City University London

Discovering Services from Requirements: The SeCSE way

Neil Maiden

Centre for HCI Design

Overview

Service-centric systems
- Implications for requirements engineering

SeCSE components
- Research activities
- Requirements process
- Requirements and service discovery modules
- Formative evaluation

Service discovery algorithm
- Term disambiguation and expansion procedures

Ongoing work and opportunities

Service-Centric Systems

Emerging paradigm

- Integrate independent web and software services over internet through well-defined interfaces
- Market worth $11 Billion by 2008 (Leavitt 2004)
- >50% companies working on web service projects

Impacts on requirements processes

- But service-centric computing only recently recognized in RUP or in requirements research

EU-funded SeCSE IP

- €10m 4-year project on service-centred systems

City University London
Centre for HCI Design

SeCSE - secse.eng.it

Mission statement

- "*Create new methods, tools and techniques for systems integrators and service providers that support the cost-effective development and use of dependable services and service-centric applications*"

Four activity areas

1. Service engineering: specification of services
2. Service discovery: discovering and retrieving services at development, deployment and run-time
3. Systems engineering: service-oriented architectures
4. Service delivery: deploying, monitoring and switching services

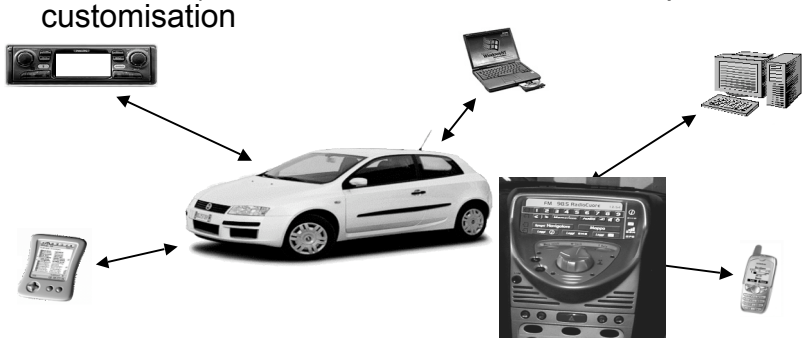
Industrial evaluation and application

- Fiat, DaimlerChrysler, Telecom Italia, Telefonica, Computer Associates, Microsoft, ATOS, Engineering

Example of Industrial Context

Fiat's customer care service centre:

- Service centre acts as service integrator
- Composing services for Fiat car owners e.g. navigation, mobile office, remote maintenance (after vehicle breakdown), remote mail service activation, profile customisation

