Developing Advanced Services for SMEs using Service-Centric Tools: Experiences and Challenges

Francisco J. Garijo[1], Juan Pavón[2], Carlos Rodríguez[2], and Damiano Spina[2]

[1] Telefónica I+D,
Emilio Vargas 6, 28043 Madrid, Spain
fgarijo@tid.es
[2] Facultad de Informática, Universidad Complutense de Madrid,
Profesor José García Santesmases s/n, 28040 Madrid, Spain
jpavon@fdi.ucm.es, carlosrodriguez@computer.org,
damianospina@gmail.com

Abstract. This paper presents the experiences and the results achieved for delivering sophisticated Business Support Systems to Small and Medium Enterprise (SMEs), using advanced service centric solutions developed in the SECSE project. The SECSE methodology provides concepts methods, processes, and techniques for developing service centric applications, and the supporting toolkit include tools for supporting the principal activities such as publication, discovery, execution, monitoring, and testing. The paper illustrates the use of the methodology for building a pilot, which delivers to SMEs integrated solutions including computing and communication infrastructure, and complex functionality such as CRM, Work-Flow processing, and logistics. Metrics and evaluation results for assessing both the technical benefits and the business benefits of the SECSE outcome are also described.

1 Introduction

Materialization of the “service paradigm” which includes concepts like Service Oriented Development, Service Oriented Architecture, Service Oriented Applications and others, call for the need of a joint effort between industry and academia to provide technology, methodology and supporting tools for building service systems, and making them accessible to users. The SECSE project [1] has faced this challenge by gathering IT companies, developers and research labs to find out new solutions for exploiting the full potential of software services and for providing a sound engineering framework for the creation of services and service-centric systems. The project has delivered two principal outcomes: A methodology providing a conceptual model, methods, processes, and techniques for developing service centric applications, and supporting tools including capabilities to describe, compose, discover, monitor and deliver services. Research results have been validated by Industrial partners exercising both the techniques and the supporting tools for developing application pilots in Automotive and Telecommunication domains. Additional development experiences have been done integrating new services from each industrial partner, in order to set up a common experimental setting with which to assess technical and business benefits. Research effort lead by academic partners has been extensively described in different papers [9,10,11,12,8,5,14,7,16]. This paper focus on the industrial experiences and specifically in the experiences and the results achieved developing advances services for SMEs. The organization of the paper is the following. After this introduction, section 2 gives a brief outline of the SECSE methodology and the supporting toolkit. Section 3 presents the evaluation approach describing the Logistic4SME pilot, the evaluation process and metrics. The evaluation results are in section 4, and finally the Conclusions and open issues are summarized in section 5

2 The SECSE methodology and toolkit

The Project has produced two principal outcomes: A methodology and supporting tools for developing service centric systems.

The SECSE methodology provides a conceptual model including terminology, concepts and relationships for service engineering, methods, processes, and techniques for developing service centric applications. The functional areas are depicted in figure 1.

- *Service-centric System Engineering* comprises the principal processes for building service-centric systems such as analysis, design, development, deployment and operation. SECSE provide specific subprocesses and supporting tools for design-time and runtime.
- *Service Engineering* addresses development practices aiming to deliver atomic services.
- *Service Acquisition and Provisioning* focus on process and tools for "service marketplace" actors such as service consumers, service providers and service brokers, to publish, negotiate, monitor, and deliver services.

The SeCSE suite aims to support service stakeholders to apply the methodology for building and managing service-centric systems. The toolkit covers the entire life cycle including capabilities to describe, compose, discover, monitor and deliver services.

1. *Service Discovery* tools provide specialized support for finding services and approximate orchestration starting from business level UML use cases and captured requirements. The SECSE Tools are Early Service Discovery Suite ESD, Architecture Service Discovering (ASD) and Runtime Service Discovery (RSD).
2. *Service Specification* tools enhance standard service specification and exposition formalisms by means of facets [7] incorporating semantic descriptions. The SECSE Tool is Facet Management Tool.

