Contribution to flexible service coordination in mobile prosumer environments

Ramón Alcarria – PhD Dissertation

Tomás Robles – Advisor

November 2013
Outline

- Motivation, goals, and objectives
- Related work
- Contributions
  - Methodological approach involving prosumers in service development
  - Template personalization
  - Service fragmentation
  - Distributed service coordination
  - Resource access
- Validation
- Conclusions and Future Works
Motivation, goals, and objectives

Motivation

- Existing creation tools are not tailored to prosumer potential
- There are not any service development methodologies that explicitly take into account the new prosumer role
- Service execution in mobile environments implies coordination issues, lack of connectivity, arrival of new participants, changes in the availability of resources.
Motivation, goals, and objectives

Main goal

The provision of a service creation, distribution and execution model for mobile environments which allows non-programmer users, but experts in a given domain, to create and execute their own applications and services.
Motivation, goals, and objectives

Objectives

#1 Define the prosumer user in the context of the mobile prosumer environments
#2 Involve prosumers in the development of a service creation framework
#3,4 Define prosumer service lifecycle and supporting architecture
#5 Model user personalization decisions and their implications
#6 Integrate guidance mechanism for service creation and error detection functionality
Motivation, goals, and objectives

Objectives

#7 Define service decomposition mechanisms for distributing service fragments efficiently

#8 Establish an adaptive model of service fragment coordination and synchronization for the execution of distributed services

#9 Provide robustness features in the interaction between services against link losses and terminal connections and disconnections
Motivation, goals, and objectives

Objectives

#10 Integrate into the prosumer environment mechanisms to select at runtime the optimal resource for each service component

#11 Provide a communication infrastructure for uniform and efficient resource access in continuous service execution scenarios.

#12 Implement applications, prototypes and proofs of concept and validate the contributions in different domains related to mobile prosumer environments.
Related work

Service co-creation

- Introduced by *Prahalad and Ramaswamy (2000)*, emphasizes the generation and ongoing realization of mutual firm-customer value
- Value provision in the IT field is related to content provision and service provision.
  - Evolution from content prosumer to service prosumer
- Several disciplines defining the relationship between users with creating intentions and professional developers:
  - End-user programming, end-user development (*Lieberman, Paternò, & Wulf, 2006*), end-user software engineering (*Ko et al., 2011*)

Definition of prosumer and service models
Related work

Service creation methodologies considering user involvement

- Software methodologies for user-centered design (Agile, NSD) does not consider the prosumer role.
- Current creation frameworks for end-user development are mashup based, data driven and with a single creation strategy.
- We study customizable services (Rosson et al., 2010), WYSIWIG (Shroff, Agarwal, & Devanbu, 2009), workflow based service composition (Laga, Bertin, & Crespi, 2008), and assisted composition (Chowdhury, Rodríguez, Daniel, & Casati, 2011).

- Proposal of method for development of a service creation framework, tailored to prosumer characteristics
- Contribute to service personalization strategy
Related work

Service distribution

- Mobile prosumer environments

Motivation of service distribution:
- Efficiency and privacy involved in processing the information from the terminal that generates it
- Reinforcement of the sense of ownership and authorship of prosumers when providing their services from their own mobile phones
- Factor studied in some end-user development works *(Gerhard Fischer, 1994)* *(G. Fischer, Giaccardi, Ye, Sutcliffe, & Mehandjiev, 2004)*

Prosumer model

Distributed execution of mobile services
Related work

Service distribution

- Related work from workflow decomposition, model transformations and activity distribution.
- We highlight the works by Vanhatalo et al. (2007–2008) and Polyvyanyy et al. (2010–2012) of workflow decomposition
- Problems:
  - lack of granularity of SESE fragments

▸ Contribute to service decomposition into fragments using mechanisms for efficient service distribution
Related work

Service coordination

- Need to specify the control flow of distributed services
  - Workflow patterns from *Aalst, Hofstede, Kiepuszewski, and Barros (2003)*
  - Define logic gates, based on workflow patterns to establish and optimize communication channels
- Need to decouple producers and consumers of information to solve coordination problems
  - Propose the publish-subscribe communication model to ensure functional decoupling in time and space.
  - Use dissemination techniques to resolve problems related to link losses, and participant discovery and disconnections.
Related work

Continuous service execution

- Runtime adaptation to changes in the availability of resources is not very tackled for mobile middleware (Lopez-de-Ipiña et al. 2010)

