

A service requirements engineering method for a digital service ecosystem

Anne Immonen · Eila Ovaska · Jarmo Kalaoja · Daniel Pakkala

Received: 3 September 2014 / Revised: 2 February 2015 / Accepted: 3 February 2015 / Published online: 14 February 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract A digital service ecosystem enables value creation and the co-development of services in a value network under a common ecosystem regulation. The ecosystem members are able to focus on their core competences and can strengthen their forces by co-operating; yet remaining able to act independently. However, due to regulated environment, the ecosystem elements—i.e. ecosystem members, capabilities, infrastructure and the existing ecosystem assets—have an influence on digital service engineering, especially in the service requirements engineering phase. The main contribution of this paper is to describe how to specify the requirements of digital services in a digital service ecosystem. To this aim, this paper introduces the basic definitions and elements of the digital service ecosystem, and a scenario-based service requirement engineering (RE) method for the digital service ecosystem. A practical example is given to illustrate the use of the RE method. The collected feedback from the RE method users highlights the user experiences on the advantages and limitations of the proposed method.

Keywords Requirements engineering · Digital service ecosystem · Service innovation · Service co-creation

1 Introduction

Recently, digital service providers have been strengthening their forces by co-operating, creating value networks flexibly and dynamically to provide services under the concept of a digital service ecosystem. A digital service ecosystem is a new kind of self-organised environment that addresses openness and dynamicity, enabling collaborative innovation and co-creation among ecosystem members. A digital service can be anything that is delivered digitally, is entirely automated and which is controlled by the customer of the service [1]. Service development in a digital service ecosystem sets new kinds of features for the service engineering process, but also new challenges. RE in a digital service ecosystem is not yet a standardised process, and only few studies exist. Like in service-oriented systems, service ecosystems include RE challenges, such as requirements change and evolution, quality requirements gathering and assessment, and uncertainties caused by the dynamic nature and unknown deployment environment, composition and users [2–7]. Moreover, digital service ecosystems provide new challenges in co-evolution among ecosystem members and in customer participation. Therefore, the models are required to propagate value in service value network, contextualise requirements, map them to sub-systems and communicate them to stakeholders [8,9].

The members of digital service ecosystem aim at the co-innovation and co-creation of new digital services within the dynamic value networks, while the utilisation of existing assets of the ecosystem assists in achieving the business goals. However, the current service engineering (SE) approaches do not define what the ecosystem elements are and how to go further from service innovation to service requirements specification. The SE approaches that utilise the existing assets, such as knowledge, do not consider the

A. Immonen (✉) · E. Ovaska · J. Kalaoja · D. Pakkala
Digital Systems and Services, VTT Technical Research Centre
of Finland, P. O. Box 1100, 90571 Oulu, Finland
e-mail: Anne.Immonen@vtt.fi

E. Ovaska
e-mail: Eila.Ovaska@vtt.fi

J. Kalaoja
e-mail: Jarmo.Kalaoja@vtt.fi

D. Pakkala
e-mail: Daniel.Pakkala@vtt.fi

ecosystem context. Thus, digital service ecosystems require a new kind of RE method that:

- Defines the ecosystem elements that are involved in service RE and defines the phases, activities and techniques to be used in each RE phase,
- Enables to define the role of each ecosystem member in the service RE process, supports the members' participation in all RE phases and provides the practices for co-innovation and co-creation,
- Helps in identifying the role of an ecosystem member in each value network depending on the context, and enables the value co-creation via digital service engineering in accordance of the roles and efforts,
- Assists in innovating new digital services by exploiting the existing resources (i.e. knowledge, assets and services) and maximising their use in different contexts, and
- Makes it easy to develop digital services that are interoperable, available and easily consumed by taking into account the specific capabilities of the ecosystem.

As a result of our work, this paper provides a service RE method for digital service ecosystems. The service RE method provides the following contributions:

- *Definitions* The digital service ecosystem is defined based on a thorough state-of-the-art analysis related to different kinds of ecosystems: business, service and software. Accurate definitions are required in order to get a mutual understanding of what digital service ecosystems embody.
- *Elements* The elements of the digital service ecosystem that influence the service RE have been defined. Definition of the elements that are present in the dynamic structure and behaviour of the digital service ecosystem makes it easier to communicate, negotiate and understand the big picture of a digital service ecosystem and its way of influencing service engineering.
- *Service innovation* The method enables the ecosystem members to innovate digital services by defining the scenarios and use cases that describe business goals and the usage of new digital services. Service innovation is supported by the existing ecosystem assets, such as the domain model, and the templates for requirements elicitation and identification, and for communication, knowledge sharing, negotiation and decision-making.
- *Business analysis* The method enables a multi-analysis approach that helps in making design decisions based on accurate justifications. The analysis provides insights into the business potential of new digital services by exploring market trends, customer needs and existing business know-how, also combining the impact analysis

with the results of the analyses of risks, implementation technology and its complexity.

- *Requirement analysis, negotiation and specification* The method provides a repetitive activity-loop of service requirements analysis, negotiation and specification, where service ecosystem members are actively collaborating in defining a coherent and complete set of service requirements specifications. These specifications are provided as an output of the service RE method to the next phase of the service engineering—service architecture modelling.

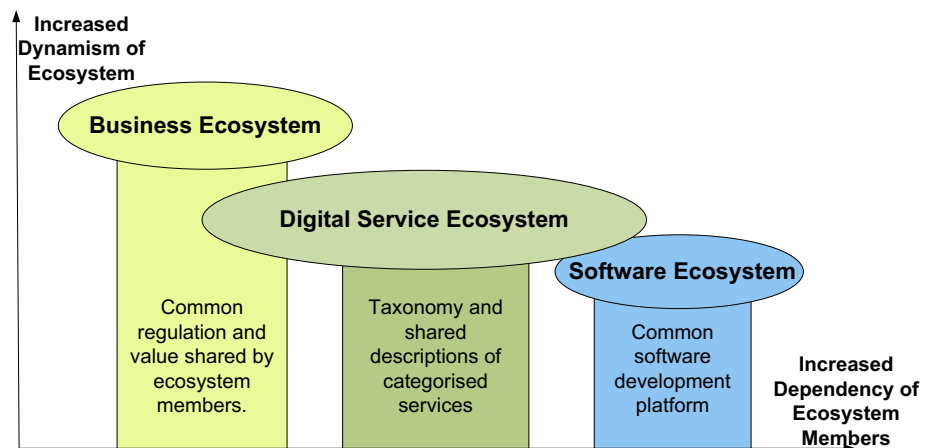
This paper is organised in the following way: Sects. 2 and 3 provide a definition of the digital service ecosystem and the service engineering model as part of it. Section 3 also introduces the service requirements engineering method and practices in the context of a digital service ecosystem. A practical example that guides the use of the service RE method is provided in Sect. 4. Thereafter, in Sect. 5 the lessons learnt are provided, which helps users in adopting the service RE method by illustrating its strengths and shortcomings based on empirical evidences. In addition, Sect. 5 describes our ongoing research on applying the RE method, and future research directions. Section 6 summarises the main research results and closes the paper.

2 State-of-the-art

2.1 Business ecosystem, digital service ecosystem and software ecosystem: definitions

There are three different types of ecosystem definitions that are related to each other: the business ecosystem, the digital service ecosystem and the software ecosystem. The difference between these three is illustrated in Fig. 1. The digital service ecosystem can be positioned between the business and software ecosystem, taking characteristics from both sides and filling the gap between the two. Currently, as far as we know, there exists no research about digital service ecosystems. The business ecosystems [10–12] and software ecosystems [13–15] have been interested researchers extensively. Recently, also research has been carried out on service ecosystems [8, 16, 17]. Service ecosystems are closely related to research in the area of 'service value networks' and the 'Internet of services' [16]. Service value networks provide business value through the agile and market-based composition of complex services from a pool of complementary service modules by the use of ubiquitously accessible information technology [18]. The concept 'Internet of services' considers the Internet as a global platform for the retrieval, combination and utilisation of interoperable services [19]. Thus, especially in the case of web services, the

Fig. 1 Autonomy and dependencies of the different types of ecosystems



service ecosystem has gained a lot of attention in research [16,20,21].

The business ecosystem is a dynamic structure of organisations that work together in a specific core business [12], creating value in a network of actors. The ecosystem's actors affect, and are affected by, the creation and delivery of each other's offerings. Thus, the business ecosystem is composed of inter-member flows of material, energy, knowledge and money [11]. The ecosystem may emerge spontaneously due to a common interest or demand, or as a result of long-term strategic planning. The members share the common ecosystem regulation but are able to act independently, and join and leave the ecosystem freely, since there is no dependency between ecosystem members.

A service ecosystem is a socio-technical complex system that enables service providers to reach shared goals and gain added value by utilising the services of other members in the ecosystem [17,22]. Digital service ecosystem is a part of a service ecosystem, but only covers the digital part, leaving out the purely social part. Digital service ecosystem can be characterised according to [1], being an open, loosely coupled, domain-clustered, demand-driven, self-organising agents' environment, in which each species (human, economic species and digital species, i.e. computer, software and application) is proactive and responsive for its own benefit or profit. The product of a digital service ecosystem is a digital service that is entirely automated and that can be anything that can be delivered through an information infrastructure, e.g. web, mobile devices or any other forms of delivery. For example, the digital service ecosystem can provide the devices and applications as services used by a medical team, but not the whole treatment process (including doctors, nurses, etc.) is provided as a service. Similarly, as in a business ecosystem, partner networks are created inside a digital service ecosystem, but there also exist other dependencies between the ecosystem members than business dependencies. The members share the service taxonomy and service descriptions that can be categorised, for example, by domain,

purpose or technology. The focus is on dynamic, behaviour and conceptual interoperability and interactions between services, and between humans and services.

A software ecosystem has some common elements with digital service ecosystems, such as self-regulation, networked character and shared value [14,15]. However, the definition of a software ecosystem suggests that in general there will be some technology underpinning the ecosystem [13–15], whereas in a digital service ecosystem the members are not bound to a shared development platform or technology. Business and digital service ecosystems can be created dynamically, whereas in software ecosystems, a common platform is required to be developed first. A software ecosystem can be a part of a digital service ecosystem, but in that case the software must be provided as a service to the ecosystem. In a software ecosystem the focus is on technical, syntactic and semantic interoperability and interactions between systems and humans, and there is an increased dependency between ecosystem members.

