Service-Centric Systems and Requirements Engineering

Luciano Baresi, Neil Maiden, Peter Sawyer

Politecnico di Milano, City University London, Lancaster University

Why?

- Different communities
  - Requirements engineering
  - Service-centric computing
  - Little communication, let alone collaboration
- Shared concepts
  - Requirements monitoring, quality-of-service, service discovery queries
- But missed research and practice opportunities
- The SeCSE project

Mini-Tutorial Agenda

- Services and service-oriented systems
- How each can influence the other
- Some SeCSE solutions: processes and software
  - Publishing services based on provider specifications
  - Discovering services that meet consumer needs
  - Monitoring services for compliance
- Future trends
- Questions

SeCSE

- Project Acronym: SeCSE
- Project Title: Service Centric Systems Engineering
- Project instrument: Integrated Project
- Consortium: 15 organisations from 6 countries
- Thematic area: Open development Platforms for software and services
- Duration: 48 months (important results from every year)
- Budget: 15,2 MEuro (Funding 9.2 MEuro)
- Partners include: Engineering, Fiat, Telecom Italia, Telefonica, ATOS Origin, Microsoft, CA, Politecnico di Milano, City University London, Lancaster University

SeCSE
Service Centric System Engineering

Services and Service-Oriented Systems

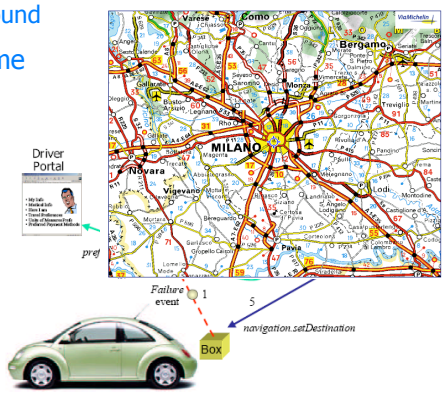
Neil Maiden

The shift towards dynamic environments

- Closed world environments were unrealistic
  - Dynamic, open environments become the norm
  - Requirements cannot all be discovered upfront
  - Many unforeseen stakeholders emerge
- Flexible support for change
  - Incremental, agile and prototype-based approaches
  - Modular, distributed design; changes need recompilation and redeployment
  - Information hiding, encapsulation, interface versus implementation; all need new languages
  - Component-based software
- Critical in emerging domains
  - Ambient intelligence, context aware applications, pervasive computing, web 2.0

Autonomic systems

- The system needs to change when the context changes
  - Some of its parts can disappear
  - Some new parts can be found
- Re-organize itself at run-time



The open-world assumption

- One remaining assumption
  - We own the software modules and components
- Services become key actors in open-world systems
  - Resources made available on a network as services with which our software can interact remotely to obtain a goal
  - Loosely coupled
  - Accessed on demand
- Key differences
  - Services are owned by other people and organizations, and not under our jurisdiction
  - Contractual arrangement between service consumer and provider

Services

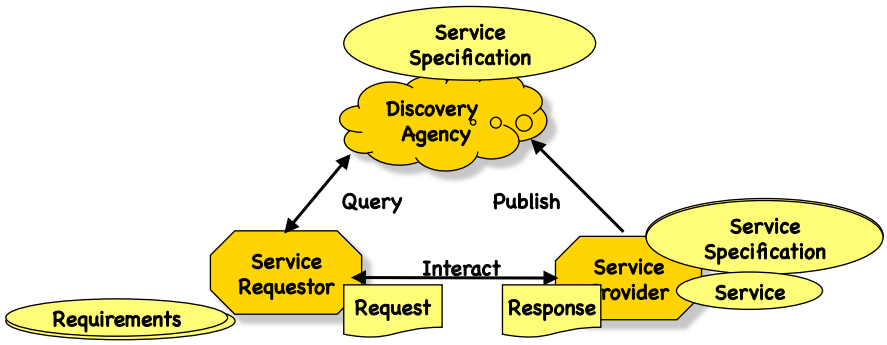
- Services are self-describing, open components that support rapid, low-cost composition of distributed applications
- Service providers procure the service implementations, supply their service descriptions, and provide related technical and business support
- Since services may be offered by different enterprises and communicate over the Internet, they provide a distributed computing infrastructure for both intra and cross-enterprise application integration and collaboration
- Service descriptions are used to advertise the service capabilities, interface, behavior, and quality