  ➢ Middleware solution for continuous service execution in mobile environment

Communication with unknown services

- Via messaging (XML-RPC, JavaRMI, COAP) or via web technologies (WS-BPEL)

  ➢ OSGi solution based on direct bundle communication (better performance)
Contributions

- Methodological approach involving prosumers in service development – Section 4 & 5
- Template personalization – Section 6
- Service fragmentation – Section 7
- Distributed service coordination – Section 8
- Resource access – Section 9
- Validation – Section 10
Methodological approach involving prosumers in service development

Contributions

- Prosumer methodology – Section 4.1 & Section 4.3
- Service lifecycle – Section 5.2
- General architecture – Section 5.3

Section 4 & 5
Methodological approach involving prosumers in service development

Prosumer methodology – Section 4.1

- Roles
- Actions

Roles
- Framework provider
  - develop framework
- Domain expert
  - provide creation goals
- Prosumer
  - create/consume
- Consumers
  - consume

Actions
- Framework generation
- Service 1
- Service 2
- Service 3
- Prosumer framework
- Supporting applications

Contributions
Methodological approach involving prosumers in service development

Prosumer methodology – Section 4.1

Contributions
Methodological approach involving prosumers in service development

**Prosumer methodology – Section 4.3**

- Two views, depending on the responsibilities of each role, the prosumer’s view and the developer’s view
- Based on Agile technologies and NSD

![Diagram showing lifecycle stages and views]

Contributions
Methodological approach involving prosumers in service development

Service lifecycle – Section 5.2

- Template: previous state of a service. It is transformed to a final service through an adaptation process.
- Each process is defined in SPEM
Contribution: General architecture – Section 5.3
Methodological approach involving prosumers in service development

Conclusions

- The prosumer methodology considers the creation of prosumer services using a creation framework.
- The service lifecycle involves the processes of creation, classification, publication, search, adaptation, execution and verification.
- Supporting tools for these processes determine the general architecture of the service provision and consumption framework.
Template personalization

Section 6

Context

- We explore the service creation strategy based on service personalization, for the adaptation process.

Contributions

- Define a variability model supporting the interdependence between user decisions in the service personalization process
  - Implement the notion of constraint in the variability model – Section 6.2
- Develop a semiautomatic generation of a personalization wizard – Section 6.3
- Develop a consistency checker – Section 6.4
Template personalization

Variability model (Section 6.2)
Template personalization

Implementing the notion of constraint in the variability model – Section 6.2

- Dependencies in variability points and options are defined in the model.
- We define a language that relates personalization actions with the automatic realization of other actions.
- A declarative language based on LTL (Linear Temporal Logic) is used to describe the activities that can be performed and the constraints prohibiting undesired behavior.
Template personalization

Language for defining dependencies in user decisions

- We define de events: OS (option selection), PD (point delegation) and PO (point omission), as LTL propositions

\[ \phi_1: \Box PD_{P1} \rightarrow \neg \Box PD_{P2} \]

Delegating \( P_1 \) to the runtime phase blocks any other delegation of \( P_2 \)

\[ \phi_2: \Box (OS_{P1} \rightarrow \bigcirc OS_{P3}) \]

A configuration of \( P_1 \) should be immediately followed by a configuration of \( P_3 \)

Contributions
Template personalization

Develop a semiautomatic generation of a personalization wizard – Section 6.3

- Guides the user to personalize every variability point, taking into account dependencies in user decisions.
- Variability points must be ordered according to the dependencies among them in order to first show the user options with none requirements.
- This sorting is done with a topological sorting algorithm based on depth-first search and runs in linear time.
Template personalization

Algorithm 1. Topological sorting algorithm for VP dependencies

Main function main()
Input: L: empty result list with sorted nodes, P: Unordered set of all nodes
Output: L: result list with sorted nodes
1. getIndependentNodes();
2. for each node i, from I
3. visit (i);

function getIndependentNodes ()
1. Def: I: empty list for unrequired nodes
2. For each p_i from P
3. If (post(p_i)={Ø}) add p_i to I;

function visit (n)
1. If (n is visited) return;
2. Mark n as visited;
3. for each p_i from P where p_i=pre(n);
4. visit (p_i);
5. Add n to L;
Template personalization

Topological sorting algorithm for VP dependencies

- VP dependencies are seen as a node tree, where node links express dependencies.
- Independent nodes are saved in the output list, releasing the parent node.
- Cyclic dependency should be found and extracted with an algorithm such as the one provided by Tarjan, 1972.
Develop a consistency checker – Section 6.4

- Informs the user if the performed action generates an inconsistency, allowing more flexibility in user decisions.