2.2 Challenges of ecosystem-based service requirements engineering

The importance of service innovation has become the key issue due to dynamicity in customer demand, faster time-to-market, increased competition and the possibilities of co-creation in value networks. Open innovation breaks the boundaries around a company in the innovation phase; the companies can create ideas by themselves and use external ideas or co-create ideas with other companies or the actors of other communities. Due to these characteristics, open innovation is well-suitable for ecosystems. Service innovation can have two forms; outside-in and inside-out [23]. Outside-in innovation is required in cross-domain service engineering and in open data ecosystems that freely exploit the available data. Inside-out innovation focuses on opening internal data, not useful as such for its provider, for other actors' use or for sharing service ideas that an inventor is unable to develop by

himself/herself. In the outside-in process, the external knowledge and innovation components are used in service development, whereas in the inside-out process, the company allows external parties to use its knowledge and innovation components in the service development. In an ecosystem, the value is co-created in a value network, which can be formed as a result of long-term co-operation, or dynamically among members to reach the solution. Several value networks co-exist inside the ecosystem. Value networks can be formed already during the service innovation phase, when each actor has his/her own interest in the service.

As the digital service ecosystem enables members to utilise the methods and technologies that best suit their own needs, two main elements are required to be defined and provided by the ecosystem to engineer services in an ecosystem:

- *The ecosystem infrastructure* is required to make services interoperable, available, and easily consumed and thus manage all service ecosystem operations [22,24,25].
- *The knowledge repositories* are required for storage of the collaboration models, service descriptions and ontologies of service types to support interoperability validation [25–28].

The intent of the knowledge management model is to guarantee the effectiveness of the service ecosystem by maximising semantic interoperability and alignment among ecosystem members, services and technologies. The knowledge base, including business know-how, assets, architectural knowledge and tooling, is required and exploited in each service engineering phase.

In summary, four main challenges for ecosystem-based service RE can be identified:

- **Service co-innovation:** The open innovation between ecosystem members to identify ideas for a service must be enabled.
- **Service value co-creation:** The co-creation of value in the value network must be enabled by utilising the ecosystem's rules, methods and practises for service engineering.
- **Enabling infrastructure:** The infrastructure with the support for service collaboration and co-operation of ecosystem members must be provided.
- **Utilisation of ecosystem's assets:** The existing ecosystem assets must be able to be utilised.

2.3 The service requirements engineering methods for ecosystems

Several surveys and reviews of the RE frameworks, approaches and methods have been concluded recently, such as

[29–33]. In [32], Service-Oriented RE and the Scenario-Based RE are defined as emerging trends in RE. Despite the research results made in the context of service-oriented models and techniques, there is still a need for new techniques and approaches for RE activities [2,4].

2.3.1 Service co-innovation

Service innovation is already taken into account in the recent research on service ecosystems. In [34], a common underlying architecture is used to connect different pieces of innovation components, and it also considers the value proposition for different participating partners. An open government data portal in [35] is also used as an open innovation platform to attract businesses and citizens to create e-services. An innovation framework introduced in [36] supports the development of new services through integrating customers, suppliers, complementors and competitors. A conceptual framework for web-service ecosystems proposed in [16] emphasises a central platform from which the companies try to extract ideas for service innovation and use these ideas to create new, or improve existing, services. The structure of the value chain affects innovation, requirements engineering performance and software success, such as described in [37]. A method introduced in [38] emphasises socio-technical aspects such as context, environment, and team management in service innovation. In [39], two types of requirement engineering methods are suggested to emerge for IT services: RE for *service consumers* and RE for *service providers*. Consumers will focus on identifying the tasks that need support, whereas providers will focus on achieving economies of scale such as offering a new service for multiple customers, or applying a specialised skill to a common problem. In several approaches, such as [29,40–43], the identification of business goals and the business processes that support those goals are used as a starting point for the RE of services. Several approaches for the requirements engineering of service users are also suggested, such as [44–46]. Both these viewpoints are required in ecosystem, since the usage goals of consumers and the business goals of service providers must be fulfilled by the services. The fact is that the current approaches and methods lack of tool support, they do not cover all the phases of RE, or they concentrate only presenting only a technique applicable in a certain RE phase [33].

2.3.2 Value co-creation

Although the significance of open innovation has been detected in the context of the ecosystem, there is not much research in the literature on how to go further from ecosystem-based service innovation to service co-creation inside an ecosystem. However, some approaches exist that deal closely with the subject. The Inter-enterprise Service

Engineering Framework [47] supports three phases of e-service development in business ecosystems: requirements analysis, service design and service implementation, assigning them to strategic, conceptual, logical and technical abstraction layers. Service requirements are identified in the strategic perspective in the form of a business model. However, the service ideas are innovated, identified and evaluated and their business potential is analysed prior to the development of the strategic perspective, but the approach does not define how this is done. In [48], an RE approach is introduced, which uses guided questionnaires to elicit the requirements coming from the current business situation, and a workshop to define the basic requirements for each Manufacturing Service Ecosystem scenario. The approach enables the relevant ecosystem actors to participate in scenario identification and requirement elicitation, but it does not define the content of RE phases and the impact of other ecosystem elements on RE. In [36], there exists a mapping of information collected in the Innovation Repository accessible to service engineering, but the approach does not describe how this information affects to service realisation.

2.3.3 Enabling infrastructure

The interoperability models and rules enable the loosely coupled services to collaborate. In [25], six interoperability levels are defined for smart environments: conceptual, behavioural, dynamic, semantic, communication and connection. In [49], four inter-related metamodels are suggested for ecosystem interoperability: domain ontology, methodology, domain reference, and knowledge management metamodels. In addition to service interoperability, pragmatic interoperability [49] is achieved between ecosystem members when their intentions, business rules and organisational policies are compatible. To detect service interoperability, the service must be specified in an understandable way in the ecosystem's service registry. Three levels of service specifications can be identified [26]. The importance of interoperability models has been recognised in the context of ecosystems; still there is a lack of application of these models in practise.

2.3.4 Utilisation of ecosystem's assets

The knowledge models are usually described using ontology models, which are guided by knowledge management. The knowledge- and ontology-based requirements engineering has a long history [50]. Even currently, an increasing amount of research has been conducted to utilise ontologies in RE [51]. Different kind of approaches have been suggested, such as for generating a requirements model based on the concepts in the service requirements modeling ontology [52] or establishing a mapping between a requirements

specification and ontological elements [53]. A knowledge-based development of intelligent smart spaces is introduced in [27], which support the service innovation and co-creation, exploiting the usage scenario description, the set of use cases that define the specific viewpoint of the usage scenario, and the reusable artefacts, such as ontologies, models, patterns and rules, provided in the knowledge base. In [25,54], an approach for developing intelligent applications/services for smart spaces is introduced that exploits the ontology models, interoperability models and context models for describing self-adaptable services. The approach is multi-technology and multi-domain oriented, but it still lacks the business (ecosystem) viewpoint.

2.3.5 Summary

We can notice that there already exist several methods and approaches that enable some parts of the ecosystem-based RE for services. Several innovation methods exist for services that enable the open innovation among ecosystem members. Several models have been introduced to verify interoperability in ecosystems. Also, support is provided for knowledge-based service engineering that would enable to utilise the existing assets of the ecosystem. However, the interoperability models and knowledge base service engineering have not been present in methods for ecosystem-based service engineering. Service innovation approaches either do not take these into account. Furthermore, the innovation approaches are separated from the further phases of service development. The identified methods and approaches for ecosystem-based service development are loose; they are only concentrating their own viewpoint and not working together. Clearly there still exists a lack of methods how to take the digital service ecosystem elements into account in service engineering, especially when they have a direct effect on RE of services. The diversity of RE approaches and their use in different contexts highlights the importance of the formal definition of terms and the definition of uniform modelling methods and techniques. The identified challenges for service RE [2–7] and for service RE in ecosystem [8,9] still remains, meaning that the service RE is not mature enough as such and not especially for digital service ecosystems. There are multiple actors, viewpoints and ecosystem capabilities that affect the service RE in an ecosystem, but at this moment, there is no RE method for digital services that take these account.

3 Scenario-based service requirements engineering in a digital service ecosystem

Based on the presented state-of-the-art analysis, we have defined a service engineering model for digital service ecosystems, concentrating on the RE of services. Ser-

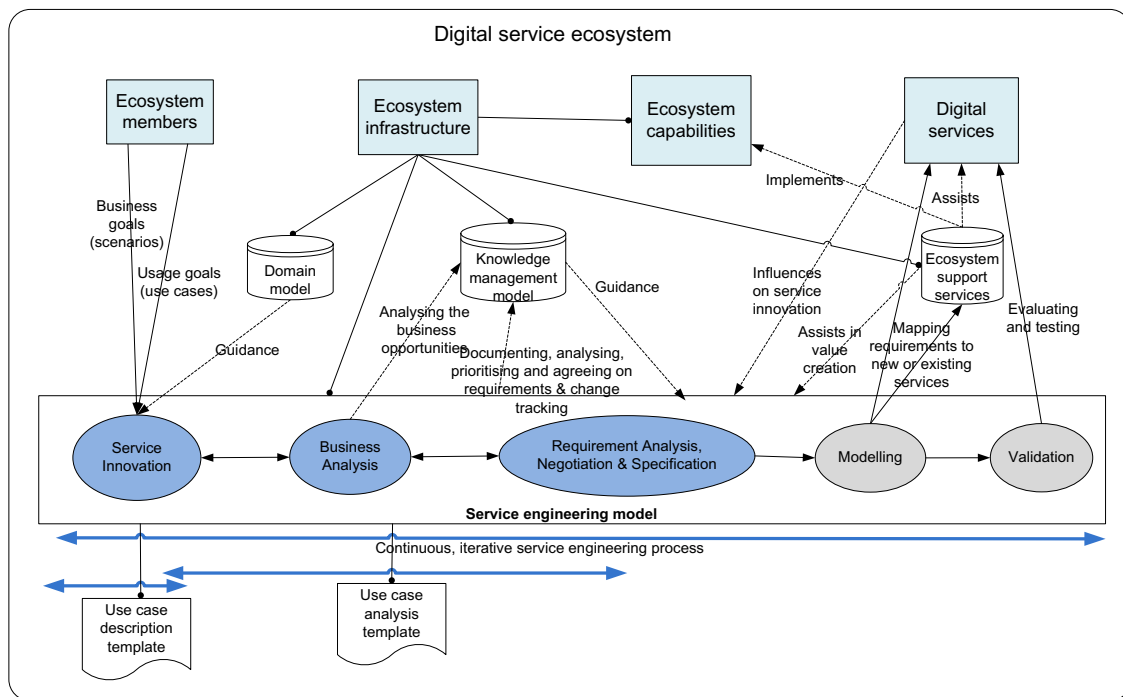


Fig. 2 The elements and phases of service RE in a digital service ecosystem

vice requirements can be classified into functional, non-functional and business requirements and constraints. Functional requirements describe the behaviour of a service that support the tasks or activities of the user. Non-functional requirements describe the qualities of the system, which can be defined as externally and internally observable properties of software systems [55]. From the non-functional requirements viewpoint, quality is regarded as constraints exhibited over the functionality of the service. Business requirements assist the service provider in achieving the business goals. Constraints are those characteristics that limit the development and use of the service.

