Open SOA value add for collaborative services delivery to rural SMMEs

Rudi DE LOUW¹, Jörg DÖRFLINGER²
¹SAP Research, CEC Pretoria, Persequor TechnoPark, Pro Park Building 3, 29 de Havilland Crescent, Scientia, Pretoria 0001, South Africa
Tel: +27 12 349-3100, Fax: +27 12 349-3131, Email: rudi.de.louw@sap.com
²SAP Research, CEC Karlsruhe, Vincenz-Priessnitz-Strasse 1, Karlsruhe, 76131, Germany
Tel: +49 6227 7-52556, Fax: +49 6227 78-47958,
Email: joerg.doerflinger@sap.com

Abstract: Small, Medium and Micro Enterprises (SMMEs) in rural South Africa face difficult business conditions. The Rural Living Lab (RLL) in Sekhukhune introduced Infopreneurs that provide a Collaborative Working Environment to deliver collaborative business services to the rural enterprises, based on an Open Service Oriented Architecture (Open SOA) approach.

The paper provides information about the methodologies and concepts of the technical solutions developed and planned to overcome the SMMEs difficulties. The aim is to adapt the technologically advanced OSOA, used for all the RLLs in the C@R project, for use in the emerging economy circumstances specific to the South African RLL, while staying true to the needs, requirements and contexts of the local users and communities. Due to the current stage of advanced prototype development and end user interaction in the living lab this paper will already present important lessons learned and clear benefits of the C@R software approach.

Keywords: Rural Living Labs, SMME (Small, Medium and Micro Enterprises), Collaboration, Collaborative Working Environments (CWE), Service Oriented Architecture (SOA), Open Service Oriented Architecture (OSOA), Collaborative Services, Web 2.0, Semantic Web, Ontologies

1. Introduction

In South Africa, as in most emerging economies, Small, Medium and Micro Enterprises (SMMEs) constitute a significant portion of the economy. Especially in rural areas, SMMEs face difficult business conditions, mainly related to the lack of adequate infrastructure, disconnection to economically strong markets, relatively high transport and transaction costs because of long distances, low supply chain volumes, limited economies of scale, skills shortages, missing managerial know-how, non-awareness of business opportunities, erratic and costly connectivity, limited cash flow and difficulties in securing business financing [15].

The Collaboration@Rural (C@R) [1] project attempts to promote economic development in the rural areas by overcoming many of these challenges and to promote the incubation of SMMEs and their collaboration. The project is an Integrated Project, funded by the IST programme of the European Commission's 6th Framework Program [2, 3]. Following the European Network of Living Labs (ENoLL) initiative [4], a Rural Living Lab (RLL) is set up in the Sekhukhune region of South Africa, and applies Collaborative Working Environments (CWEs) as a technological response to the challenges that prevent rural development.
To overcome the difficult business conditions (as described above), a collaborative-based SMMEs incubator in rural areas is created as a facility designed to encourage and foster entrepreneurship in rural and remote regions, and to minimize obstacles to SMME business formation and growth. This is accomplished by providing the collaborative services and tools integrated in the C@R platform. Infopreneurs [5], a kind of e-entrepreneur that act as service agents for SMMEs, and apply the collaborative incubation mechanisms to the SMMEs, were introduced in the Sekhukhune RLL. Several service-oriented, collaborative scenarios are identified in the Sekhukhune RLL. [10] to enhance socio-economic development for the rural community.

Up to the commencement of the C@R project, there were no e-business activities in any of these communities. Now the Infopreneurs will provide, via bandwidth-efficient systems, an extended collaborative service bundle that will offer, amongst others, mentoring support, business management tools and business development services to local SMMEs. The main impact of this service bundle is to reduce high transaction and logistics cost, to upscale supply chain volumes, to release product related market constraints, to share knowledge and know-how, and to ensure long-term economic viability. The Infopreneurs being part of the community is absolutely necessary for trust and acceptance issues, and to be role models of entrepreneurship. The usage of the OSOA architecture approach provides a clear business benefit by significantly reducing and simplifying the implementation efforts and thus decreases implementation time and costs.