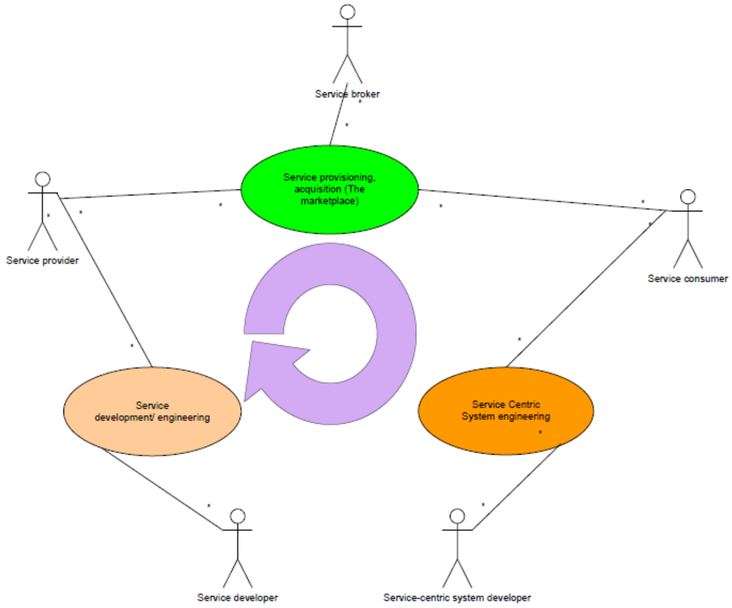

Fig. 1. SECSE Methodology (from [1])

3. *Service Publication* tools allow service registration and exposition. The SECSE Tool is SeCSE Registry.
4. *Service Testing* tools test services to build a confidence in the service's functional and non functional features. The SECSE Tools are Testing Facet Generator, Regression Testing Tool and Quality of Service Tool.
5. *Service Monitoring* aim to provide service-centric facilities for checking compliance with functional and non-functional requirements. The monitoring tools work both at process level (loose monitoring) and at single operation level (tight monitoring). The SECSE Tools are Tight Monitoring and Loose Monitoring.
6. *Service Composition* tools enhance standard BPEL[13] tools incorporating execution and monitoring rules. The SECSE Tool is Composition Designer.
7. *Service Deployment and Execution* tools execute enriched BPEL services by offering negotiation, binding and re-planning mechanisms as well as run-time service discovery. The SECSE Tools are Deployer and Runtime Execution engine.

3 A SECSE case study: Advanced Logistic Service for SMEs

The SECSE project has developed several case studies for assesment of the SECSE methodology and tools [2,3]. One of them is the a Business Support System for SMEs. The business goal of this system is to make available for SMEs through Web service technologies, sophisticated and expensive applications similar to those available in big companies such as CRM, EnterpriseWorkforce Management, accounting and billing.

A concrete case has been considered for a SME specialized in home repair, this includes every mishap bound to happen in your home, from a dripping tap to a blown pipe, to a malfunctioning heater, etc... This functionality might be adapted to other domains related to repair and maintenance.

The Logistics4SME service is delivered to SMEs as an integrated, flexible, scalable, and customizable package; for home repairing.

The system supports the following SME Actors: Secretary who manages phone calls from customers, creates incidence records, assigns priorities to tasks, and assigns tasks to technicians; Technician who gets his/her daily work plan, performs the tasks according to the schedule, accepts/rejects new tasks, sends task reports back to the company.

The system has been built by an iterative process of creating, selecting, adapting, validating, and assembling in-house components and commercial components. In-house components were initially based on the BOGAR [6] library which has proven their effectiveness for the development of several kinds of interactive services in Telefnica I+D. Using BOGARs components leads to a logical architecture structured into two layers: The control layer populated by control components, and the Resource layer made up of Application resources which offer interfaces to the controllers, an/or send information to them.

The pilot is made up of the following Controller components implemented with BPEL:

- **Access Controller** which manages user access and authentication. It uses the following resources:
  - *Access Visualization*: Implements access visualization through web interfaces
  - *Persistency Resource*: Provides persistency operations related to store, retrieve, and validate user access and authentication
- **Secretary Assistant** is responsible of the following activities:
  - *Customer management*: functionality related to the management of SME clients. It allows adding and removing customers from the database, modifying their personal data or searching among customers with varied criteria.
  - *Failure incidences*: functionality and support to create incidence records assign priorities to tasks and assign tasks to technicians.
  - *Technicians work flow*: functionality and support for Technician's workflow management. It helps the secretary to add new technicians to the database, manage their timetables or search which ones are working at a certain moment.

  The resources needed to achieve its activities are the following:
  - *Secretary Visualization*: Implements the secretary User Interface, providing high level operations for visualizing and managing application information, and user input.

  - *Enterprise Work Force Management*: Provides operations for finding the most suitable technician to achieve repair tasks taken into account the type of incident, technician, skills, user location, and technician work-plan. It also elaborates, manage and provide upon request technicians work-plan.
  - *Persistency Resource*: Provides persistency operations related to store, retrieve, and validate customers, incidences, and technicians.
- **Tecnician Assistant** is responsible of managing the daily work plan of each technician, informing about the failure incidences he has accepted (customer information, failure description, priorities). He/she can accept or reject proposed tasks, and send task reports back to the SME, after completing the reparation.
  The resources needed to achieve its activities are the *Enterprise Work Force Management Service* which provides operations to retrieve, and update technicians work-plan; the *Billing Service* which provides operations for elaborating customers billing; the *Spare Parts Service* which provide operations for searching and locating spare parts providers.