Figure 2 describes the service RE elements and phases in a digital service ecosystem, which are introduced in more detailed in Sects. 3.1 and 3.2. The last two phases in Fig. 2, modelling and validation, are our ongoing work and are described in our upcoming paper on service architecture design and are therefore out of the scope of this paper.

3.1 Digital service ecosystem elements

3.1.1 Ecosystem members

The members of the digital service ecosystem can be defined to include service providers, service brokers, service consumers and infrastructure providers. Service consumers use the services and define the usage goals for the services, i.e. tasks that need support. They may also report on problems

and failures in the service usage and provide feedback for the service validation. Service providers are independent members that provide digital services to be used by other ecosystem members or consumers. Service brokers promote the services, enable service delivery and match the demand with the best available services. Infrastructure providers provide services that implement the purpose and capabilities of the ecosystem, such as establishing, modifying, monitoring and terminating collaborations, and operations for joining and leaving collaborations. The ecosystem can also include other infrastructure provider roles, such as service market-place providers, tool providers, cloud service providers and interface providers [56]. The ecosystem provider is usually the key organisation, which establishes and maintains the ecosystem, controlling its function, members and services.

3.1.2 Ecosystem capabilities

The ecosystem capabilities describe the capability model that defines the properties of the ecosystem, and how these are implemented using the ecosystem services that the ecosystem infrastructure provides. The capabilities define the purpose of the ecosystem, its ability to perform actions and the rules of how to operate in the ecosystem. The capabilities define the governance activities and regulation processes for the ecosystem for directing, monitoring and managing the ecosystem. These include, for example, how a trusted collaboration can be established between members, what the interaction rules

are, how to join and leave the ecosystem, and how to describe and deliver services. In addition, the ecosystem capabilities define how the knowledge is managed.

3.1.3 Ecosystem infrastructure

Infrastructure implements ecosystem capabilities, supporting the utilisation of core competencies and core assets, flexible business networking, and efficient business decision-making. The infrastructure also includes the taxonomy and shared descriptions of services (categorised by domain, purpose and technology etc.). The infrastructure provides the following models and assets that assist in the RE:

- *The domain model* describes the concepts of the domain and the relationships between those concepts. The domain model can be used for configuring and adapting service artefacts for use in other domains. Thus, it supports evolution of the service ecosystem.
- *Knowledge management model* enables reuse of the existing knowledge on the business, and design practices and existing assets in the development of new service, maximising semantic interoperability and alignment among ecosystem members, services and technologies. For example, quality ontologies define the concepts, relations, rules and their instances, which comprise the relevant knowledge of a topic and assists in reaching quality requirements, and quality-driven design methods enable to achieve the quality requirements in the architecture.
- *The service engineering model* describes and guides how the services are being engineered in service ecosystem, assisting in innovating, analysing, modelling and documenting requirements. The scenario-based RE technique is chosen, because it enables to describe both of the viewpoints of RE: business and usage. The main activities are supported by the two templates: The Use Case Description template is used for service innovation and The Use Case Analysis template assists in identifying, analysing and specifying the requirements.
- *Ecosystem support services* are responsible for providing tool support for the activities of the service engineering, e.g. for using quality ontologies in all development phases (design, implementation, and testing). In addition, support services assist its members in value creation, for example, by contract making, finding partners, services, and/or markets, and analysing business models [56]. The infrastructure should also provide collaboration support services for ecosystem members, e.g. for communication and document sharing. The ecosystem should be aware of the quality of the services in the ecosystem's service registry; therefore, the ecosystem should also include support services for run-time quality monitoring and analy-

sis of services [25,30]. Furthermore, the service registry should provide feedback mechanisms for users to provide feedback about the service, thus supporting requirements change and evolution.

3.1.4 Digital services

In a digital service ecosystem, digital services are provided by independent ecosystem members, where they provide additional value for both service consumers and other service providers. Service providers do not necessarily provide a complete service for consumers but can just provide part of a composite service [57]. Service level agreements (SLAs) are negotiated between the atomic service providers and the composite service provider, describing the agreed-upon terms with respect to quality of service and other related concerns. The results of the RE process, i.e. service requirements, either result in new digital services, or they are mapped to existing digital services. The requirements can also be identified as new ecosystem support services, or they can cause changes to the existing ones.

3.2 Service RE phases

Service RE in a digital service ecosystem is an iterative process consisting of three phases. These are described in the following sub-sections. Before the RE can begin, the following activities need to be performed inside the digital service ecosystem:

- **Identifying the value networks:** The actors that have their interest in the service are identified and their contributions in the value network are defined.
- **Identifying roles in RE:** The roles and responsibilities in service co-innovation and co-creation are defined for each member considering all activities in RE.

3.2.1 Service innovation

The service innovation phase starts the service RE in digital service ecosystem. The main purpose is to identify the ideas for new services, scope and analyse them and transform them into service requirements. The innovation can be divided into two sub-phases: requirements elicitation and the requirements identification of services.

Requirements elicitation

Purpose: Requirements elicitation is a practice of obtaining the requirements from all stakeholders. The requirement elicitation method defines what, how and from whom the requirements should be elicited, and guides the elicitation process.

Activities: The elicitation process can be divided into the following steps:

- (i) Identifying responsibilities: The members responsible for the activities of the elicitation process, such as coordination, requirements collecting and management, are identified.
- (ii) Identifying ecosystem assets: The domain model(s) of the service ecosystem assist(s) in understanding the relevant domain concepts that affect the service RE. The knowledge management model provides knowledge, know-how and assets, whereas the capability model describes the ecosystem support services, tools and guidance for RE.
- (iii) Identifying requirements sources: service provider identifies requirements from the business goals: service consumers provide ideas, requirements or feedback for services, service broker helps in service delivery and mediates between service providers and consumers; and ecosystem infrastructure defines domain ontologies, knowledge management models, etc.
- (iv) Analysing stakeholders: This analysis extends the stakeholders from business, user, technical and management points of view. The analyst shall have a clear vision and understanding of what kinds of stakeholders are to be covered and what are to be scoped out.
- (v) Introducing the approach and tools: The approach used for a service RE is a scenario and use case description-based elicitation approach. Business scenarios are described by the service provider that wants to create value and achieve economic returns with the help of the service. The scenarios are refined into use cases that describe the user's point of view. Several UML-compliant tools enable the definition of use case diagrams.
- (vi) Eliciting the requirements: The Use Case Description Template is a Microsoft Word document template that assists in identifying and documenting the motivation of the use case, and inspecting and documenting the use case from the following viewpoints: contextual, functional, non-functional, business and constraints and threats.

Practice: The domain model(s) describes the relevant domain concepts that should be considered in the RE. The knowledge management model provides ontologies, methods, tools, templates and guidelines for requirement elicitation. The Use Case Description Template ensures that all the use cases are described adequately, consistently and with the same accuracy. The use of the template is to make it easier to transform from use cases to identifying services and their relationships. Functional properties can be identified from the use case flow, which is a detailed description of the user actions and system responses during the execution of the use case. All the

elements of the use case, i.e. actors, the use case function, actor-use case relations and the use case environment, should also be considered from the non-functional viewpoint, which describes the quality properties of the use case. For example, reliability properties describe the issues affecting the failure free operation of the use case (how the fault prevention, tolerance, removal and recovery from failures are considered in the use case), and availability properties describe the issues ensuring that the use case function is ready for use when required. The Use Case Description Template includes the classification of quality properties and their detailed description to assist users to consider each quality property.

Outcome: The outcome of the phase consists of the following for each use case: *General information:* the identification, introduction and rationale of the use case, *Contextual settings:* A description of the context of the use case, including actors, a vision of the infrastructure, the physical resources and required artefacts, *Functional and non-functional description:* the description of the main functions and the quality properties of the use case, *Business properties:* a description of business properties or the use case, such as customer segments, value proposition, customer relationships, etc., *Constraints:* a description of the constraints associated with the elements of the use case, and *Threats and exceptions:* a description of the threats for the use case, responds to those threats, and description of anticipated exceptions and error conditions.

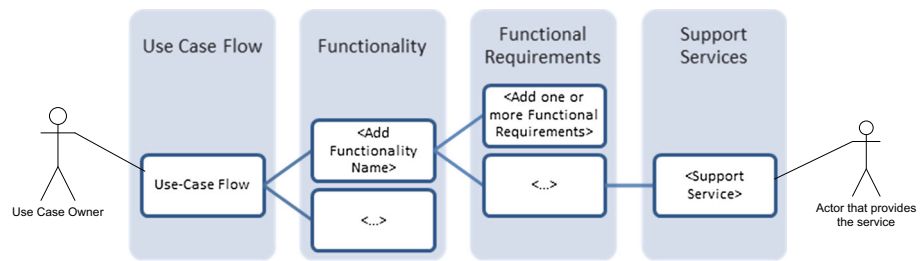
Service requirements identification

Purpose: The purpose of the requirement identification phase is to identify, classify, merge and prioritise service requirements using the use cases defined in the previous phase as input.

Activities: Each member identifies one or more functionalities for the use case, and identifies and maps the functional requirements to each functionality.

Practice: The Use Case Analysis template assists ecosystem members in this activity, using as input the information gathered in the previous phase. The template enables to analyse the use cases from business and user points of view and to identify and describe functional and non-functional requirements and constraints. The template also assists and guides in making an initial mapping between the identified requirements and the existing digital services and potential new required ones. Figure 3 describes the mapping with the Microsoft Word smart-art tree diagram provided in the template. The owner of the use case plays a key role in mapping the identified requirements to the support services provided by the use case partners involved in this particular use case. Each functional requirement results in one or more 'support services' that implement the requirement. The support

Fig. 3 Service requirements identification and mapping requirements to services



service is a new or existing that provides the functionality and quality that is required to implement the use case. If the requirements cannot be mapped to already existing services, totally new services are identified here, which are responsible for implementing the requirements. When candidate services have been identified, non-functional requirements can be mapped to them. This especially concerns the execution qualities, such as performance, availability, reliability and security, but evolution qualities (such as reusability, modifiability and maintainability) are focused on later on in the service modelling phase, where the architectural knowledge guides the design work.