The following two chapters describe the development methodologies and technical solutions which build the technical basis for the C@R project. Chapter five and six provide information about the current status of the development and already achieved development results. The business benefits in chapter seven verify the efforts spent and expose the clear benefits of the C@R architecture approach.

2. Objectives

The C@R project defines a common three-layered Service Oriented Architecture (SOA) [10] for the seven RLLs involved; six in Europe and the one in South Africa. Our aim is to adapt this technologically advanced common architecture to the emerging economy circumstances specific to this RLL, while staying true to the needs and requirements of the local users and communities, taking into account the local African and rural contexts, and following the principles of Participatory Design and Design in Use [6, 10]. By leveraging the synergies within C@R LL network, we plan to introduce the use of Collaborative Working Environments (CWEs) as key enablers, catalyzing rural development.

3. Methodologies

The main methodology of Living Labs is followed to realize sustainable trusted innovation in a user centred co-design process. User-centric research methods are used in real life environment to identify and build solutions in the constantly changing ‘living’ environments. In the RLL contexts, users and developers are co-creating innovation, and users are working and living with those services and applications in their daily life settings. The Living Lab evolves [11] from preparation, vision building, user community building, and analysis of the current situation to mock-up development, and finally to extensive application development and field experimentation. The RLL becomes an innovation environment and impact on the rural development can be measured.
Within this broad framework, ‘Participatory Design’ and ‘Design in Use’ sessions are scheduled, making use of a support environment called “Studio” [12], where we start on the expert level, but refine the requirements via end-user input.

The first design sessions are executed at expert level to develop the use cases.

Then the actual end-users (mainly Infopreneurs) get involved. As most of the SMMEs in the Sekhukhune area do not have strong technical skills to properly document their needs and wishes, and to ensure their requirements are properly understood, some training and guidance is required. The users are not merely consulted, but collaborate in the development of the system.

An iterative process, with rapid design cycles, is used. This is similar in many aspects to the so-called ‘agile’ development methodologies [13, 14]. With the ‘Design in use’ methodology, users’ real-life working behaviour is used to develop and refine the systems.

This demands a mindset change in designers and developers to accept users as co-designers, and changes in application as a part of the development process. [10]

One of the problems is that the mindset change does not easily take place with the developers, especially when the development team is distributed, with more than half the developers in Europe and having no regular contact with the end-users. Another problem is that many end-users depend on these applications for an income, so the prototyping cycles become very short and rapidly have to produce full working applications.

Technical and Development Principles

On a more technical approach, we define a few guidelines for the development principles:

- Offline usage should be the norm: Due the lack of infrastructure, erratic and relatively expensive communications, all applications should be able to run offline. When connectivity is available, it should be used as efficient as possible. Another reason for offline usage is the business principle that all local data belongs to the Infopreneurs.
- Multi-modality: Various User Interfaces (UIs) should be accommodated, e.g. mobile phones, web pages, Rich Web UIs using AJAX, etc.
- Legacy systems: Compatibility with legacy systems is a requirement.
- Development tools: For the C@R project, the majority of the development tools are Free and Open Source Software (FOSS). Most of the tools used are from the Apache Foundation [16]
- Simplification: From the Collaborative Knowledge Sharing scenario (as discussed in the ‘Developments’ section) we have learned that it helps to simplify, to get a common theme, and to adapt the systems to the technical restrictions of the emerging economy environments.

4. Technical Architecture

The C@R architecture provides a platform for user centric Collaborative Working Environments (CWE) which are used to create a distributed knowledge network in rural environments. The architecture approach is based on the principles of an Open Service Oriented Architecture (OSOA) [7] and thus supports the interoperability as well as the distributed development and instantiation of software components used to realize end user applications. The advantages and principles of this approach will be discussed in detail in the following sections.
4.1 C@R OSOA architecture

The C@R architecture is based on a three layered model (see figure 1). From bottom to top, Collaborative Core Service (CCS) components in Layer 1 are the atomic resources which will be orchestrated by a control middleware represented in Layer 2 using service scripts and collaborative functions (Orchestration Capabilities) into Software Collaboration Tools (SCT) in Layer 3. These SCTs than will be instantiated into applications utilized in the Rural Living Labs (RLL) being part of the CWE.