M.P. Papazoglou, D. Georgakopoulos (eds.): Special Issue on Service Oriented Computing. Communications of the ACM 46(10), 2003

Example of Services in Dynamic Environments

- 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

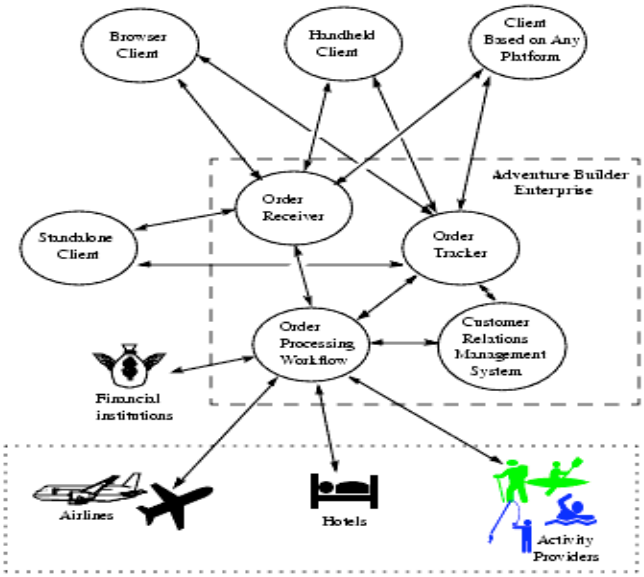
Service-oriented systems



Service composition

- Main development task in SoAs
  - Applications created by combining building blocks provided by services
  - Service compositions may themselves become services
  - Recursive service composition
- Service composition should
  - Use functional requirements
  - Be based on quality-of-service parameters
  - Use P2P conversational interactions
  - Exploit multi-party interactions
- Many composition models are possible and available

Integrating heterogeneous systems



Is this the whole story?

Business Domain Specific extensions	Various	Business Domain
Distributed Management	WSDM, WS-Manageability	Management
Provisioning	WS-Provisioning	
Security	WS-Security	Security
Security Policy	WS-SecurityPolicy	
Secure Conversation	WS-SecureConversation	
Trusted Message	WS-Trust	
Federated Identity	WS-Federation	
Portal and Presentation	WSRP	Portal and Presentation
Asynchronous Services	ASAP	Transactions and Business Process
Transaction	WS-Transactions, WS-Coordination, WS-CAF	
Orchestration	BPEL4WS, WS-CDL	
Events and Notification	WS-Eventing, WS-Notification	Messaging
Multiple message Sessions	WS-Enumeration, WS-Transfer	
Routing/Addressing	WS-Addressing, WS-MessageDelivery	
Reliable Messaging	WS-ReliableMessaging, WS-Reliability	
Message Packaging	SOAP, MTOM	
Publication and Discovery	UDDI, WSIL	Metadata
Policy	WS-Policy, WS-PolicyAssertions	
Base Service and Message Description	WSDL	
Metadata Retrieval	WS-MetadataExchange	