Outcome: The outcome of this phase includes a detailed description of supporting services identified in use case analysis, including functional requirements, non-functional requirements, data resources, constraints and their mappings to the identified support services.

3.2.2 Business analysis

Purpose: The goal of this phase is to identify which use cases have the most business potential in the ecosystem, i.e. the business analysis helps to define what requirements to implement, and how.

Activities: The identified use cases are collected together in the ecosystem and the members responsible perform the business analysis. Therefore, each use case is analysed by several analysts according to the following criteria (defined in the knowledge management model of the ecosystem) [27]:

1. Maximum business impact (rated: 1–10)

- Added value: The usefulness to customer is higher than adoption effort and costs;
- Partners interest: The business opportunities for ecosystem members;
- Market penetration: The time period the solution can be brought to the market.

2. Fast and low-risk realisation (rated: 1–10)

- Availability of technology: compatible with and builds on popular legacy technologies and assets;

- Implementation complexity: nature and amount of R&D effort is known and feasible.

The given numbers can be weighted to be appropriate for the case. After the business potential analysis, the most relevant use cases from the business viewpoint are identified, and the initial mapping of the identified requirements of these use cases to the responsible services (according to Fig. 3) are verified. At this point, the actors that provide the required support services (see Fig. 3) are identified. Some of the requirements can be implemented by already existing services, when contracts are being made with other service providers, whereas some of the requirements result in identifying new services.

Practice: The knowledge base of the digital service ecosystem contains business knowledge, architectural knowledge, assets and supporting knowledge management tools [27]. Business knowledge also provides information about the earlier ideas that have been evaluated but not realised. The reasons behind the earlier decision are valuable in making feasibility checks while further defining the usage scenario and use cases in hand. The knowledge base also includes documentation about existing assets that resolve their availability, suitability and quality for the service development in hand. The architectural knowledge is used to estimate what artefacts can be used as such, or modified, what is the quality of the artefacts and whether the quality is satisfactory.

Outcome: The outcome of this phase is a set of use cases, the related requirements and the analysis results on business impact and risks related to the use cases.

3.2.3 Requirements analysis, negotiation and specification

The last phase of the service RE includes three sub-phases that are highly interrelated and iterative by nature. The main purpose of the phase is to provide a complete requirement specification of the needed services that is used as input in service architecture modelling.

Service requirements analysis

Purpose: The goal of requirement analysis is to determine the consistency and completeness of requirements, and also

the priority of requirements. The purpose is to analyse the capabilities and constraints of the existing services, different potential technologies for service creation and the contribution of the service to the different business cases defined in the earlier phase.

Activities: The starting point of this phase is the existing service architecture and/or description of services, or a sketch of new service architecture. The candidate services for the use case identified during service requirement identification and business analysis are listed and checked with partners. The similar services are merged and those with no business potential are rejected. The taxonomy and shared descriptions of services provided by ecosystem infrastructure enables to categorise services by technology, domain and purpose. The classification according to taxonomy assists in getting better understanding about requirements. For each service, the requirements are analysed, and combined if necessary. As a result, two different services might be required for different users. Thus, a variability analysis is made from four viewpoints: the service provider, the service user, the usage context and the implementation technology. The trade-off analysis is performed for conflicting requirements according to their importance and the results of the business analysis. Also, quality requirements are prioritised based on their importance to stakeholders and as a result of the trade-off analysis.

Practice: The knowledge base includes rules for trade-off analysis, e.g. the rules for ranking quality attributes. Each quality attribute is a representation of a single aspect or construct of a quality. The Use Case Analysis template enables the detection of the requirements mapped to each service, and the importance for each requirement.

Outcome: The updated architecture/vision of the services, and the analysed and prioritised quality requirements for each service.

Service requirements negotiation

Purpose: The purpose of requirement negotiation is to communicate the service requirements to the business and technical stakeholders involved in service development and agree the service requirement specification with the ecosystem members. This means active collaboration among the members of the digital service ecosystem by exploiting the ecosystem infrastructure.

Activities: The starting point of this phase is the analysed service requirements description that gives the first proposal of the balanced service requirements. E-mail and collaboration support services (e.g. document share points, co-design tools, video conferences and telecommunications) can be used for communication and negotiation between ecosystem members. Voting is rarely needed for design decisions, but it is

also possible to be organised by using the digital ecosystem support services. Typically, several negotiations are carried out with different focal points: e.g. big picture, domain variations and business and implementation constraints. That is why several rounds of the analysis-negotiation-specification loop are required.

Practice: The domain model assists in getting common understanding. The knowledge model assists in sharing evolving knowledge and specifications. The knowledge base includes guidelines on how the service negotiation is to be carried out and how the design decisions are to be documented and recorded for future needs. Due to service evolution, special attention is to be paid to documenting the agreed design decisions, the proposals that have been discarded and the reasons behind the decisions. The ecosystem support services (automated or practical guides) assisting in negotiation among the ecosystem members. The Use Case Analysis template can be used to document the design decisions but tools that provide automation support are preferred.

Outcome: Negotiated service requirements and a list of agreed and rejected service requirements and the design rationale behind the decisions.

Service requirements specification

Purpose: The purpose of this sub-phase is to describe the service requirements specification by using textual and graphical notations that make requirements specifications complete, understandable and useful for all ecosystem members. The requirements specification is a complete description of the behaviour of the services to be developed.

Activities: Several rotations are required to illustrate domain requirements, business requirements and technical requirements for services. The first round ends with an initial service taxonomy that defines what kind of digital services are needed and clusters them to the groups of services based on their usage or/and technical relations. The last round ends with service specifications that provide a big picture as a set of master use cases that describe the behavioural service architecture. A master use case is made by integrating the related use cases defined by different business actors (see Fig. 3, use case owner). The master use cases give a mutual understanding of the service architecture and how it is used for realising the different use cases by diverse actors. Thus, the service requirements specification includes the activities that transform informal service specifications into semi-formal descriptions to be used as a starting point in the service architecture modelling phase.

Practice: The Use Case Analysis Template assists in describing the requirements in a consistent, accurate way. The

Table 1 Preliminary set of digital services in the ICARE ecosystem

Acronym	Description
PRM	<i>The Processing Resource Manager</i> is in charge of content ingest and cloud resources management (e.g. load balancing)
BWM	<i>The BandWidth Manager</i> regulates the networks according to traffic overload and user requests
SCM	<i>The Security Content Manager</i> controls the networks to get a good QoS and guarantees that content is delivered at the right place at the right time
CDM	<i>The Content Database Manager</i> can be used for publishing and retrieving content. It knows content properties in the cloud infrastructure and can retrieve them for play-out delivery
MAM	<i>The Media Asset Manager (MAM)</i> and its compounds (i) handle the descriptive metadata that are delivered along with the Media assets, (ii) manage the work orders for traffic managers and video engineers, and (iii) manage the deep archive, the transcoding and the delivery

description of the service taxonomy and services is guided by the knowledge management model.

Outcome: A service taxonomy and a set of master use cases that describe how the digital services—to be developed and existing ones—interact and cooperate with each other in order to provide the required end-to-end digital services. The updated architecture/vision of the digital services, and the analysed, prioritised and agreed requirements for each service.

Since the service engineering model is iterative (described as double-headed arrow in Fig. 2), it takes into account the requirements evolution caused by new requirements, feedback achieved from the use of the service, and change requests in current requirements. The identified new or changed requirements proceed from the service innovation to the service identification, where the requirements are mapped to the existing digital services and potential new ones. The requirements go normally through the business analysis and requirements analysis, negotiation and specification. It is the responsibility of the modelling phase (out of scope of this paper) to decide how to implement the new and changed requirements.

4 Example of using the digital service RE method

So far, the RE method has been in use in two different cases. This section describes the first usage of the service RE method, when it was applied to specifying the digital services and related support services for an interactive multi-screen TV services ecosystem in the Innovative Cloud Architecture for Real Entertainment (ITEA2-ICARE) project.¹ This digital service ecosystem includes 25 service ecosystem members from Europe providing and using digital cloud-based services related to the operation of end-to-end interactive multi-screen TV services. The ecosystem member roles include, for example, multi-modal content service

providers, communication infrastructure service providers, cloud platform service providers and consumer cloud service providers. The ecosystem services include content processing, multi-channel user interaction and content access management services, which are all needed as part of operation in the final end-to-end TV service provided to end users.

The goal for applying the RE method was to collect and analyse requirements from the ecosystem members towards a shared service-oriented platform enabling the provisioning, integration and use of services amongst the members of the ecosystem. An additional aim was to support ecosystem members in specifying their digital service offerings and needs via describing use cases with business model analysis within the digital service ecosystem from the viewpoint of an individual ecosystem member. In the early phase of the ICARE project preparation, the domain model of cloud-based multi-media services was sketched and analysed by the collaboration partners. As a result, a preliminary set of digital services for the ICARE ecosystem was proposed (Table 1). These service descriptions are abstract and intended to give a mutual understanding the purpose and ability of the ICARE ecosystem. These services are covered by the existing services and new services provided by the ecosystem members.

4.1 Service innovation

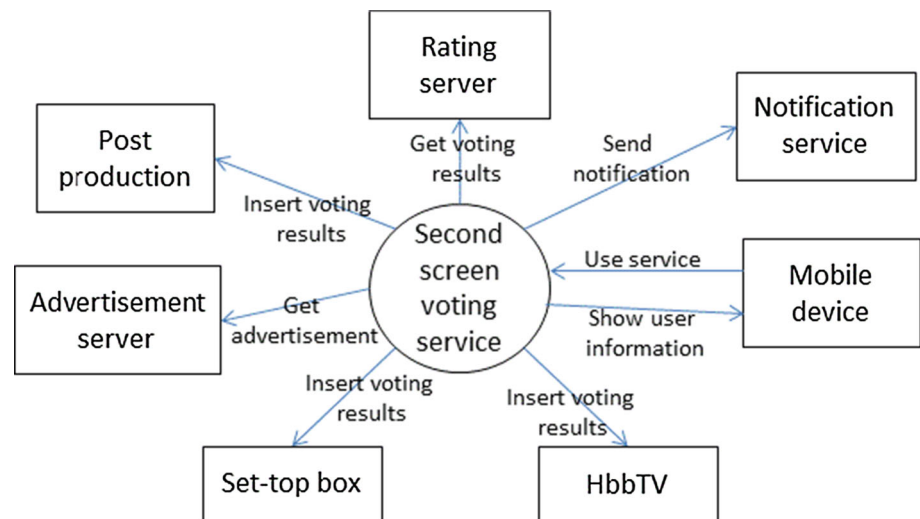
4.1.1 Requirement elicitation

The Use Case Description Template in the form of Microsoft Word-file was used for collecting input from the ICARE ecosystem members. Thus, the usage of the template did not require special tools. In addition, guidance was provided in the form of instructions for using the templates. The instruction included the definition of the main elements, description of the purpose and goal of the Use Case Description and common instructions for fulfilling the templates. Both business and technical experts were asked to be involved in defining use cases. However, only one contact point was defined by

¹ <https://itea3.org/project/icare.html>.