Figure 1: C@R layered architecture with additional Web 2.0 and semantic layer

A key component in the C@R architecture is the Bus in Layer 2. It acts as a resource broker, enabling the system to search for resources and managing their interconnections. Another important functionality provided by the Bus is the internetworking capability amongst other C@R architectures to be able to establish a CWE. Since the bus supports the interoperability by using the OSOA design principles the C@R platform is able to cooperate with other tools, services or even platforms (including other CWE platforms as well). The CCS components in Layer 1, which are software components with a defined functionality, will be implemented as web services that register to the Bus via a well defined interface to offer their functionality to the CWE. Due to the interoperability of the Bus CCS components can be developed and used distributed amongst the participating RLLs to support reusability of CCS components. The Orchestration Capabilities (OC) in Layer 2 (context awareness, distributed workspaces, advanced services) act as components providing common services for all SCTs. Each OC will harmonise and integrate into a single model all the information related to the same issue, providing functionality in a uniform and transparent way. The SCTs, which are final user applications that serve the needs of the RLLs, will be described by Scripts and interpreted by the Composition Engine in Layer 2. These Scripts contain the specifications of all necessary elements (CCSs), workflow descriptions as well as the interactions and calls to the collaborative functions and libraries provided by the OC components.

In addition to the three basic layers a Web 2.0 and a semantic layer have been introduced to cope with the infrastructural and environmental impediments in rural areas.
The Web 2.0 layer is located in Layer 3 to improve the functionality and behaviour of the SCTs and their instantiated user applications in the RLLs. The semantic layer is a horizontal layer that provides functionalities to all three layers. A more detailed view on these advanced concepts is depicted in Figure 4.

The Web 2.0 technologies are located on the client side UI (see Figure 4, top) and communicate with the Adaptive Hypermedia System Engine [18, 19] which acts as an intermediate between client and server side. The semantic technologies (ontologies and semantic rules) build the knowledge base and processing engine on server side (see Figure 4, bottom). Some of the advantages when utilizing these advanced concepts are:

- **Context awareness**: The SCTs will be enhanced by Web 2.0 and semantic technologies to adapt itself regarding the current context (location, hardware used, user profile).
- **Automatic web service orchestration**: Improved web service discovery, invocation, composition and execution due to semantic web service annotation.
- **Improved usability**: Web 2.0 UI is adapted to the individual user requirements and thus copes with the less computer literacy in rural environments.
- **Optimized bandwidth usage**: Due to the bandwidth limitation in rural environments it’s key to reduce client-server communication using asynchronous calls.
- **Adaption of the UI/backend services in real time**: The application front-/backend adapts itself regarding the user behaviour/needs to support the user as much as possible.

4.2 Service mapping of a use case onto the architecture

This section validates the theoretical approach by mapping a real use case onto the C@R architecture. The mapping will be explained on the basis of the “Collaborative procurement and logistics” use case (see figure 2) – one example of a real business case in the South African RLL (for a more detailed description of this use case see [20]).
The collaborative procurement and logistics use comprises three main actors: Spaza Shop Owner, Infopreneur and Supplier. A Spaza Shop Owner sends his order via mobile phone (plain SMS or GPRS depending on if the mobile phone is java enabled or not) to the Infopreneur, who is a self-sustainable, micro, service entrepreneur providing information and data related services in rural areas. The Infopreneur uses his application suite to process all incoming orders (check, monitor, select, bundle and forward) and to send the bulked orders to the supplier. This use case now will be mapped onto the three layered C@R architecture (see figure 3).

Layer 1 accommodates simple CCS components like SMS communication, GPRS handler, Single Sign On, GIS/GPS as well as more complex CCS components like Order Creation and Order Maintenance which are itself composed by several simple CCS components.

