through the process by the system itself. Information about user appointments is gathered in one common view and all operations on them are accessible from one place.

ESM supports also service providers in day-to-day operations of company management by providing them with tools for managing: company profile, its services, its performers and all company appointments (FR3). In addition, it offers a wide range of price configuration, with choice of the currency, price units and so on (FR7). Discounts could be also defined, e.g. for specific dates offered in direct booking process (which allows for offering discounts to specific time slots, e.g. less popular hours of using a specific service). Users have an access to the calendar which could be synchronized with external calendars with the iCalendar (.ics) file format (AR3) to have a better control of his appointments schedule (FR2, FR3, FR6).

The proposed architecture, thanks to the Smart-Client Layer concept, and using REST as communication channel allows to offer the application simultaneously as the web application (meeting the responsive web design standards) as well as the mobile application for Android devices (AR1/FR1). The backend of the application could be also easily used for providing business logic for other devices, like Smart TV or wereables. Thanks to that user can have an access to his appointments from multiple devices and operate on them easily. Recommendation tool (AR6/FR5) is only partially implemented and relies on the intermediary-triggered approach where intermediary can recommend a particular service offer, which is marked with special formatting on public catalog views. In general this marketplace is using most of the features provided by the ESM and all of them have a high rate of acceptance within the local community of users as well as service providers, who have got the another profitable channel of distribution of their services. The ESM is also continously maintained and improved on the basis of user feedback and experiences from another field trials.

B. ESM in rural setting

The next setting, rural, is focused on the voluntary transportation services (SeniorenMobil) for senior citizens (Weil der Stadt, Southern Germany), supervised by one of the Consortium's end user organisation. The existing process works as follows:

1) The elderly citizen calls in to make a request for a drive.
2) The order is taken by the intermediary via their Internet-Platform or phone.
3) The service provider is informed about the request and looks for a possible driver, most likely to be the scheduled driver for the requested date.
4) The drive is conducted.

In this process, the drivers at first have to schedule the dates on which they are free to drive for the transportation service. They have to be available on the scheduled day via e-Mail or telephone. If they get a request for the scheduled day, they make contact with the customer to confirm the request. Then they pick up the customer and carry out the drive. One of the most important requirements from the end user organization is to support AC by an automatic matchmaking of appointment requests with driver's schedules at the initial stage of the process. It is an important part of the FR3 requirement. ESM, by satisfying the AR2 requirement, provides necessary enhacements to Business Entities component (especially to Performer and Appointment entities) as well as in Appointment Coordination Processes component (improved manual dispatching logic and added the handling of the algorithm). The frontend part, UI Logic and HTML Templates components contain the templates and web services for handling the operations on new calendar view asynchronously. As a result, service providers get the dedicated calendar view for managing theirs performers and theirs availability schedules. The service providers have access to company management tools, and among them the "Availability schedule". The tool is responsible for managing schedules of all performers - drivers in the test case. The provider has permissions to modify schedules of all his performers. In parallel, performers (drivers) are able to see all the schedules in their personal profiles, but they can only modify their own schedule by entering or editing availability slots. As already mentioned, in the particular AC process, a special algorithm of automatic dispatching was included. This algorithm takes into account drivers' availability, and the appointment is dispatched to the first available driver, whose availability slot is long enough to fit the current service duration. For that purpose each appointment should have identified the duration parameter, which depends on the service specificity. The driver, within the AC process, could accept the appointment or refuse it, what triggers the next iteration of the algorithm, when the appointment is proposed to the next driver (all based on normal AC proccess, FR2). If none of appointments is accepted, the intermediary is asked for manual dispatching.

The proposed ESM architecture was able to handle the case-specific requirements and make the whole process very clear and efficient. Users were trained for using the ESM and now the service is offered to the end users. First user feedback shows they like the flexibility of the service, it spares time of both sides, drivers as well as intermediary. The information on availability of drivers is clear and usable on calendar view (as a Gantt diagram) and works efficiently because of using Smart-Client layer concept and processing data asynchronously, mostly on user device.

C. ESM in urban setting

Third pilot implementation takes place in Hamburg, the second largest city in Germany and is performed for social organisation that supervises several neighborhoods and aims at helping elderly people in their inclusion into social life. The first neighborhood, which is using the ESM, is Rungehaus that consists of 73 fully accessible apartments and offers shared space of 220 square meters. Its quarter-related approach of living with secured maintenance and without a lump sum for care is a peculiarity of this model. By integration into the existing housing quarter and the neighborhood of the Rungehaus, living for people who do and do not need assistance, or are old is enabled. Similarly to the Rungehaus, tenants from the extended neighborhood (3 housing complexes that consist of 1,027 apartments owned by the same owner as Rungehaus) have also access to common rooms. The rooms are open to use for the whole neighborhood, and there is a special location for professional service providers. The service providers of the quarter office enable a broad range of services which is available at all times. In general, assistance and supporting services are available for all tenants. The range of services is ensured by a mixed financing of ambulant care service providers and integration assistance.