4 The evaluation process

The evaluation process has been based on the development of three versions of the pilot. It was achieved by tight cooperation among service providers, service integrators for service assembly, and service developers.

Logistics4SME_V1 was developed using open source development tools such as Eclipse and Active BPEL.The pilot implemented an initial set of functions for the secretary and the technician. Access and integration of key components such as the EWMS platform, and support for user mobility, were simulated.

Logistics4SME_V2 incorporates technicians mobility (PDA), integration of the EWMS platform and new Web Sercies for third parties such as User Location and Spare Parts Management.

Logistics4SME_V3 was a re-engineered version of Logistics4SME_V2 using the SECSE methodology and the SECSE toolkit.

The two scenarios have been used to define a collection of testing cases for validating the requirements on the SECSE tools.

The first scenario aims to assess application of the SECSE methodology for the development of a service component which will be integrated with other single and composite service for delivering the final service to the SME. The methodology Areas covered by this scenario are: Service Engineering, Service Acquisition and Provisioning. The tools used in this scenario are: Specification tools, Service Publication and Delivery tools, and Testing tools.

The second scenario assumes that the composite Web Service has been developed using conventional BPEL based tools. SECSE tools are used for enhancing the composite Web Service using the binding and negotiation mechanism and the monitoring rules. This scenario covers methodology activities in Service-centric Service Engineering and Service Acquisition and Provisioning. SECSE tools used are: Service-centric Service Engineering tools, Service Discovery tools, and Testing tools.

The development of the first prototype led to the definition of an initial set of 345 requirements, and the metrics to measure the satisfaction of the requirements. Four types of metrics have been defined.

- M1: addresses the degree of the implementation. Values range from 1: not implemented, to 5 fully implemented.
- M2: measures the accuracy level – percentage of acceptable errors, with error levels defined/measured as aside. 1 is 0%, 2 is 25%, 3 is 50%, 4 is 75% and 5 is 100%.
- M3: estimates the quality of documentation : values range from 1 awful, 2 is bad, 3 medium, 4 good and 5 excellent.
- M4: estimates the quality of the user interfaces. Values range from 1 awful, 2 is bad, 3 medium, 4 good and 5 excellent.

The initial list of requirements has been refined during the second phase of the pilots obtaining a final set of 190 requirements which have been evaluated using the testing cases and the evaluation metrics.

The evaluation approach consisted on associating to each testing case the requirements covered, and the evaluation metrics for each requirement. Then executing the testing cases using the computing and communication infrastructure of the third prototype, and estimating the metric value according the result of the testing case.

5 Evaluation results

Overall evaluation has been done gathering the testing results by each partner, grouping metric scores by service engineering activity, and then calculating average values for each group. More detailed description of this process can be found in [4]. The results are summarized in the table 1 [3].

Global scores shows that values for tools supporting the principal activities of the methodology ranges from a minimum of 3,6 to a maximum 4,2. This may be interpreted as a score closed to good, according the metrics used. It should be noted that this score come from to the evaluation of functional aspects, consequently the results confirm that the acceptability of the tools delivered.

Score differences among partners are due to their pilots and testing cases, which cover specific requirements. In general, score under 3 are due to the partial coverage of required functionality in the tools, and usability under the acceptable levels. The lack of value in TID's service discovery row was due to the fact that the Early Service Discovery tool was not considered in the testing case. In addition, the functionality of Architecture Service Discovering Tools (ASD) and Runtime Service Discovering tool (RSD) couldn't be tested because they were

[3] ATOS (Atos Origin), CRF (Centro Ricerche Fiat), KD (KD Software), TID (Telefnica Investigacin y Desarrollo)

Table 1. Overall evaluation scores

Activity	ATOS	CRF	KD	TID
Service Specification & Testing	4,5	2,9	4,1	3,0
Service Discovery	4,8	3,1	3,7	
Service-centric System Engineering & Testing	3,8	4,1	2,0	3,6
Service Publication & Delivery	4,2	4,0	5,0	3,8
Integration	3,7	4,1	4,6	3,6
General requirements	4,4	2,7	4,1	4,0

not integrated in the SECSE run-time platform. Experiences on tool usage and suggestions for improvement identified by TID where the following:

Service Specification and Testing tools. The Facet Management Tool eases the process of describing the most significant characteristics of a web service. It provides an initial collection of service concepts and properties which are represented as facets. These facets covers functional aspects such as interfaces, signature and semantics, constraints and exceptions, quality of service aspects, and business aspects. Facet description is based on XML. This facilitates flexible formal definition and syntactic validation of facets, and also to attach semantic annotations which would allow the semantic validation of local properties and the overall service properties.