- We define a set of constraints $C := \{\phi_1, \phi_2, ..., \phi_n\}$ and a set of personalization actions $\Pi$.
- $\Pi$ is satisfied if $\forall \phi_i \in C$: the set of actions of $\pi$ makes $\phi_i$ evaluate to $true$.
- $\Pi$ is temporarily violated if $\exists \phi_i$ evaluated to false, but a longer trace $\Pi' \supset \Pi$ is satisfied.
- $\Pi$ is permanently violated if $\exists \phi_i$ evaluated to false, but there isn’t a longer trace $\Pi' \supset \Pi$ that can be satisfied.
Template personalization

Conclusions

- Application to a service personalization environment for drug management in hospital pharmacy scenario
- Personalization wizard that hides, darkens and adds new options and variability points.
- Consistency checking mechanism based on ECA rules.
  - Requirements:
  - The service must contain at least one component of type Event, Condition and Action (existence).
  - $C_{in}$ should occur immediately after $E_{out}$, $A_{in}$ immediately after $C_{out}$ and $E_{in}$ immediately after $A_{out}$ (immediate response).
Contributions
Template personalization

Conclusions: consistency checker

- With these requirements we define a set of constraints:
  \[ \phi_1 := \phi_{1E} \land \phi_{1C} \land \phi_{1A} = \Diamond E_{out} \land \Diamond C_{out} \land \Diamond A_{in} \]
  \[ \phi_2 := \phi_{2E} \land \phi_{2C} \land \phi_{2A} = \Box (E_{out} \rightarrow \Box C_{in}) \land \Box (C_{out} \rightarrow \Box A_{in}) \land \Box (A_{out} \rightarrow \Box E_{in}) \]

- And a subsequent finite state automata
Template personalization

Conclusions: consistency checker

- We implement these constrains in our environment. If the configuration is satisfied, once the user clicks the save button, all the displayed components are surrounded with a green glow.
Service fragmentation

Section 7

Context

- Mobile prosumer environments are distributed environments for the execution of mobile services.
- Lack of granularity of SESE fragments, cost models for fragmentation not defined

Contributions

Fragment identification technique based on SPQR-tree decomposition – Section 7.2

Activity distribution algorithm using process model transformation techniques over a cost model – Section 7.3

Validation through application in a case study - Section 7.4
Service fragmentation

Fragment identification technique based on SPQR-tree decomposition – Section 7.2

- From BPMN to Workflow graph
Service fragmentation

Fragment identification technique based on SPQR-tree decomposition – Section 7.2

- From SESE to SPQR-tree
Service fragmentation

- Fragment identification: From SESE to SPQR-tree

Definition of the aggregation function $f : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}'$

(a) Sequential aggregation
(b) Parallel aggregation

Contributions
Service fragmentation

Activity distribution algorithm using process model transformation techniques over a cost model – Section 7.3

- Our model associates a normalized value [0..1] to each edge connecting a pair of activities, which represents the communication cost
- $distribCost(a_1, a_2)$ returns a normalized real number, indicates the degree of difficulty to separate two activities
- We feed the algorithm with information about activity execution costs and the information about the activities that are planned to be fixed
Service fragmentation

Activity distribution algorithm using process model transformation techniques over a cost model – Section 7.3

Contributions
Service fragmentation

Validation through application in a case study - Section 7.4
Service fragmentation

Validation through application in a case study - Section 7.4

Contributions
Service fragmentation

Conclusions

• Service fragmentation is needed for distributed execution of prosumer services

• We propose a SPQR-tree technique for workflow fragment identification, offering more granularity than SESE decomposition.

• We define the aggregation function and use it in an activity distribution algorithm considering a cost model.

• We validate the algorithm with a real BPMN workflow and propose the transformation to BPEL.

Contributions
Distributed service coordination

Context

- Coordination is required in mobile prosumer environments, where services are distributed.
- Orchestration approaches considers an execution model in which the services are fully specified before runtime.
- Choreography approaches introduces the challenge of an emergent behavior of the services.