Table 2 General information of the second-screen voting use case

Summary	This use cases defines how mobile phone-based services can be linked with live TV broadcast. An end user has a mobile application related to a TV programme. The user gets some insight into the upcoming shows with the application. Also during the live TV programme interactive voting services will be provided to an end user. This interactive voting service contains advertisements that can be personalised
Rationale	People are watching less and less live TV programmes. This can be problematic from the advertiser and broadcaster point of view. Providing new ways for people to be committed to live TV programmes can make advertising more efficient. Also the user commitment to TV programmes can be increased. The user will spend more time with using TV and the additional services
Description	<ol style="list-style-type: none"> 1. Alice installs the Talent application from the programme's web page 2. Alice uses the application to get some insight into the upcoming shows before the live show 3. Alice is watching the live Talent programme 4. After the programme, a competitor notification is sent to all watchers who have installed the application 5. The notification includes a voting form in which the user can give 1–5 star rating to the current competitor. The notification also contains ads which can be personalised 6. Alice saves the discount coupon for her favourite store shop, which is included in the notification 7. After saving the coupon, Alice rates the programme and presses the submit button 8. On the live TV programme, the average user ratings are shown

Fig. 4 Second-screen voting use case context diagram

each partner, and it was not visible to other ecosystem members which kind of expertise was used to define the use cases.

All in all 41 use cases were defined. Some of them were variations on a similar theme, the actual number of different use cases being about 30. Filling a use case took about a week by each partner, but it took 3 months to collect all of these use case descriptions because of the participants' work schedules and the need to iterate the use cases. The activities of the ecosystem members were divided according to work package descriptions of the project; thus, one partner was responsible for collecting and coordinating the use cases. To illustrate the outcome of the Service Innovation phase, the definition of the 'second-screen voting' service

defined by an ecosystem member is used here as a practical example.

General information: ICARE UC No. 4 second-screen voting

The general information of the use case is represented in Table 2

Contextual settings

A context diagram Figure 4 describes the context diagram of the second-screen voting use case.

Actors: The identified use case actors and their responsibilities included the following: End user—uses an interactive

TV service via a mobile device application; Advertiser—provides advertisements to the broadcaster to be delivered to the end users; and Broadcaster—provides additional information to show based on the user interaction and provides an interactive service to end user.

Resources: The number of needs of physical resources and locations is more or less an implementation-specific issue. However, some parts of the system, such as software components for a set-top box and HbbTV will be in the home environment running on previously mentioned devices. The notification service and the rating service can run in the same server. Advertisements are most likely provided by third-party actors from their own servers. In addition, a post-production server must be in its own physical environment.

Frequency of use: The frequency of the usage depends on the content and users. Average usage could be 5–8 voting (notifications) during a one-hour programme. If the user base is large, there could be 10–100 thousand simultaneous notification and voting results to be handled.

Description of service properties

The Use Case Description template assisted in describing the use case, and the service innovation phase in the case of the ‘second-screen voting’ service resulted in the service description represented in Table 3.

4.1.2 Service requirement identification

After all use cases were identified and described, each partner identified the functional requirements of the use case, starting from normal flows using Microsoft Word’s smart-art tree-based diagram provided in the Use Case Analysis template. The running ID of each identified requirement and potential support services are also shown in the diagram. In the example diagram in Fig. 5, the use case owner identified two new services to be implemented and also potential for utilising partner-provided or existing technologies to complete the use case. The requirements shown in the diagram were then specified in more detail. The use case number and use case-specific requirement ID was used as the global ID for each requirement. The detailed description is provided in Table 4. The importance (imp) of the requirement ranges from 1 to 5, (1 = not relevant). The template also provided the possibility to define non-functional requirements and constraints, and associate them with functional requirements. For example, two availability requirements were associated with the functional requirements 4.1 and 4.2 (see Table 4). The category (cat) describes the type of the requirement (F = functional, NF = non-functional, C = constraint). The table was scalable and could be complemented in more detailed afterwards; the ‘Details’ column allowed to insert more information.

Several data resources were also identified: TV programme info, Notification content, Voting results and User profiles. These are specified in the data resource definition template with characteristics of resource name, description, related requirement(s), standards, quantity and size, privacy and details.

4.2 Business analysis

Table 5 gives an example how the business impact and risk analysis results could be collected and compared in order to make decisions on how to proceed and with what use cases. Both the business impact and risk analysis were rated in a range of 1–10. As can be seen, the development should be started from UC No. 1 if maximum impact with low risk (i.e. availability of technology and implementation complexity) is preferred. Also, it can be seen that UC No. 5 is not realistic and should be rejected.

According to Fig. 5, the identified potential for utilising partner-provided services to complete the use case enabled the finding of co-operation partners with a similar focus.

4.3 Requirements analysis, negotiation and specification

4.3.1 Requirement analysis

The services and service providers identified during service requirement identification and business analysis were collected into a list preserving the link to the original use case. The list of candidate services was checked with partners, and similar services with different naming merged and those with no business potential rejected. At this stage, the service candidates were quite heterogeneous both in scale and conceptual level. A service candidate comprised functionalities from algorithms and functions to sub-systems consisting of several servers. In order to better grasp the overall requirements for the ecosystem, the services were classified into service taxonomy based on their technology position. The services were classified into two dimensions: cloud services vs home network services and infrastructure vs end-user services. A partner interested in providing a service candidate could then check the requirements set to it and similar services tracing their links to the use case descriptions. Based on analysis of the business models presented in the use case descriptions, the shared requirements were also identified (see Table 6).

Several of these requirements could be assigned to individual use cases or service candidates. For example, the cross-distribution platform interoperability could be linked to interactive TV directly supporting different distribution platforms, and also to multi-screen interaction that can cope with emerging technologies helping the user with access to different mobile devices, advanced interaction technologies and multiple screens at home.

Table 3 Description of the service properties

<i>Functional properties</i>	
Preconditions and assumptions	End user must have installed the mobile application on his/her mobile phone
Trigger	Live programme starts and there is programme that requires voting
Normal flow	Users receive a voting notification. The user notices the ad in the notifications and clicks/saves it. The users vote. The end results are visible in the live programme.
Post-conditions	System contains the voting results
<i>Non-functional properties</i>	
Reliability	The user must have a working data connection on his/her mobile device in order to vote
Availability	All services must be up and running and linked. The live programme must be broadcasted
Performance	Sending several parallel notifications and processing the voting results are the bottle necks of the system. This means that there must be scalable hardware resources for system-critical services
Security	The application on the mobile device must be registered with a notification service. Additional information about the user or device (IMEI) can be added on the registration request
Interoperability	Services provide HTTP APIs for IOP usage
Adaptability	The notification service provides different kinds of notification types for different usage. Showing the result can be done in many ways
Variability	The content of the pushed notification can be varied based on the situation
Scalability	The notification and voting services can be implemented as cloud services
Personalisation	The mobile device needs a personal account (e.g. Google account) which is needed for registering with the service
<i>Business properties</i>	
Customer segment	Common people who watch TV. Probably younger ones
Value proposition	Users get some dedicated information about the programme which they cannot get elsewhere. Users can also vote for free. They can get some discount coupons
Channels	Application delivery will be done via mobile marketplaces. Marketing and promoting of the service is done during the programme and on the websites of the programme
Customer relationship	Users are committed to watching the programme interactively and have the possibility to have influence. Also, using discount coupons as an advertisement will keep up the user's interest
Revenue streams	Ad-based revenue is the main source
Key resources	Mobile device, notification and voting services
Key activities	Voting and advertisements receiving
Key partnerships	Broadcasters with advertisers
Cost structure	Almost everything can be automated. Only selling and linking the ads needs some interaction
<i>Constraints, threats and exceptions</i>	
Location	Mobile device (application) needs working data connection
Misuse cases	Someone might want to distort the voting results by accessing the voting server directly. However, the voting server can monitor the clients that are accessing it and prevent phony connections from non-application sources
Exceptions	Problems in the data connection can cause distortion of voting results. The voting server can conduct based on the registered clients and number of voters if there is no reason to show results. This means that no voting results are shown

Table 3 continued

Other relevant information

This application-based solution is a direct competitor for SMS voting. From the user point of view, this is a preferable solution because the voting is free and much more convenient. From a broadcast company point of view, the SMS are good source of revenue. This means that, in the application-based solution revenue must come from advertisements. It is also good to bear in mind that both solutions could be used together. SMS could be used by occasional watchers and the application-based solution is targeted at the fans of the programme