Improvements to meet industrialization requirements are: consistency checking and semantic validation of facets, and compatibility with existing standard, e.g. UDDI, in order to guarantee universal access and use of deployed services. Enhancement of user assistance for data definition, diagnostic and explanations when errors occurs, and pro-active support for error correction would also be necessary.

The two tools provided by the Regression Testing Toolkit: the Testing Facet Generator and the Regression Testing Tool, are suitable for industrial use. The Regression Testing Tool executes the testing cases associated to the service testing facet, which could be generated by the Testing Facet Generator tool. In practice, there is a strong need to maintain the consistency among the service code, the testing cases uploaded in the registry and the testing results obtained using the Regression Testing Tool. This could be achieved through the integration of both tools. To facilitate the service validation process there is a need for a single interface integrating the functionality for generating the testing cases, uploading them into the service testing facet, and then executing the testing cases using the service code.

Service Discovery tools. Three different tools for Service Discovery have been successfully tested independently. The Early Service Discovering tool ESD provides a wide range of facilities for expressing search criteria by combining functional and non functional requirements. Industrial use of the ESD tool is strongly dependent on the availability of a large amount of registered services with clear and consistent specifications based on widely accepted standards.

The Architecture-time Service Discovery Tool (ASD) offers very interesting features but definition of search queries during the design phase requires the use of UML profiles created with proprietary UML tools.

Runtime Service Discovery Tool (RSD) offers dynamic and contextual queries during service execution. Industrial use of the RSD will be strongly dependent on fulfilling the highest quality standards for robustness, reliability and efficiency.

Service Publication and Delivery. Publication facilities are delivered as part of the SECSE registry. User functionality incorporate: web based interfaces for registration and management of Web Service descriptions using facets. Specific functions for uploading, browsing and deleting Web Service are available. Right now it is mandatory to use the publication facilities provided by the SECSE registry in order to use the overall SECSE environment. However it would be advantageous that publication tools would be decoupled from the registry. Overall the experience of using the SeCSE registry User Interface is satisfactory. Suggested improvement are: stability and robustness of the User Interface functionality still necessary, version control, and additional operation and User Interfaces to get the Service Access Points (SAP).

Federation capabilities work well, and are one of the most appealing features of the toolkit.

Service-centric Systems Engineering tools. This includes tools supporting service composition, service execution and service monitoring. The Composition Designer -CD- provides facilities to enhance standard BPEL tools incorporating innovative features such as dynamic binding and execution and monitoring rules. Its textual interface for defining the workflow logic and the binding rules might be improved to ease understandability and usability. In its present state this can be an obstacle for industrial use, due to the existence BPEL design tool which offer more sophisticated graphical interface

The Runtime Execution engine executes enriched BPEL compositions defined with the CD. This includes implementation of negotiation, binding and re-planning mechanisms as well as runtime service discovery.

Monitoring capabilities are fundamental in order to get quality control metrics, and if necessary undertake corrective actions. The monitoring tools have been tested in isolation providing satisfactory results; however integrated testing using the SECSE execution environment couldnt be done. Successful integration of the CD, the run time engine, and the monitoring tools to fulfill industrial requirements regarding robustness, stability, reliability and efficiency still an open issue. Although its resolution would require more effort than planned, the business advantages in terms of cost-benefit would be largely positive.

6 Conclusions and future challenges

The SeCSE project is one the leading EU funded IPs which has assumed the challenge to find out innovative technologies, and supporting tools for building service centric systems. Technical and business evaluation of both the SECSE methodology and the supporting toolkit has been done exercising them developing application pilots by industrial partners involved in the project.

The evaluation conclusions which were drown from the experiences of developing advanced logistics service for SMES are detailed below.