Mixed approach

- Service Orchestration in each entity
- Service Choreography between fragments
Distributed service coordination

Contributions

- Adaptive model for service fragment coordination – Section 8.3.2 & 8.3.3
- Runtime coordination with recovery against falls – Section 8.3.4
- Support discovery of new participants – Section 8.3.5
Distributed service coordination

Adaptive model for service fragment coordination

- Definition of logic gates based on workflow patterns
- Channel creation and optimization

Contributions – Section 8.3.2 & 8.3.3
Distributed service coordination

Runtime coordination with recovery against falls

- At runtime, control events are transmitted through the created channels.
- Pub/Sub is used to implement logic gates.
- Robustness features in the interaction between services against link losses and terminal connections and disconnections.
- Dissemination techniques based on Gossip algorithms.

Section 8.3.4
Distributed service coordination

Support discovery of new participants – Section 8.3.5

- Definition of two aggregation modes:
  - Static scope aggregation
  - Instance-based aggregation
    - A new service instance is defined for the new participant
    - Aggregation of this new instance in the shared topic tree
Distributed service coordination

Conclusions

- We manage workflow coordination by Pub/Sub models
- We compare the performance of standard Pub/Sub solution and Gossip-based solution regarding performance against falls and channel creation.
  - Dissemination overcomes falls more efficiently (dynamic path rerouting)
  - The gossip dissemination mechanism converges before the technique based on recursive search of unvisited neighbors implemented in the standard Pub/Sub solution
Resource access

Section 9

Context

- Capabilities are heterogeneous in their technology, access protocols and communication models
- The availability of capabilities is continuously changing in mobile environments

Contributions

- Communication with unknown resources – Section 9.2
- Middleware for continuous service execution – Section 9.3
We use the OSGi technology to enable flexible composition of services that try to access to unknown bundles.

Technical difficulties in OSGi to recognize new services:
- How consumer bundle receives the signature of the service
- How consumer bundle imports the service package and invokes it

We define three reference architectures to overcome these problems:
- Specific Service: Uses the Java reflection library for bundle invocation and a SDL document for service signature discovery
- Common Service: Use dynamic proxy generation
- WSIF Invocation: Automatic generation of WSDL interfaces for unknown services. Bundle invocation through WS composition (using Apache WSIF)
Resource access

Communication with unknown resources – Section 9.2

Conclusions

• To support communication with unknown resources
  • We propose the OSGi technology for resource access.
  • We define the SS, CS, WSIF architectures to solve OSGi limitations for managing unknown bundles
  • We evaluate these architectures theoretically and practically, by a performance analysis.
  • CS for remote invocation environments
  • SS is the best default option regarding performance
Resource access

Middleware for continuous service execution – Section 9.3

- Capabilities are heterogeneous
  - Propose a middleware solution for homogeneous capability access
  - Heterogeneity of communications is solved by the definition and integration of communication paradigms
  - Heterogeneity of protocols and data is solved by the support of connection drivers.

- Availability of capabilities continuously changes
  - Runtime capability change
Resource access

Prototype implementation

• Capability access middleware
  • Integration of Request/Reply, QP2P and Publish/Subscribe communication paradigms.
  • Implementation of communication Drivers for REST, Bluetooth (RFCOMM and OBEX), SOAP/DLNA and Java for local capabilities
  • Java ME, tested in Nokia N97
Resource access

**Middleware for continuous service execution** – Section 9.3

**Conclusions**

- To support continuous service execution we develop a middleware solution
  - Integrating several communication paradigms, and mechanisms for runtime paradigm change
  - We use software design patterns support requirements of strategy-based resolution, low coupling, asynchronism, reusability and efficient resource management
Validation

- Application of the thesis contributions to various application scenarios:
  - Prosumer concept, service lifecycle and supporting architecture (#1, #3, #4)
  - Method for the development of prosumer services (#2)
  - Service personalization (#5)
  - Wizard and consistency checker (#6)
Validation

- Application of the thesis contributions to various application scenarios:
  - Workflow decomposition de service distribution (#7)
  - Distributed service coordination (#8, #9)
  - Continuous service execution (#10)
  - Communication with unknown resources (#11)

Section 10

Emergency situations

The new intelligent universe
Validation

- The new intelligent universe
  - The mobile device is the gateway for the interaction with available services and capabilities in a digital environment
  - Analysis of different service creation strategies (service composition, WYSIWYG)
  - Implementation of the prosumer service lifecycle in the context of the mIO! project (2008-2012)
Validation