Layer 2 contains the three already mentioned orchestration capabilities which provide services related to the same issue. Example services for the individual OCs regarding this use case might be:
- Context Awareness OC: User profile, Tracking user behaviour, Hardware detection
- Distributed Workspaces OC: Data storage, Data synchronization, Version control
- Advanced Services OC: Voice over IP, Multimedia content, Chat

In Layer 3 the end user applications (SCTs) are built by Scripts interpreted by the Composition Engine. Example SCTs in the context of the collaborative procurement and logistics use case might be a portal used by the infopreneur, a catalogue used by the Spaza Shop Owner on his mobile phone and a GIS (Geospatial Information System) application using GPS information to visualize the position and business information of a Spaza Shop (e.g. to plan the route for the product delivery from the Supplier to the Spaza Shop) to enable location based services.

Since the C@R architecture supports the internetworking between other CWE participants the CCS, OC and SCT components might be spatial distributed among the different participants.

5. Development steps

Several prototypes and proof-of-concept programs have been developed, and each of the main collaborative scenarios have been addressed in the first development cycle.
- Collaborative procurement and logistics: In this scenario several SMMEs place their orders with the Infopreneurs. The network of Infopreneurs then aggregate these orders, and use the larger order quantities as bargaining power to attain discounts. The bulk orders are also delivered via shared transport, bringing logistics costs down. The scenario reduces high transaction and logistics costs, and thus improves cash flow and
competitiveness for the SMMEs.

As technological solution a data model, a geospatial mock-up, and a mobile phone user interface for the SMMEs to place orders were developed.

- **Collaborative stock inventory management:** In this scenario the local community offers their goods to a wider market. The local SMMEs can put their inventory on the Internet, and attract inflowing currency to the community.

  A data model, a user portal, a data abstraction layer (to retrieve data from any database or web service), a mobile phone user interface for the ‘outside world’ to place orders, and a guesthouse booking system were developed.

- **Collaborative knowledge sharing and skills development:** In this scenario the Infopreneurs need efficient tools to develop, share, support, and query the collective intelligence of the various communities. One of the key aspects is to combine both informal collaboration (e.g. blogs, wiki, forums, chat, web articles, podcasting and SMSs) and formal collaboration (e.g. workflow-driven collaborative applications).

  These are all new, but well-known technologies. The innovative [17] variation is to adapt these technologies for use in the emerging economy circumstances specific to this RLL, e.g. offline usage and synchronization via erratic and costly connectivity.

  An elegant solution was developed for collaborative knowledge sharing.

  One of the lessons learned here was that breaking a scenario down to its ‘atomic’ technical subparts can help tremendously with simplifying the solution. During the analysis of the collaborative knowledge sharing scenario, we broke the various informal collaboration methods down to a few basics. A forum consists of a whole lot of paragraphs, all on one page, and may contain certain media, e.g. photographs. The page is a ‘thread’, which is part of a topic, which fits into a list or hierarchy of topics. The basic components are a page, paragraphs, media and a list of categories. We found we could break a wiki, a chat room, web articles and blogs down to these same generic components. By having just a few components, it became easy to develop an efficient offline and synchronization mechanism. We thus managed to adapt ‘always connected’ Web 2.0 technologies to local ‘mostly offline’ conditions.

  The next, and more difficult, step was to merge the various prototypes into a full end-to-end working scenario, based on the common C@R architecture described. The various prototypes were a mixture of HTML, PHP, Java, JavaScript, Flash, Mobile, PowerPoint, Access and MySQL databases. The use cases were taken, and mapped onto the OSOA layers. From this mapping a potential common service library and the various required services were uncovered. The common service library requires structured governance across the seven Living Labs. Coordination of development efforts from a high-level, yet technical viewpoint, is both necessary and tricky.

  Legacy systems are more often than not present in any environment. The impact of and on these legacy systems should be investigated and planned early in the design process. Wrapping the legacy functionality in a service-enabled wrapper is normally quite possible. Service-enabling the legacy systems to use the common OSOA architecture can be more of a challenge. Should you wish to migrate the legacy system to the new OSOA paradigm, proper migration planning is required to ensure minimal impact on the current users of the legacy system.

  Currently we are developing an automated procedure to help users recover from most computer-related disasters. Should media get corrupt, compromised by viruses, or software configurations go sour, the rural users need a fast and efficient method of restoring business
functionality quickly. The automated procedure will make use of intelligent scripts and compressed archives.