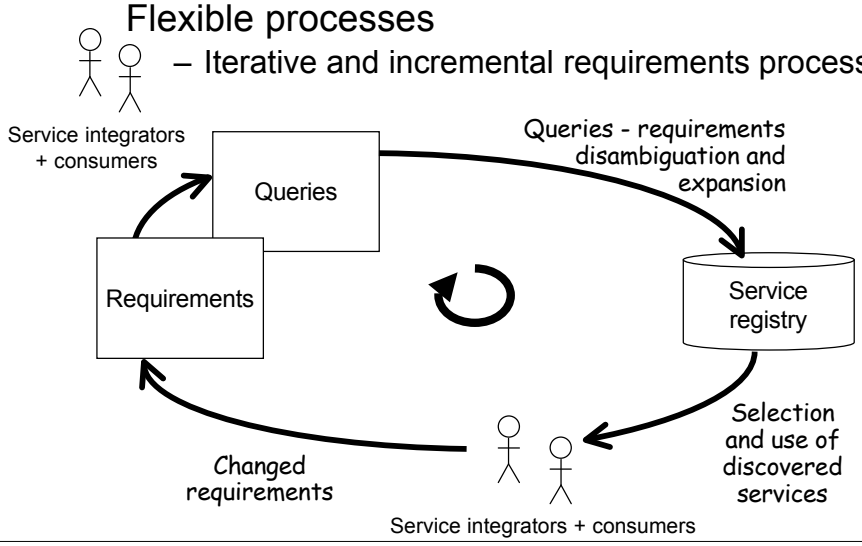


City University London
Centre for HCI Design

SeCSE Requirements Process

Flexible processes

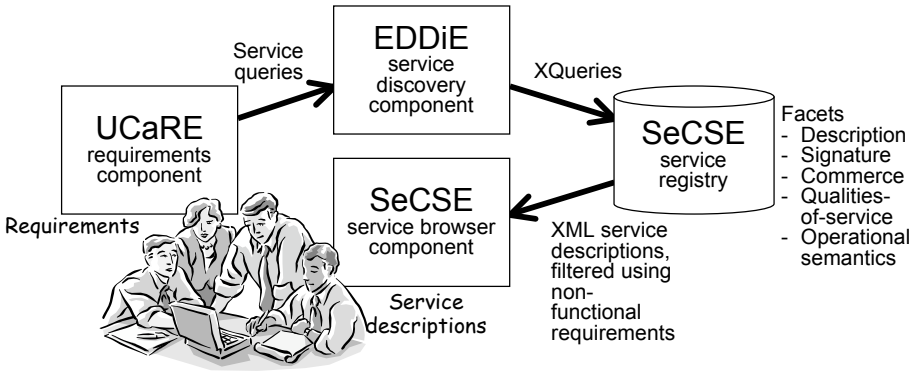
- Iterative and incremental requirements process



Four Key Modules

Web-based application

- .NET implementation using a three-layer model



City University London
Centre for HCI Design

Automotive Domain Example

Customizable telematics devices in FIAT vehicles

- Services for repair diagnostics, navigation, spares, etc

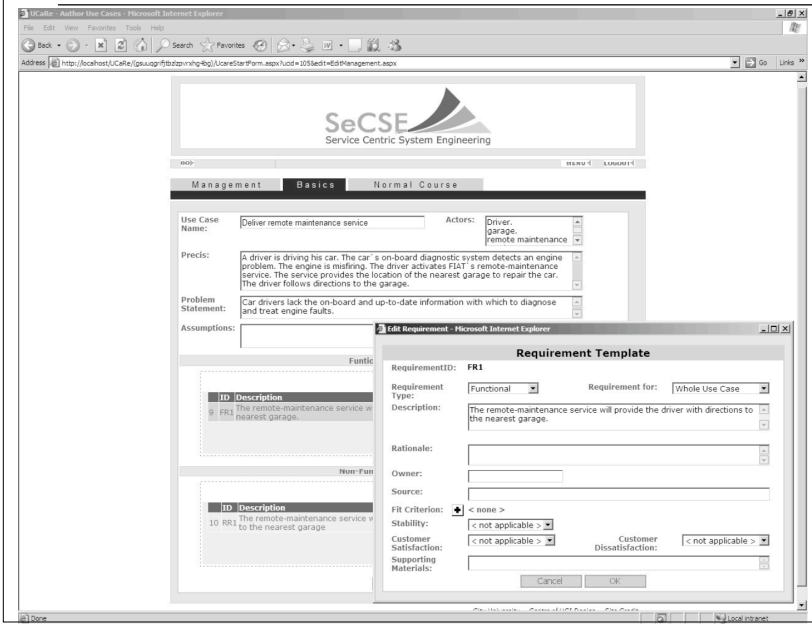
Initial use case precis

- A driver is driving his car. The car's on-board diagnostic system detects an engine problem. The engine is misfiring. The driver activates FIAT's remote-maintenance service. The service provides the location of the nearest garage to repair the car. The driver follows directions to the garage