From the architectural point of view, the most important requirements concern the necessity of modification of the AC process (AR2, AR5). End users expect to have information about engagement opportunities, about different possibilities to become involved as a volunteer, and about searching and offering neighbourly help in the immediate vicinity. Both - provide support and receive assistance - should be possible. Any typical ESM system is able to handle supply-side processes of publishing offers of services. One of the ESM innovative approaches is that the user can define his expecations and create a demand which is propagated to all formal and informal providers (some of them, potientially interested are also directly informed by the notification). The process is being continued by gathering offers for that particular demand, choosing of offer(s), and creating the respective appointment(s) on that basis. Next steps are the same as in standard process. The presence of a new entity in Business Entities component requires to introduce new data structures. This new entity still can share many of these structures with the core entity - Service (e.g. Address or Category data structure).

Another important issue is to support intermediaries in the mediation of volunteers as well as searching and finding volunteers (AR2, FR3). Intermediaries have the possibility to act on behalf of other users and enter a demand for another person if he/she is not able to do that because of some kind of disability. What is more, intermediaries and service providers can create specific user groups and manage them on the separate view. The users gathered in user groups can get special offers with discounts or other benefits depending on business context (AR4). That is why the core AC processes and searching/browsing offers had to be updated (AR2) to be able to handle the specificity of user groups.

The Portal Frontend component is able to meet the specific requirements of having the web application developed in compliance with responsive web design paradigm to be available on such devices as PCs, laptops, tablets and smartphones. All features, including consumer and service providers tools, are available to use on mobile platforms. The implementation in the urban area of Hamburg is only a part of a larger, integrated solution, however this topic goes beyond the scope of the paper. More about the integrated project in: [27], [17], [19]. The first user feedback in test implementation of the integrated solution, including the Amiona ESM instance per neighborhood, provides a good feedback and shows an interest of potential users.

V. Summary and Future Work

The digitalization of the consumer world brings a significant shift for companies. Electronic markets are growing and more and more consumers are looking for convenient ways of satysfying their needs in order to improve their quality of life. The consequence of this shift could be a discipline of life engineering which together with business engineering have to continue to ensure efficiency in the satisfaction of needs [28]. ESM plays a special role in providing access to brick and mortar services to users via the electronic channel in easy, quick and secure way. Such circumstances require a strong software foundation for ESM system. The SA needs to be flexible and allows for its customization for specific cases.

In order to find the universal ESM architecture Authors gathered a set of functional requirements from many real-life use cases and synthesised them into 8 main ones. On that basis the architectural requirements were developed and the final shape of the ESM architecture proposed. The ESM system implemented with the new architecture principles was evaluated in many use cases, however three of them are presented in the paper.

The results of evaluation show that the architecture is flexible enough to handle specific requirements of different territorial settings and business context (from commercial services to voluntary help). Still, the new opportunities emerge and for sure the proposed architecture will evolve over time. However, the taken approach of using Service-oriented architectural principles and modern design patterns can assure the evolution won't turn into a revolution in the next few years.

Not all the described requirements (see Section II) were covered by the Evaluation phase (see Section IV). The two-sided confirmation requirement (part of the AR2) was never implemented in any of the Evaluation iterations. The AC process confirmation was implemented only as a one-sided (the service provider confirms that the service is done, not confirmation on the consumers side is done). The second requirement not yet implemented and validated is the recommendations tool (AR6). In this case, there are no notifications assigned to the recomendation process yet, as well as so far consumers cannot recommend any offer to each other. Both requirements are thought to be a part of the solution, and will be fully implemented in the future iterations.

References

[1] P. Schenkel, P. Osl, and H. Österle. Towards an electronic maketplace for bricks-and-mortar services. In *24th Australasian Conference on Information Systems*, pages 1–11, Melbourne, 2013. ACIS 2013.

[2] J. Bosch. Software architecture: The next step. In F. Oquendo, B. C. Warboys, and R. Morrison, editors, *Software Architecture*, pages 194–199. Springer, Berlin, Heidelberg, 2004. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24769-2_14

[3] M. Babar. A web-based system for managing software architectural knowledge. In J. Yao, editor, *Web-based Support Systems*, pages 305–332. Springer, London, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-1-84882-628-1_14

[4] M. Babar. Supporting the software architecture process with knowledge management. In M. Babar, T. Dingsyr, P. Lago, and H. van Vliet, editors, *Software Architecture Knowledge Management*, pages 69–86. Springer, Heiderberg, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02374-3_5

[5] A. Schmidt, B. Otto, and H. Österle. Integrating information systems: case studies on current challenges. *Electronic Markets*, 20(2):161–174, 2010. [Online]. Available: http://dx.doi.org/10.1007/s12525-010-0037-8

[6] Z. Zhou. *The Development of Service Economy : A General Trend of the Changing Economy in China*. Springer Singapore, 2016. [Online]. Available: http://dx.doi.org/10.1007/978-981-287-901-1

[7] B. Otto and H. Österle. Relevance through consortium research? findings from an expert interview study. In R. Winter, J. L. Zhao, and S. Aier, editors, *Global Perspectives on Design Science Research*, pages 16–30, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13335-0_2

[8] H. Österle and B. Otto. Consortium research. *Business & Information Systems Engineering*, 2(5):283–293, 2010.