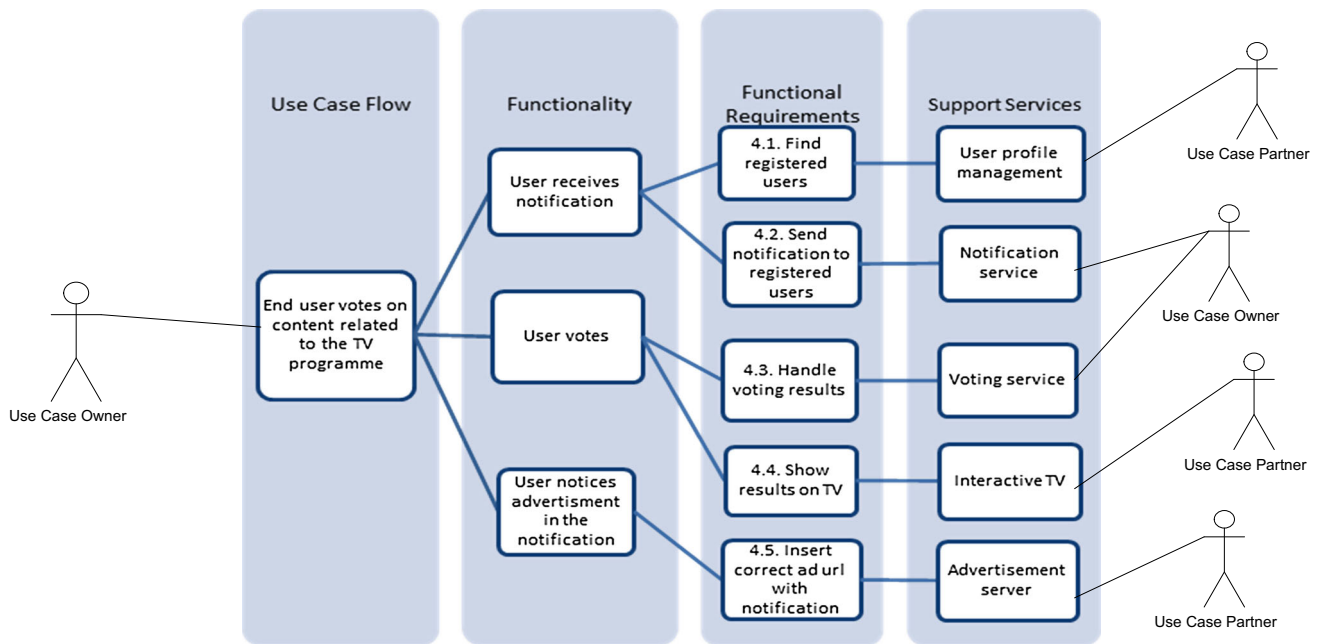


Fig. 5 Functional requirements analysis for UC no. 4 second-screen voting use case

Table 4 An example of identified use case related requirements of the Second-screen voting service

ID/req.	Imp	Description	Details	Cat
4.1	4	Find registered users to watch the programme	Input: programme name Output: List of users	F
4.2	4	Send a voting notification to registered users	Input : List of users	F
4.1, 4.2	4	Users must be registered	Rationale: If users are not registered, the notification cannot be sent Classification: Availability	NF
4.2	5	Advertisement have to be mapped with notifications	Rationale: Advertisement has to be mapped with the right notifications Classification: Availability	NF

4.3.2 Requirements negotiation

The requirements were negotiated in smaller groups of four to five partners with a similar focus and potentially compatible business models. An overall master scenario (i.e. a

‘big picture’) between those partners was drafted and iterated in workshops and details refined through e-mail discussions. The master scenario draft (Fig. 6) shows the technical deployment of services and partner technologies and candidate services. This work resulted in changes in the ideas

Table 5 A fragment of the business analysis results

Scenario	Business impact	Availability of technology	Implementation complexity	Market penetration	Weighted sum
ICARE UC No. 1	6	2015	3	6	17
ICARE UC No. 2	7	2016	5	6	15
ICARE UC No. 4	6	2014	6	6	15
ICARE UC No. 5	6	2016	8	4	9

Table 6 An example of common business requirements derived from business model(s) and use case deliverables

Requirements	Description
Community building-related services (wiki/social /support)	These services will allow customers of the platform to share experience, receive advertisements about new features of services and content, ask for support as well as to facilitate the management of the platform. Support means for the content creation community building may needed as well
A cross-cloud/service infrastructure interoperability	The ability of the platform to utilise the services (computing, storage, support, etc.) from various cloud providers and service infrastructures
A cross-distribution platform’s interoperability	Targeting different distribution platforms (traditional broadcasting, HbbTv, VOD, etc.)
Multiply the SLA options available for content /platform services consumers	SLA options can support different models of content distribution depending on customer profile and content type. Revenue streams related to content distribution can be based on various fee models (usage fee, subscription fee, etc.)

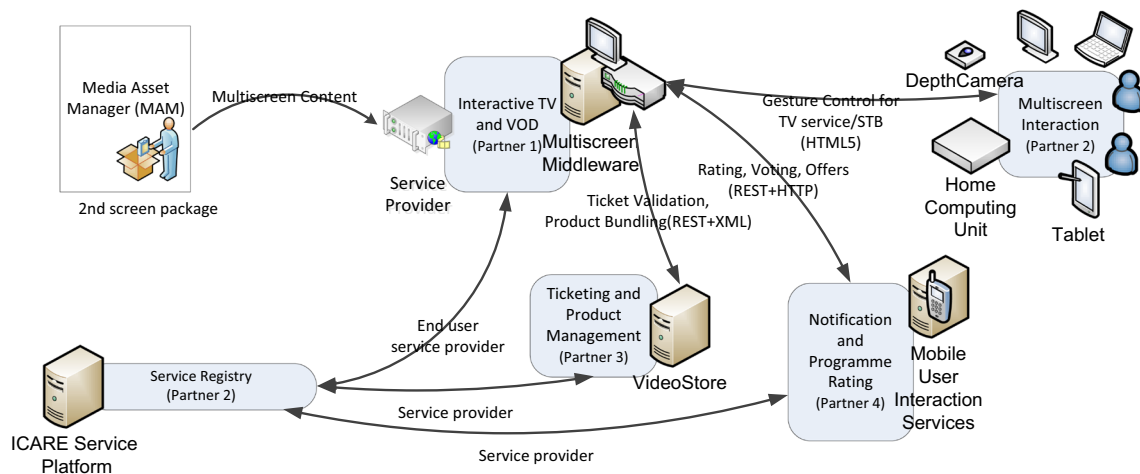


Fig. 6 The big picture describing how partners’ services and technologies are related

presented by each partner in the original use cases because of better understanding of the available ecosystem services provided by the other partners. For example, there was no need for sending notification individually to the watcher device. Instead a ‘red button’ indication could be inserted into an interactive TV broadcast stream that a user watching the programme could select either directly on TV remote or based on advanced gesture detection. The work was documented as more or less informal architectural drawings and textual

documents shared using e-mail and the master scenario was iterated until each partner was satisfied with their role in the master scenario and the support to be utilised from other partners.

4.3.3 Requirements specification

The scope of the master use case was clarified according to the master scenario and details were defined using the use

Table 7 An example of the identified functional requirements of services

Service name	Description	Service provider	Technology	
Notification	Notification to the end user of events	Neusoft	REST	
Req. ID	Importance	Description	Details	
			Category	
5	5	Interactive TV Service or Multi-screen Interaction sends to the end-user device a notification of the rating	Input: User, notification content (rating link in this case) Output: Notification to the user device	F
6	5	Interactive TV Service sends a notification of the reward to the end-user device	Input: User, notification content (Movie rental ticket in this case) Output: Notification to user device	F

case description template with the exception that now each service provider focused only on the parts (business models and functionalities) relating to their own services. The master use case focused on the services providing interactive TV content related to the use of secondary devices. The rationale was that while use of secondary devices in conjunction with traditional broadcast TV consumption is becoming more popular, there are no methods available for personalised second-screen content that is introduced in line with the traditional content. The master use case was coordinated mainly by two active services—Interactive TV and multi-screen interaction—and consisted of several supporting services integrated with the service registry. The service dependency matrix was added to the template to aid the requirements specification. The matrix included provider services that respond to the services that implement the requirements and the dependent services that set the requirements to the provider services. The related requirements were referred with the ID number of requirements (set by dependent services). The requirements were then grouped for each service and refined as presented in Table 7. The information represented in ‘details’ section in Table 7 depends on the category of requirement; for functional requirements, the inputs and outputs are defined. Each partner then continued the detailed service design from there on.

4.4 Case summary

All in all, valid results were achieved in the ICARE project using the RE method. Altogether nearly 275 requirements were identified, including functional, non-functional and business requirements, and constraints. Especially in the case of the functional requirements, all the phases of the method could be performed according to guidance. The non-functional requirements, however, were seen more problematic. The definition of non-functional requirements was easy for those who had experience dealing with them, but it was clear that strong understanding about the quality issues were

required. Although the templates included the description of quality attributes, that was not enough to support their use. Furthermore, the non-functional issues could be inspected from two different viewpoints, which also caused confusion, service provider’s viewpoint vs. service consumer viewpoint. Different non-functional requirements were identified from these viewpoints, and the templates did not specify how to handle them. It is clear that in the case of the non-functional requirements, more support is required from the ecosystem.

5 Lessons learnt

5.1 Experiences of the usage of the method

The RE method has been applied in two different cases: in the Innovative Cloud Architecture for Real Entertainment (ITEA2-ICARE) project² and in the Connecting Digital Cities (EU-EIT-CDC) project.³ To validate the usage of the service RE method in practice, we performed a feedback collection among the partners that were involved in the requirement engineering in the ICARE and CDC projects. The purpose was to receive user experiences and opinions about the method and to find out its advantages, shortcomings and development targets. The feedback collection was implemented using a web-based questionnaire that was accessible through a web page to the project partners that filled the templates. The questionnaire was implemented in April 2014 and November 2014. Altogether, we received 15 completed questionnaires. The next sub-sections describe how the users experienced the RE method.

5.1.1 The characteristics of the respondents

A total of 67 % of the people that completed the questionnaire were R&D personnel, and the rest were equally divided

² <https://itea3.org/project/icare.html>.

³ EIT ICT Labs project No. 14465.

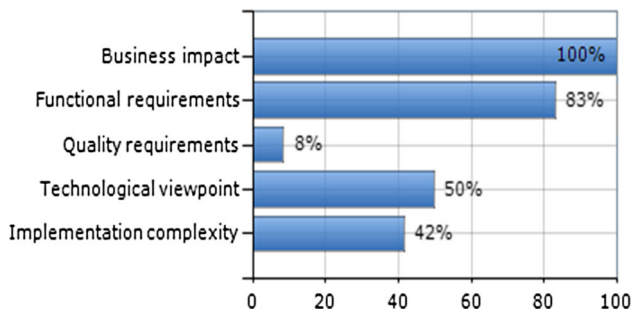


Fig. 7 The analysed viewpoints of the use cases

between business developers and project managers. There were project managers, work package leaders, task leaders, coordinators and developers among the respondents. Five of the respondents felt that their company was a part of a service ecosystem, acting as a service provider, a technology provider, or as a GUI and platform provider. A total of 40 % of the respondents confirmed that their company utilises third-party services/technologies in software development. The target of the requirements engineering was clear enough in the project for almost all of the respondents. According to one respondent, the purpose was not explained clearly enough. One respondent felt that more information was required for focusing the scope of the project.