- Service developers found the SECSE methodology easy to understand. The conceptual model, and description of functional areas, processes, tasks, and work-products, allows the development team to have a common view of the development process, then facilitating sharing of knowledge and improving team communication. The methodology provides advice and guidance for accomplishing tasks using the SeCSE Tools. It supports rapid and incremental development including specific aid for the integration of legacy systems and heterogeneous components.
- The SECSE toolkit gives integrated functionality and support for achieving key service engineering activities such as service specification, publication, delivery, composition design, service execution, and monitoring, and testing. Tool integration is key to optimizing the development process, lowering costs and speeding up service provisioning. Thus using SECSE tools increase business efficiency reducing service development and maintenance costs, decreasing the time to market of new services, and facilitating extensibility, and envolvability.
- Evaluation conclusions based on the testing cases have also demonstrated business viability and possible industrialization of tools supporting service specification publication and delivery. For the rest of the tools, evaluation of the prototypes has shown the potential of key disruptive features such as the specification and execution of service monitoring properties, negotiation, dynamic binding and re-planning mechanisms, as well as run-time service discovery. Although this functionality has been implemented according to industrial needs, its assimilation into service engineering practices would require the tools to satisfy the industry standards of usability, user interface, documentation, stability and efficiency. Tool installation and interoperability issues should also be considered, as they were a serious obstacle for running the tools and the testing cases.

Overall industrial evaluation has confirmed the feasibility and the potential of the methodology and the supporting tools. Future challenges for persistency, acceptance and evolution of the SECSE work are: influencing standards bodies to recognize service specifications enhancement; demonstrating robustness, scalability, interoperability and usability of architecture and computing models designed and implemented in the project. Delivering the project implementations under open source would facilitate overcoming this issues which require an active participation in standard forums, and spending additional development effort to upgrade the prototypes into stable tools.

References

1. A5.D4.2 SeCSE Methodology V1.3 .(http://secse.eng.it).
2. A6.D11 - CRF Automotive pilot development architecture (v2) (http://secse.eng.it).
3. A6.D12 - T-A Telecommunication pilot development architecture (V2) (http://secse.eng.it).
4. A6.D16 Testing and evaluation of web services for pilots.
5. F.J. Nieto, L. Bastida, M. Escalante, A. Gortazar.: "Development of Dynamic Composed Services based on the Context". Proceedings of the 2nf International Conference Interoperability for Enterprise Software and Applications (I-ESA 2006), Bordeaux, France, March 2006.
6. Garijo, F J., Bravo S., Gonzalez, J. Bobadilla E. BOGAR_LN: "An Agent Based Component Framework for Developing Multi-modal Services using Natural Language". L.N.A. I, Vol. 3040. pp 207-. Springer-Verlag. 2004.
7. J. Walkerdine, J. Hutchinson, P. Sawyer, G—. Dobson, V. Onditi : "A Faceted Approach to Service Specification", Proc. Second International Conference on Internet and Web Applications and Services (ICIW07), Mauritius, May 2007. IEEE Computer Society Press.
8. Kozlenkov A., Spanoudakis G., Zisman A., Fasoulas V., Sanchez Cid F.: "Architecture-driven Service Discovery for Service Centric Systems" Journal of Web Services Research, 4(2), pp. 81-112, 2007.
9. L. Baresi and S. Guinea. "Towards Dynamic Monitoring of WS-BPEL Processes", Proceedings of the 3rd International Conference of Service-oriented Computing (ICSOC'05). Amsterdam, The Nederlands, 2005. Lecture Notes in Computer Science, Vol. 3826, pp. 269-282.
10. M. Colombo, E. Di Nitto, M. Mauri: SCENE: A Service Composition Execution Environment Supporting Dynamic Changes Disciplined Through Rules. ICSOC 2006: 191-202.
11. M. Di Penta, R. Esposito, M.L. Villani, Roberto Codato, M. Colombo, and E. Di Nitto. WS-Binder: "A Framework to enable Dynamic Binding of Composite Web Services". In Proc. International Workshop on Service Oriented Software Engineering (IW-SOSE'06), pages 77-80, Shanghai, China, May. 2006. ACM.
12. M. Di Penta, G. Canfora, M. Bruno, G. Esposito, V. Mazza – "Search Based Testing of Service Level Agreements" to appear in proceedings of the Genetic and Computation Conference (GECCO 2007), July 2007, London, UK, ACM Press.
13. OASIS Web Services Business Process Execution Language Version 2.0 Committee Draft, 25 January 2007.
14. Spanoudakis G., Mahbub K., Zisman A.: "A Platform for Context Aware Runtime Web Service Discovery", IEEE 2007 International Conference on Web Services (ICWS 2007), Salt Lake City, Utah, USA.
15. UDDI v3.0, OASIS Standard, February 2005, http://www.uddi.org
16. Zisman A. Mahbub K., Spanoudakis G.: "A Service Discovery Framework based on Linear Composition", IEEE 2007 International Service Computing Conference, Salt Lake City, Utah, USA.