[9] H. Österle, J. Becker, U. Frank, T. Hess, D. Karagiannis, H. Krcmar, P. Loos, P. Martens, A. Oberweis, and E. Sinz. Memorandum on design-oriented information systems research. *European Journal on Information Systems*, 20:7–10, 2011.

[10] J. R. Venable. The role of theory and theorising in design science research. In *in: Design Science Research in Information Systems and Technology*, 2006.

[11] T. Vale, I. Crnkovic, E. S. de Almeida, P. A. da Mota Silveira Neto, Y. C. Cavalcanti, and S. R. de Lemos Meira. Twenty-eight years of component-based software engineering. *Journal of Systems and Software*, 111:128 – 148, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2015.09.019

[12] L. Maciaszek, T. Skalniak, and G. Biziel. A meta-architecture for service-oriented systems and applications. In *Fourth International Symposium on Business Modeling and Software Design*, pages 20–28. ScitePress, 2014. [Online]. Available: http://dx.doi.org/10.5220/0005423900200028

[13] L. Maciaszek, T. Skalniak, and G. Biziel. Architectural principles for service cloud applications. In B. Shiskov, editor, *Business Modeling and System Design*, volume 220, pages 1–21. Springer International Publishing, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-20052-1_1

[14] L. Maciaszek and T. Skalniak. Confluent factors, complexity and resultant architectures in modern software engineering: A case of service cloud applications. In *Fifth International Symposium on Business Modeling and Software Design*, pages 37–46. ScitePress, 2015. [Online]. Available: http://dx.doi.org/10.5220/0005885300370045

[15] G. Biziel, A. Pyka, T. Skalniak, J. Słowik. Platforma zarządzania usługami jako narzędzie wspierające życie osób starszych (Service Management Platform as an Independent Living Supporting Tool for Senior Citizens). *Informatyka Ekonimiczna (Business Informatics). Wydawnictwo Uniwersytetu Ekonimicznego we Wrocławiu*, 2(32):33–40, 2014.

[16] Ł. Łysik, R. Kutera, P. Machura. Social Collaboration Solutions as a Catalyst of Consumer Trust in Service Management Platforms - A Research Study. In W. Abramowicz, editor, *Business Information Systems: 18th International Conference*, pages 220–232, 2015. Springer International Publishing. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19027-3_18

[17] L. Maciaszek, W. Gryncewicz, R. Kutera. Integrated Service E-Marketplace for Independent and Assisted Living – Business and Software Engineering Challenges. In B. Shishkov, editor, *Business Modeling and Software Design: 6th International Symposium*, pages 221–231, 2017. Springer International Publishing. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-57222-2_11

[18] W. Gryncewicz, R. Kutera, M. Leszczyńska, B. Butryn. The model of delivering an IT product the designed to activate and support senior citizens in Poland. In M. Ganzha, L. Maciaszek, M. Paprzycki, editors, *Position Papers of the 2016 Federated Conference on Computer Science and Information Systems*, Annals of Computer Science and Information Systems, pages 195–202, 2016. Polskie Towarzystwo Informatyczne. [Online]. Available: http://dx.doi.org/10.15439/2016F209

[19] A. Rot, R. Kutera, W. Gryncewicz. Design and Assessment of User Interface Optimized for Elderly People. A Case Study of Actgo-Gate Platform. In C. Rocker, J. O'Donoghue, M. Ziefle, L. Maciaszek, W. Molloy, editors, *Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health. (ICT4AWE 2017)*, Annals of Computer Science and Information Systems, pages 157–163, 2017. SitePress. [Online]. Available: http://dx.doi.org/10.5220/0006320001570163

[20] J. Berger. *Contagious: Why Things Catch On Paperback*. Simon & Schuster, 2016.

[21] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[22] I. Microsoft. The OAuth 2.0 Authorization Framework https://tools.ietf.org/html/rfc6749 Accessed: 09-05-2017.

[23] I. Alfresco Software. Activiti BPM Platform http://activiti.org/ Accessed: 09-05-2017.

[24] A. Yadin. *Computer Systems Architecture*. Chapman & Hall/CRC Textbooks in Computing. Taylor & Francis, 2016.

[25] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Professional, 2nd edition, 2002.

[26] Department of Information Systems, Wrocław University of Economics. IT in Service Economy Competence Center, *Independent Living* http://itse.ue.wroc.pl/index.php/independent-living/ Accessed: 09-05-2017.

[27] R. Kutera and W. Gryncewicz. Web Oriented Architectural Styles for Integrating Service E-marketplace Systems. In *Proceedings of the 7th International Symphosium on Business Modeling and Software Design BMSDâĂŹ17*, pages 72–80. ScitePres, 2017.

[28] H. Österle. Business oder Life Engineering? *HMD Praxis der Wirtschaftsinformatik*, 51(6):744–761, 2014. [Online]. Available: http://dx.doi.org/10.1365/s40702-014-0097-x