6. Results

The C@R project now is in the stage of advanced prototype development and end user interaction and thus there are important lessons learned and clear benefits of the C@R software approach in the area of architecture design and current IT research topics.

6.1 Lessons learned

Despite the still ongoing development of the C@R OSOA architecture there are already important lessons learned regarding:

**Advanced concepts:** Using Web 2.0 not in the common way to enrich the application by putting additional functionalities and visual effects into the UI just for a better look but using the technology to:
- reduce client-server communication to minimize the communication related transaction costs.
- reduce client-server communication and make use of asynchronous calls to cope with bandwidth limitations.
- use offline capabilities (e.g. DOJO offline toolkit [8]) to cope with connection problems.
- utilize the Web 2.0 visual capabilities to design an easy to use UI to cope with the lower levels of computer literacy.
- react on the user behaviour and context by adapting the application in real time.

Using semantic technologies to:
- support the interoperability between CWE participants by having a common metadata format.
- enable automatic web service orchestration by semantic web service annotation to enhance the web service discovery, invocation, composition and execution to enhance the performance of the OSOA approach.
- realize an adaptive hypermedia system to create a user centric and usability driven application.

**OSOA design:** In contrast to traditional SOA architectures which are built to operate in well equipped and well known environments (broadband internet connection, powerful machines and well known participants) the C@R OSOA architecture aims to be an interoperable approach that is able to:
- run in rural environments with connection limitations, low power machines and other typical infrastructural impediments.
- integrate each kind of legacy system.
- integrate new participants from other domains into the established CWE in an easy way.
- simplify the distributed development and instantiation of components (CCS, OC, SCT) used in the OSOA.
- provide the fundament for the use of collaboration tools and thus establish a common knowledge base.

6.2 Benefits of the C@R software approach

The C@R software approach will realize three reference implementations to provide best practices in emerging but currently unsolved important software areas. Due to the living lab approach an end user driven cyclic development process enables a parallel validation and
improvement of the implementation efforts to ensure the high quality of the reference implementations and thus to provide a proof of concept in a real world scenario. The following concepts will be realized as reference implementation:

Open Service Oriented Architecture (OSOA) reference implementation using open standards and SCA/SDO concepts.

Open Collaborative Architecture (OCA) [9] reference implementation which is the prerequisite for the establishment of a Collaborative Working Environment (CWE).

Reference architecture of a Service Oriented Architecture (SOA) aligned to the specific environmental, infrastructural and cultural conditions predominating in rural environments/emerging economies. This adapted SOA implementation will be the imperative necessity for the establishment of an advanced IT landscape in these areas.

Beside the architecture reference implementations the C@R software approach pushes the development and progress in important current IT research topics and provides best practices for:
- Automatic web service orchestration
- Semantic web service annotation
- Context aware solutions
- Location based services using mobile technologies (GIS,GPS)
- Mobile solutions using alternative technologies available in rural environments like Wi-Fi, GPRS and EDGE.
- Wireless mesh networks
- User centric and usability driven application design (UI as well as backend services)
- Adaptive hypermedia systems

6.3 Business benefits

The usage of the C@R architecture approach provides a clear business benefit by significantly reducing and simplifying the implementation efforts and thus decreases implementation time and costs. To illustrate how the C@R architecture approach practically realizes this benefit a completely new business case will be implemented exemplarily. The prerequisite for this example is that the C@R architecture is already established and several living labs participated to this CWE.

The further application example is in the area of health care which is a very important topic in rural environments. Since there are a lot of ill people in rural areas the rural community created an organization, the so called “home-based care workers”. These are volunteers of the local community who help the sick and the orphans. Many of the medical services are delivered by these “home-based care workers” caused by one big problem the clinics face - the lack of communication with the patients that can’t afford a cell phone. Thus one of the tasks of a “home-based care worker” is to visit the patients every day and check their shape and needs. If they need to go to the clinic the “home-based care worker” forwards this information to the infopreneur by mobile phone. Since the rural community consists of quite a number of ill people the infopreneur will get several requests for transportation of patients to the clinic. The infopreneur than organizes the transportation ordering a taxi (minibus) to pick up all the patients in the various areas.