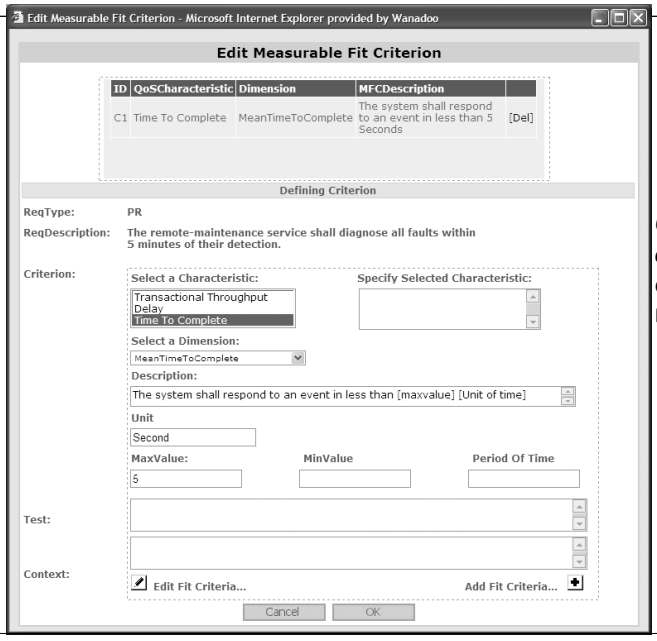
Initial requirements

FR1: The remote-maintenance service will provide the driver with directions to the nearest garage.

RR1: The remote-maintenance service will provide the driver with reliable directions to the nearest garage.



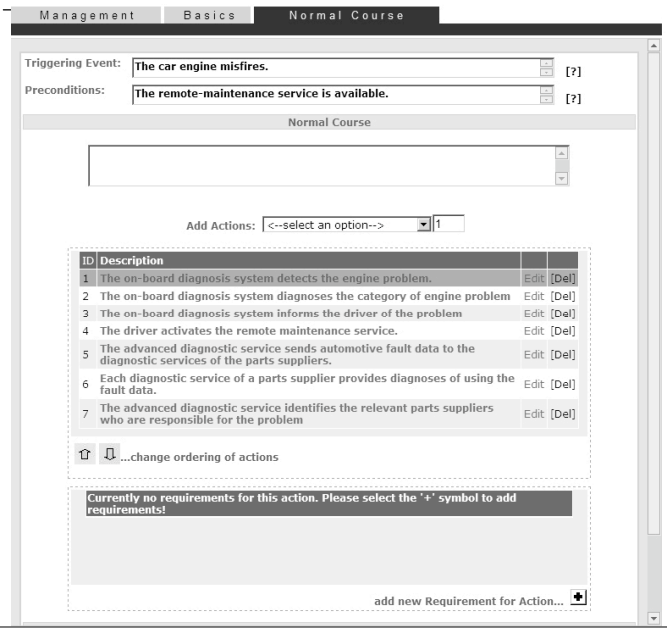
Use case-based specification of requirements