5.1.2 Use case definition and analysis

The definition of the business scenario was considered easy among 80 % of the respondents. The use cases were defined and analysed by architects, business managers and technical experts. The technical experts had the clear majority. Only in one case was the customer of a company involved in requirements identification (through product managers). In one case also the marketing personnel participated in use case definition. The definition of use cases from the scenario was considered easy or very easy among 93 % of the respondents. The use cases were analysed mostly from the viewpoints of business impact and functional requirements. Some also considered the quality requirements, technological viewpoint and implementation complexity viewpoints. The results are illustrated in Fig. 7. 73 % of the respondents considered the requirements easy to define from the use cases.

5.1.3 Service identification

The service identification was performed mostly by technical experts with the assistance of architects and business managers. In one case, the marketing personnel participated in the service identification phase. 73 % of the respondents considered it easy to define the services that are needed for the defined use cases. A total of 33 % of the respondents

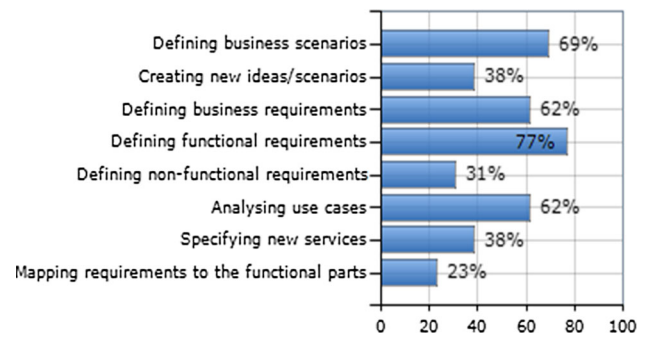


Fig. 8 The phases in which the templates were useful

exploited existing services in defining a new service. One respondent revealed that they do not exploit existing services because they try to be innovative. The analysis and prioritisation of service requirements was seen as easy by 66 % of the respondents. The respondent companies assumed mostly cloud or platform architectures, when analysing the use cases and identifying services. The mapping of functional requirements to the architecture was considered easy for most of the respondents. Some respondents considered it difficult since the architecture was lacking high-level building blocks. The non-functional requirements were difficult to define and map to the architectural elements due to the following reasons: The vision of architecture was too complex with too many small components (lacking high-level blocks), the target and output of non-functional requirements were not clear enough, and no tool exists to support this. Almost all of the participants were able to exploit the use cases in business, at least to some extent.

5.1.4 Assessments of the RE templates

According to the respondents, the templates assisted in several phases (see Fig. 8). Also, some shortcomings of the templates were identified:

- The templates were considered to be too complex and time consuming for a small company that is trying to be agile and lean.
- The templates were too product-oriented and not technology oriented.
- The targets and outputs of the non-functional requirements were not clear.
- The title ‘Data resource’ requires a more detailed description.
- The actor description may be unclear for some people

The completion of the templates by the respondents took from a few hours (15 %), one day (62 %) to several days (23 %). For most of the respondents it was easy to apply

the templates in their working practice. It was agreed that for a large European project, documentation is a prerequisite, and the templates were good for that purpose. However, in SME companies, less formalisation and paperwork is done. For a smaller company and for internal usage, the templates are too heavy. Direct communication is preferred; a few overview slides with use case diagrams, and an ROI Excel sheet. One respondent felt that his/her work is technology oriented, but the templates are more business oriented; therefore they did not fit to the working practice. The templates included enough guidance according to most of the respondents. In one case, more guidance for the identification of new services was required. Also, working examples were required in each case that could help shape the structure and content of the proposed implementations. A more specific description of the target and output for non-functional requirements was identified as a development target. Also, more detailed explanations for the data resource titles is required. A total of 80 % of the respondents would recommend the usage of the templates to their colleagues and business partners.

5.2 Application of the method

The application of the method in ICARE and CDC projects are encapsulated in Table 8.

5.3 Summary

In the ecosystem, the co-operation between members is highly important and requires negotiation and compromise. Each member should find its own role in the ecosystem and also gain benefits in operating the ecosystem. The role of the key organisation becomes important as coordinating the business analysis, the requirement analysis, negotiation and specification between members. It is important that one member takes the role of coordinator; otherwise, the RE could result in distinct requirements and services that are not useful from the whole ecosystem's point of view. The service RE is more demanding within an ecosystem, since the RE must be coordinated at the ecosystem level and requires mutual understanding, several iterations and tight co-operation between ecosystem members.

The service RE method was seen as valuable and useful in the beginning of the service engineering process when starting the long-term development of new service architecture for digital ecosystem-based services. The service RE method was especially useful for describing, documenting and communicating the capabilities of the digital services and the service architecture they require. The method was also seen as useful in the analysis phase, where the different stakeholders work together. It is clear that the target of the requirements engineering must be done clearly enough for all of the participants. The description of the purpose and goal

of the use case description and analysis provides the understanding for the partners of what they are doing, and why and what are they helping to achieve. Continuous communication between ecosystem members is one of the key issues in achieving goals: both the single member's, the collaboration partners' and the ecosystem's. When all the ecosystem members use the same RE method, communication and co-operation is easy and fluent inside the ecosystem.

Despite the many advantages, shortcomings were also identified. Especially, the definition of quality requirements needs further exploration; special skills on quality attributes, e.g. performance, reliability and security, are required and should be present in the innovation and requirements analysis, negotiation and specification phases. For the service innovation, the quality attribute-specific ontologies should be provided for the use of ecosystem members. The methods for the elicitation, analysis, negotiation, trade-off analysis and specification of the non-functional requirements should also be provided with the proper guidance. The knowledge management model of the ecosystem is responsible for providing these ontologies and the methods to be used in each RE phase to achieve the non-functional requirements. Furthermore, the use of these quality ontologies and methods should be supported by the ecosystem support services, e.g. quality specific tools as support services provided via the cloud. Therefore, there should be an arrow between Knowledge management model and ecosystem support service elements in Fig. 2. There already exist approaches that can be utilised here, such as quality ontologies [58], quality-driven design methods and tool support for attaching quality properties for architectural elements [59].

Since the ecosystem is dynamic, it evolves all the time as new members, services and value networks emerge. The knowledge management model should evolve too, adapting to the needs of the ecosystem. Also, new support services should emerge as and when needed. For example, in the case of the first usage of the RE method, a demand was identified for a service that collects the new requirements as they emerge. Since the requirements innovation is continuous inside ecosystem, this kind of new service would enable the service providers to detect easily what kind of services has demand inside ecosystem. In addition, as the ecosystem monitors the quality of its services, it should also provide a matchmaking service for service selection to match the required quality with the provided quality.

6 Conclusions

Digital ecosystems bring out new challenges to service engineering; (i) the business and development environment is highly dynamic; (ii) the needs and demands of service customers are unclear and ever changing; and (iii) heterogeneous

Table 8 The method application

	ICARE	CDC
Description of the digital service ecosystem	Ecosystem of cloud services provided for digital content management, processing and delivery in interactive multi-screen TV services	An open service platform offering open real-time data from several data providers (offering data normalisation, integration and analysis, service hosting, open data APIs, service registries and the platform modules and services to third-party application developers)
Countries and partners	5 countries, 25 partners	4 countries, 7 partners
Roles of the partners in ecosystem	Service providers for content processing and rights management, and for user interaction Cloud IaaS and PaaS providers Interactive TV application and service developers	Data providers and data brokers Platform providers Application developers Service providers Service brokers
Goal of applying the service RE method	Identifying the requirements for a service framework and platform that would enable the digital service ecosystem to build and offer interactive multi-screen TV services	Extract the high-level user and business requirements for the open real-time data platform to be developed
Application of the service RE method	The use case description template was used to collect the information. The results formed a preliminary set of common services and potential new identified services. The analysis template was used for analysing the use cases, their commonalities and differences and clustering the identified services in service taxonomy. Several iterations were required for merging and refining use cases and representing the results as a set of master use cases that as a whole defined the baseline for service architecture modelling	The questionnaire was directed to potential application developers in the consortium. Each party who planned to create a showcase application on top of the platform defined their application use cases with the given RE document. The platform architects did initial requirement analysis for the platform from a user and business perspective. The technical requirement specification was done based on the results of the first two phases. This specification was used as bases for the architecture and system design definitions
Amount of the identified requirements	238 functional requirements 21 non-functional requirements 9 business requirements 7 constraints	14 functional requirements 3 non-functional requirements 2 business requirements 4 constraints
Service taxonomy: the identified digital service groups	User services/applications Cloud services Home network services Multi-screen interaction services Infrastructure services ICARE service framework services	Multi-modal mobility services
Status of readiness	All use cases are under work. The master use cases contain approximately 50 % of the original identified services. A proof of concept implementation is under work and is estimated to be finalised by 28/2/2015	All use cases are under work. The requirement specification and system design phase started in February 2014. The development phase started in June and will last until October, after which the pilots are made. The project ends 31/12/2014

and non-stop emerging technologies are used, or are available, for service implementation. However, service architects should be able to make decisions about what, why and how to develop digital services that have high business potential,

are attracting customers and can effectively be developed, operated and maintained.

This paper introduced a novel approach to defining the requirements of digital services in an ecosystem-based man-

ner. First of all, the approach defined what the digital service ecosystem is and how it differs from other ecosystems. Second, the service engineering process of a digital service ecosystem was outlined, keeping the focus on the requirements engineering of digital services. The service RE method introduced three main phases—service innovation, business analysis and requirements analysis, negotiation and specification—as a continuous and iterative engineering process that starts from business and end-user goals and provides a service taxonomy and a set of master use cases as an outcome. Each service is described with the functional and non-functional requirements, constraints and adaptation rules. The use of the service RE method was exemplified by a second-screen voting service, a work done by the Innovative Cloud Architecture for Real Entertainment (ICARE) ecosystem. Practical experiences of using the service RE method was also collected from the ecosystem members; the method was useful for describing, documenting and communicating the capabilities of the digital services. Especially, the method was useful in the requirements analysis phase, where ecosystem members worked together. However, further exploration is required with quality requirements that need special skills and knowledge on quality characteristics in all service engineering phases.