To realize this use case we can reuse a lot of functionality (CCS, OC and SCT components) already developed in other living labs due to the interoperability capabilities of the C@R architecture. The services (CCS components) mentioned in the following list might be developed from different CWE participants and stored in spatial distributed
repositories connected together by the C@R architecture (especially the by Bus component).

- **SMS communication services** will be used to enable the SMS communication between “home-based care worker” and infopreneur
- **Location-based services/GIS services** will be used to localize the position of the “home-based care worker” and thus to localize the patient.
- Again **SMS communication services** will be used to inform the taxi driver which patients to collect by SMS.
- **GPS/GIS services** will be used to create the most appropriate route for the taxi driver to collect all the patients at the different locations. This route will be visualized by **Map visualization services**.
- **Map visualization services** will be used to visualize the route of the taxi driver and display it on his device (PDA, mobile phone, computer, …).

These are just some examples of reusable CCS components that could be used for the new business case implementation. That means that the basic functionality could be provided just by using existing spatial distributed CCS components.

In addition to the reusable CCS components the basic functionality regarding context awareness, distributed workspaces and advanced services is provided by the orchestration capabilities (OC).

- The **Context awareness OC** will be used to adapt the visualization of the taxi route regarding the device used to access the **Map visualization services** (e.g. adjust view to display solution). This might be a java enabled mobile phone, PDA, computer or any other device.
- The **Distributed workspace OC** might be used to keep documents like the patient record and patient address in sync. That means that the “home-based care worker” could access and download these documents for editing with his mobile phone and the **Distributed workspace OC** takes care of the synchronization with the original document.
- The **Advanced services OC** will be used by the **Map visualization services** to include pictures and videos of the area into the map.

The reusable CCS components in addition with the OCs provide the basic backend functionality to realize the new business case. To create a real end user application also the software collaboration tools (SCT) could be reused.

- **GIS application SCT** will be used to realize the visualization of the route used by the taxi driver. It is a flash based application for visualization of maps and routes.
- **Web portal SCT** will be used to combine the functionalities for the infopreneur to receive and process the SMS from the “home-based care workers”. It is a web portal implementation with a well defined API to include additional components and functionalities like CCS components and OCs.

The mentioned SCTs already include Web 2.0 functionalities like offline-capabilities, asynchronous calls, UI adaptation and so on. That means that they are already prepared for the usage in the rural environments coping with the specific impediments like unsustainable internet connectivity, low bandwidth or less computer literacy of end users.

Also the usage of the semantic layer is already integrated in the CCS components and OCs that make use of semantic technologies. Therefore no additional implementation efforts arise for the integration of Web 2.0 and semantic functionalities.

The example made clear the benefit of the C@R architecture approach. The layered architecture, the component based services and the well defined interfaces enable a modular
development of applications reusing already developed components from spatially distributed CWE participants.

7. Conclusions

The technologically advanced common OSOA approach, used in every RLL in the C@R project, has been adapted especially for use in the emerging economy circumstances specific to the South African RLL, while staying true to the needs, requirements and contexts of the local users and communities. Current status of advanced prototype development and the evaluation of end user interaction in the living lab indicate the importance and requirement for the C@R OSOA approach and verify the concepts and methodologies utilized.

The mapping of real use cases onto the C@R OSOA architecture approved the theoretical architecture approach and was a prerequisite for the development of C@R common and reusable services. The potential impact and benefit of the architecture has been continuously monitored by extensive user community interactions following the living lab approach.

Future work will entail:
- Leveraging the synergies within C@R RLL network to implement new scenarios in a shorter time than possible with alternative methods.
- Take end-user engagement to the next level with prototype validation and participatory design. How will the users’ input influence the application design?
- Investigate the scalability of the results. Will the results also work in rest of Africa and in other emerging economies? Which results are applicable in the EU?

References

[18] Peter Brusilovsky: Methods and techniques of adaptive hypermedia. HCII, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1996.