Ontology-based quantification of requirements

City University London

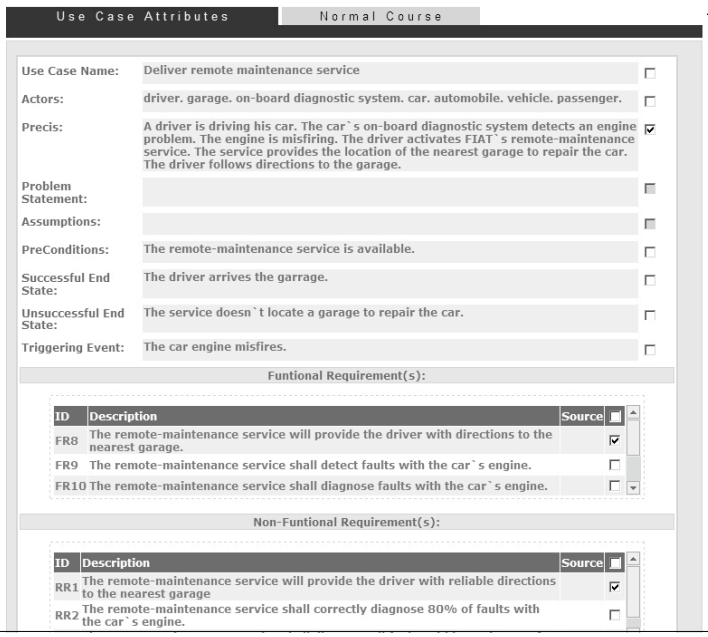
Centre for HCI Design



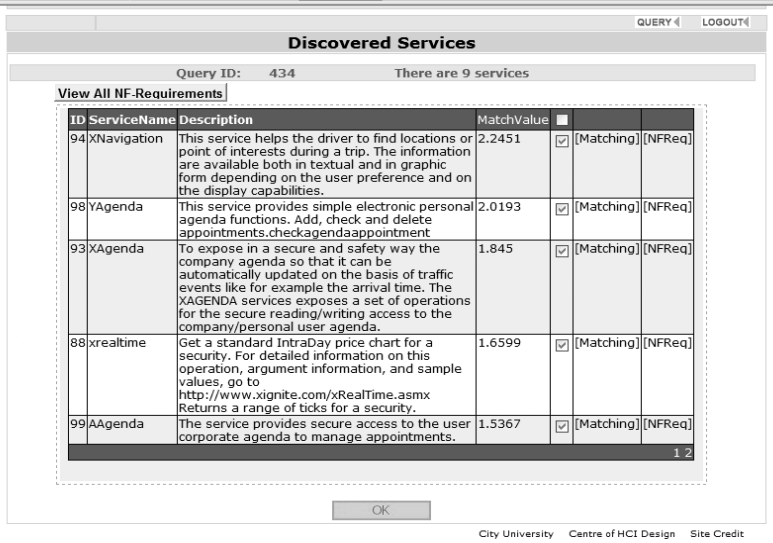
Manipulation of use case specifications

City University London

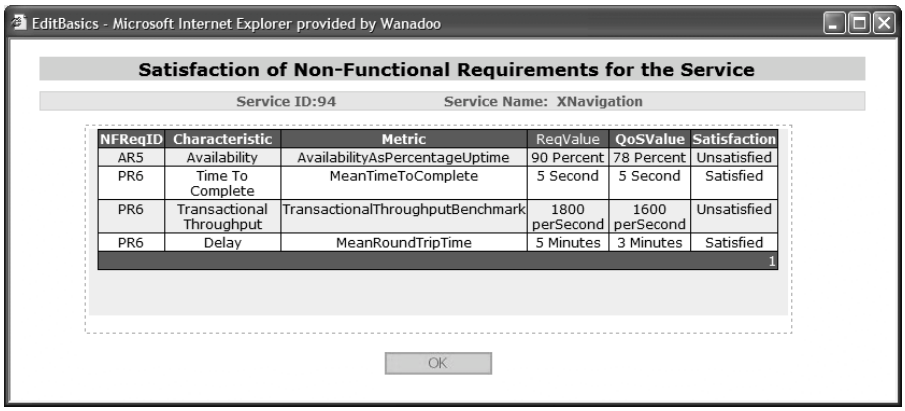
Centre for HCI Design



Seamless formulation of service requests



Retrieved services



Filter services using non-functional requirements compliance

City University London
Centre for HCI Design

Do these Modules Help?

Evaluated UCaRE and Service Browser

- Ran half-day automotive requirements workshop with 4 analysts - 2 from Centre Research FIAT, 1 from DaimlerChrysler, 1 from Computer Associates

Workshop in 3 phases

1. Walked through use case to discover requirements
2. Retrieved services from registry of 112 service descriptions
3. Walked through retrieved services to discover requirements not found using earlier walkthrough

Outcomes from Requirements Workshop

Workshop basics

- Phase 1 lasted 60 minutes: 27 requirements specified
- Phase 2 lasted 10 minutes: 11 services retrieved
- Phase 3 lasted 50 minutes: 8 services retained as relevant: 20 requirements specified

Retrospective questionnaire results

		R	M	S	P
Average importance rating	Stage1	2.7	2.3	2.0	2.6
	Stage2	2.3	2.1	1.9	2.2
Average novelty rating	Stage1	1.3	2.0	1.6	2.4
	Stage2	1.7	2.2	2.3	2.9