- Hospital services
  - Objectives:
    - Validate the methodology involving prosumers in framework development
    - Evaluate the service adaptation strategy

- Development of the GISAI-Pharma tool
  - Involving domain experts in their design.
  - Mashup-based
  - Allows prosumer users (pharmacy workers) to create, personalize, share and consume location-based services, stock control, notifications and alerts for drugs and other assets present in the pharmacy
Validation

- Hospital services
  - User perception in a case study
    - Ten first year students of telecommunication engineering
    - Created three services to solve three specific problems and then they adapted other three services to solve another three specific problems
Validation

- Hospital services
  - User perception in a case study
    - Usefulness of the prosumer framework -> 50% of the participants gave the highest score
    - Usefulness of service adaptation -> 60% of the participants gave the highest score
    - Average quality of all services

![Quality of services created](image1)

![Quality of services adapted](image2)
Validation

• Emergency situations
  • Objectives:
    • Test the contributions related to coordinated execution of services between unknown devices and which suffer disconnections
    • Demonstrate that the proposed solutions, called Standard Pub/Sub and Gossip-based, behave as expected

• To simulate topologies with a large amount of nodes we use the ns-3 simulator with a virtualization technique called containers.
• We use MQTT protocol for Pub/Sub
• All the network topology parameters are provided by using a script that defines the nodes, network interfaces, connections and applications (Section 10.4)
Validation

- Emergency situations: Pub/Sub vs Gossip
  - Validation of channel creation phase:
    - If there is only one path between the publisher and the subscriber, so the Pub/Sub solution is 17% faster than Gossip solution
    - If we use a simple recursive search in MQTT to support multipath for Pub/Sub solution, Gossip converges in less hops.
  - Validation for runtime coordination phase with link failures

![Graph showing publication delay vs link loss]
Conclusions

- The prosumer model is a new way of creating and consuming services in mobile environments.
- We develop a methodological approach for the development of a service creation framework.
- Focusing in the service personalization process, we model user personalization decisions and their implications:
  - The variability model defined implements the notion of constraint between personalization decisions.
  - Enables the generation of a personalization wizard and a consistency checker.
- We propose a service fragmentation technique based on activity distribution.
Conclusions

- We create an adaptive model for service coordination, based on channel establishment between logic gates
  - We support recovery against falls and arrival of new participants
- We contribute to resource access in mobile environment
  - OSGi-based solutions for communication to unknown resources
  - Communication middleware for continuous service execution
- The proposed contributions have been validated through three real scenarios
  - The new intelligent universe
  - Hospital services
  - Emergency situations
Future works

Searching, modeling and validating strategies to improve prosumer success

- What is prosumer success and which are its metrics?
  - Some measurable parameters: people involved, competition degree, service creation difficulty, time required for training, expected pay-off (reduced future cost, fun, knowledge, interest)
- How to select the best domain expert?
  - Selection criteria: capability of the expert to gather information in the entire domain, time limitation, motivation and interest to contribute.
  - Other approaches: having an engineer with domain knowledge role-play as the customer
- Keep the motivation of the prosumer alive
  - Until the value of prosumerization becomes apparent and can be measured (Extrinsic vs intrinsic motivation)
Future works

Improve validation of GISAI-Pharma tool

- Factors that influence a correct service creation and adaptation?
- How experience influences service creation and adaptation?
- How to improve the training of the participants for the creation of better quality services?
- How collaboration among participants affects to the quality of services?
- More people involved and an experiment in a pharmacy department are needed.
Publications

JCR-indexed


Journal peer reviewed with ISBN/ISSN (non-JCR-indexed)


Publications

International peer reviewed conferences with proceedings


Publications

International peer reviewed conferences with proceedings (cont.)


Publications

International peer reviewed conferences with proceedings (cont.)

Publications

International peer reviewed conferences with proceedings (cont.)


Publications

National peer reviewed conferences with proceedings (Spanish)


Contribution to flexible service coordination in mobile prosumer environments

Ramón Alcarria – PhD Dissertation
Tomás Robles – Advisor

Thank you!
Summary of contributions

✓ Methodological approach for the development of a service creation framework
✓ Variability model for personalization process. Generation of a personalization wizard and a consistency checker
✓ Service fragmentation technique based on activity distribution
✓ Service coordination based on channel establishment between logic gates, supporting recovery against falls and arrival of new participants
✓ Resource access in mobile environments (OSGi and mobile middleware)