Acknowledgments This work was carried out at the VTT, Technical Research Centre of Finland, within the ITEA2-ICARE (Innovative Cloud Architecture for Real Entertainment) project. The authors thank the ICARE project members for their feedback and Neusoft for the example use case.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Chang E, West M (2006) Digital EcoSystems a next generation of collaborative environment. In: Eight international conference on information integration and web-based applications and services, vol 214, pp 3–23
- Ralyté J (2012) Viewpoints and issues in requirements engineering for services. In: IEEE 36th international conference on computer software and applications workshops, COMPSACW, Izmir, Turkey, pp 341–346
- Liu L, Yu E, Mei H (2009) Guest editorial: special section on requirements engineering for services. *IEEE Trans Serv Comput* 2(4):318–319
- Gu Q, Lago P (2009) Exploring service-oriented system engineering challenges: a systematic literature review. *Serv Oriented Comput Appl* 3:171–188
- Bano M, Zowghi D, Ikram N, Niazi M (2014) What makes service oriented requirements engineering challenging? A qualitative study. *IET Softw* 8(4):154–160
- Bano M, Zowghi D (2013) Service oriented requirements engineering: practitioner's perspective. In: Service-oriented computing workshops (ICSOC 2012), Shanghai, China, pp 380–392
- Bano M, Ikram N (2010) Issues and challenges of requirement engineering in service oriented software development. In: Fifth international conference on software engineering advances (ICSEA), Nice, pp 64–69
- Liu P, Nie G (2009) Research on service ecosystems: state of the art. In: International conference on management and service science, MASS '09, Wuhan, China, pp 1–4
- Knauss A, Borici A, Knauss E, Damian D (2012) Towards understanding requirements engineering in IT ecosystems. In: International workshop on empirical requirements engineering, Chicago, USA, pp 33–36
- Zhang J, Fan Y (2010) Current state and research trends on business ecosystem. In: IEEE international conference on service-oriented computing and applications (SOCA), Perth, WA, pp 1–5
- Li S, Fan Y (2011) Research on the service-oriented business ecosystem (SOBE). In: 3rd International conference on advanced computer control (ICACC), pp 502–505
- Iansiti M, Levien R (2004) Creating value in your business ecosystem. *Harv Bus Rev*. <http://hbswk.hbs.edu/item/3967.html>
- Bosch J (2009) From software product lines to software ecosystems. In: Proceedings of 13th international software product line conference (SPLC'09), pp 111–119
- Jansen S, Cusumano M (2012) Defining software ecosystems: a survey of software platforms and business network governance. In: Forth international workshop on software ecosystems, Cambridge, MA, USA
- Hanssen GK, Dybå T (2012) Theoretical foundations of software ecosystems. In: Proceedings of the forth international workshop on software ecosystems (IWSECO), Cambridge, MA, USA, pp 6–17
- Riedl C, Böhm T, Leimeister JM, Krcmar H (2009) A Framework for analysing service ecosystem capabilities to innovate. In: 17th European conference on information systems, Verona, Italy, pp 2097–2108
- Ruokolainen T (2013) A model-driven approach to service ecosystem engineering. PhD Thesis, University of Helsinki, Department of Computer Science, Helsinki, Finland
- Blau B, Krämer J, Conte T, van Dinther C (2009) Service value networks. In: IEEE conference on commerce and enterprise computing, Vienna, Austria, pp 194–201
- Schroth C (2007) The internet of services: Global industrialization of information intensive services. In: 2nd International conference on digital information management (ICDIM'07), Lyon, France, pp 635–642
- Riedl C, Böhm T, Rosemann M, Krcmar H (2009) Quality management in service ecosystems. *Inf Syst e-Bus Manag* 7(2):199–221
- Wu C, Chang E (2005) A conceptual architecture of distributed web services for service ecosystems. In: 18th International conference on computer applications in industry and engineering (CAINE), Hawaii, pp 209–214
- Ruokolainen T, Ruohomaa S, Kutvonen L (2011) Solving service ecosystem governance. In: Proceedings of the 15th IEEE international EDOC conference workshops, pp 18–25
- Chesbrough HW (2011) Bringing open innovation to services. *MIT Sloan Manag Rev* 52:85–90
- Khriyenko O (2012) Collaborative service ecosystem-step towards the world of ubiquitous services. In: Proceedings of the IADIS international conference collaborative technologies, Lisbon, Portugal, pp 19–21
- Pantsar-Syväniemi S, Purhonen A, Ovaska E, Kuusijärvi J, Evesti A (2012) Situation-based and self-adaptive applications for the smart environment. *J Ambient Intell Smart Environ* 4(6):491–516

26. Ferronato P (2004) DBE architecture requirements, del 2.2: Architecture scope document. D.B.E. Digital Business Ecosystem contract no 507953. <http://www.digital-ecosystems.org/>
27. Ovaska E, Kuusijärvi J (2014) Piecemeal development of intelligent smart space applications. *IEEE Access* 2:199–214
28. Kutvonen L, Ruokolainen T, Ruohomaa S, Metso J (2008) Service-oriented middleware for managing inter-enterprise collaborations. In: *Global implications of modern enterprise information systems: technologies and applications*, ser. advances in enterprise information systems (AEIS), pp 209–241
29. Flores F, Mora M, Álvarez F, Garza L, Durán H (2010) Towards a systematic service-oriented requirements engineering process (S-SoRE). In: *ENTERprise information systems, communications in computer and information science*, vol 109, pp 111–120
30. Immonen A, Pakkala D (2014) A survey of methods and approaches for reliable dynamic service compositions. *Ser Oriented Comput Appl* 8(2):129–158
31. Al-Fataftah IA, Issa AA (2012) A systematic review for the latest development in requirement engineering. *World Acad Sci Eng Technol* 6:691–698
32. Loniewski G, Insfran E, Abrahão S (2010) A systematic review of the use of requirements engineering techniques in model-driven development. *Model driven engineering languages and systems*. Lecture notes in computer science, vol 6395, pp 213–227
33. Husnain M, Waseem M, Ghayyur SAK (2010) An interrogative review of requirement engineering frameworks. *Int J Rev Comput* 2:1–8
34. Chesbrough HW, Appleyard MM (2007) Open innovation and strategy. *Calif Manag Rev* 50:57–76
35. Chan CML (2013) From open data to open data innovation strategies: creating E-services using open government data. In: 46th Hawaii international conference on system sciences, Wailea, HI, USA, pp 1890–1899
36. Stathel S, Finzen J, Riedl C, May N (2008) Service innovation in business value networks. In: 18th International RESER conference, Stuttgart, Germany, pp 288–302
37. Fricker S (2010) Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. *Requirements engineering: foundation for software quality*. Lecture notes in computer science, vol 6182, pp 60–66
38. Schindlholzer B, Uebernickel F, Brenner W (2011) A method for the management of service innovation projects in mature organizations. *Int J Serv Sci Manag Eng Technol* 2(4):25–41
39. van Eck P, Wieringa R (2003) Requirements engineering for service-oriented computing: a position paper. In: *Proceedings of first international E-services workshop*, ICEC 03, Pittsburgh, USA, pp 23–29
40. De la Vara González JL, Díaz JS (2007) Business process-driven requirements engineering: a goal-based approach. In: 8th Workshop on business process modelling, development and support, Trondheim, Norway
41. Cardoso ECS, Vitoria B, Almeida JPA, Guizzardi G (2009) Requirements engineering based on business process models: a case study. In: 13th Enterprise distributed object computing conference workshops, Auckland, pp 320–327
42. Khosravi A, Modiri N (2012) Requirement engineering in service-oriented architecture. In: *International conference on networks and information (ICNI 2012)*, Bangkok, Thailand, pp 101–105
43. Liegl P, Schuster R, Zapletal M, Huemer C, Werthner H, Aigner M et al (2009) A methodology for process based requirements engineering. In: 17th IEEE international requirements engineering conference, Atlanta, pp 193–202
44. Hussein M, Yu J, Han J, Colman A (2012) Scenario-driven development of context-aware adaptive web services. In: 13th International conference on web information systems engineering—WISE 2012. Lecture notes in computer science, vol 7651, pp 228–242
45. Seyff N, Maiden N, Karlsen K, Lockerbie J, Grunbacher P, Graf F et al (2009) Exploring how to use scenarios to discover requirements. *Requir Eng* 14:91–111
46. Kimita K, Akasaka F, Shimomura Y, Öhrwall Rönnbäck A, Sakao T (2009) Requirement analysis for user-oriented service design. *Asian Int J Sci Technol Prod Manuf Eng* 2(3):11–23
47. Kett H, Voigt K, Scheithauer G, Cardoso J (2008) Service engineering in business ecosystems. In: *Proceedings of the XVIII international RESER conference*, Stuttgart, Germany, pp 1–22
48. Wiesner S, Peruzzini M, Doumeings G, Thoben KD (2012) Requirements engineering for servitization in manufacturing service ecosystems. In: *Conference on industrial product service systems*, Tokyo, Japan, pp 291–296
49. Ruokolainen T, Kutvonen L (2009) Managing interoperability knowledge in open service ecosystems. In: 13th Enterprise distributed object computing conference workshops, Auckland, New Zealand, pp 203–211
50. Dobson G, Sawyer P (2006) Revisiting ontology-based requirements engineering in the age of the semantic web. in: *International seminar on dependable requirements engineering of computerised systems*, Halden
51. Castañeda V, Ballejos L, Calusco L, Galli R (2010) The use of ontologies in requirements engineering. *Glob J Res Eng* 10(6):2–8
52. Xiang J, Liu L, Qiao W, Yang J (2007) SREM: A service requirements elicitation mechanism based on ontology. In: 31st Annual international computer software and applications conference, pp 196–203
53. Kaiya H, Saeki M (2005) Ontology based requirements analysis: Lightweight semantic processing approach. In: *Fifth international conference on quality software (QSIC'05)*, Melbourne, Australia, pp 223–230
54. Ovaska E, Salmon Cinotti T, Toninelli A (2012) The design principles and practices of interoperable smart spaces. In: Xiaodong L, Yang L (ed) *Advanced design approaches to emerging software systems: principles, methodologies and tools*. IGI Global, pp 18–47
55. ISO/IEC, 2001. ISO/IEC 9126–1 international standard: Software engineering—product quality. Part 1: quality model. International Organization for Standardization, Geneva
56. Immonen A, Palviainen M, Ovaska E (2014) Requirements for open data based business ecosystem. *IEEE Access* 2:88–103
57. OASIS (2008) Reference architecture for service oriented architecture 1.0
58. Zhou J, Niemelä E, Savolainen P (2007) An integrated QoS-aware service development and management framework. In: *Working IEEE/IFIP conference on software architecture, WICSA'07*, Mumbai, India, pp 136–145
59. Ovaska E, Evesti A, Henttonen K, Palviainen M, Aho P (2010) Knowledge based quality-driven architecture design and evaluation. *Inf Softw Technol* 52(6):577–601