City University London
Centre for HCI Design

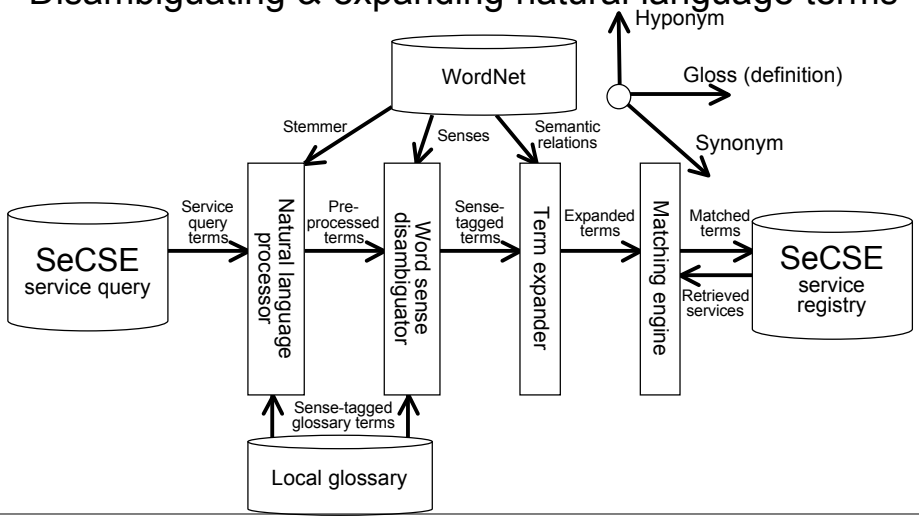
Recurring Requirements Patterns

Based on post-workshop analysis

- Requirements expressed new system features that were a consequence of implementing retrieved service
- Some expressed refinement of features of discovered service applied to the new system
- Some expressed required inputs to an application that implemented the discovered service
- One expressed a function that had the potential to satisfy service qualities described in service description
- Two could be linked through some shared deeper concepts rather than through input, output and consequence relations, and two had no discernible similarities with triggering service descriptions

EDDiE's Service Discovery Algorithm

Disambiguating & expanding natural language terms



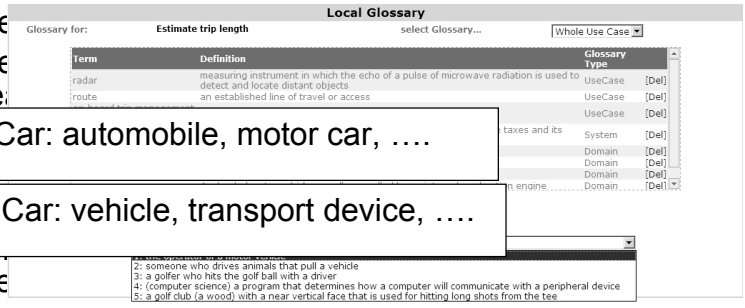
Disambiguation Strategies

Determine correct WordNet sense of each term

- Essential for effective expansion of query terms

Seven procedures - increasing cost to apply

1. Selecting senses from glossary
2. Sele
3. Sele
   alre
4. Se
   alr
5. Se
6. Co
   of te
7. User selects the correct sense during service discovery



Car: automobile, motor car, ….

Car: vehicle, transport device, ….

Expansion Strategies

Expand service queries with more terms

- Increases likelihood of discovering services not expressed using identical terms

Cannot rely on problem domain ontologies

- Nature of requirements, use WordNet as ontology

Three term expansion procedures

1. Synset expansion, with terms with similar meaning
2. Hypernym expansion, with terms with more generic meaning
3. Gloss expansion, with selected terms from definition of the original term

Driver: operator, vehicle, ….

City University London
Centre for HCI Design

Query Matching

Expanded queries expressed as XQueries

- Uses traditional vector-space model to compute semantic distance between query and service description
- Terms assigned weights according to originality and frequency of occurrence
- Computes single measure of semantic distance for each retrieved service description

Current algorithm is simple

- Further refinement within industrial evaluation

Ongoing Work and Opportunities

Delivered as part of SeCSE v2.0

- Due end November 2006, available for evaluation
- Registry crawler module, computing granularities of services in the registry
- Query decomposition to increase likelihood of discovering available services
- Creativity module, exploiting retrieved services to discover new requirements
- Process extensions to derive non-functional requirements

Extensive industrial validation

- With industrial partners including FIAT and Daimler-Chrysler
- Exploring validation and integration with IBM Rational