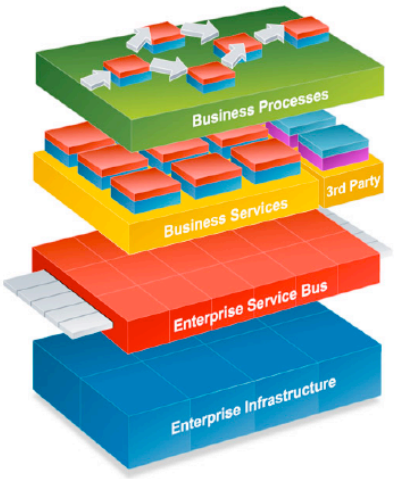
Business domain
Management
Security

Portal/presentation
Business processes

Messaging

Metadata

Service-Oriented Architectures (SoAs) and the ESB stack



Enterprise Service Bus

- **Web Services**
  SOAP, WSDL, UDDI
  XML Processing
- **Policy**
  Security
  Management
  Audit
  Configuration
  Charging
- **Adapters**
  Middleware
  Packaged Apps
  Mainframe
  DB
  B2B
- **Data Transformation**
  Format translation
  Content transformation
  XML, CSV, Fixed, Tagged, Excel….
- **Orchestration**
  BPEL Engine
  Process Monitoring & Control

SeCSE
Service Centric System Engineering

Services, Service-Oriented Systems and Requirements

Neil Maiden

Publishing Services

- Service providers
  - Explore markets for new services; product lines
  - Design and deliver (families of) services
  - Supply service descriptions to maximize discovery by consumers
  - Provide related technical and business support
- Consequences
  - Publish (yet-to-be implemented) service descriptions to investigate market – requirements exploration and acquisition
  - Batch requirements into services and service releases
  - Requirements to describe service goals, behaviour and qualities

Discovering services at design-time

- Service integrators and consumers
  - Generate service queries
  - Discover and retrieve the best-fit services
  - Select best-fit services using qualities-of-service
- Consequences
  - Service queries incorporate requirements
  - Service discovery algorithms robust enough to overcome requirements problems – incompleteness and ambiguity
  - Common non-functional requirement and quality-of-service ontologies
  - Reasoning about quality-of-service to select best-fit services

Composing services

- Service integrators
  - Compose services to deliver functional requirements
  - Compose services to meet quality-of-service constraints
  - Decompose requirements to generate required behaviour and qualities of atomic services in composition

- Consequences
  - Represent requirements using process-oriented models
  - Requirements-based application design
  - Develop arguments for requirements satisfaction by a composition

Service-level agreements (SLAs)

- Service consumers and providers
  - Negotiate what service should do for consumer
  - Agree SLAs to document results of negotiation
  - Seek to update SLAs as circumstances change

- Consequences
  - Negotiations are requirements-based, e.g. Win-Win approach
  - SLAs are developed from requirements used to discover services documented in SLAs
  - Requirements-based change processes

Monitoring services

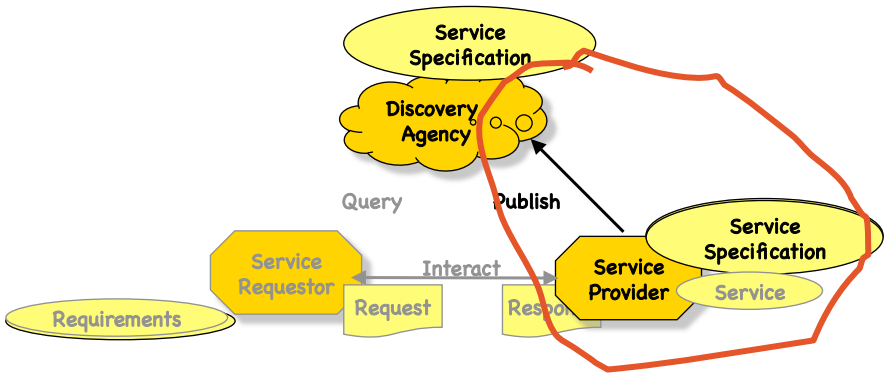
- Service-centric systems
  - Construct service monitors
  - Loose monitoring of services: runs in parallel with main execution
  - Strict monitoring of services: intertwined with main execution
- Consequences
  - Requirements – functional and quality – inform monitor construction
  - Measurable requirements to enable service monitoring
  - Monitors test services for (requirement) compliance

SeCSE
Service Centric System Engineering

Publishing Services

Peter Sawyer

Publishing focus



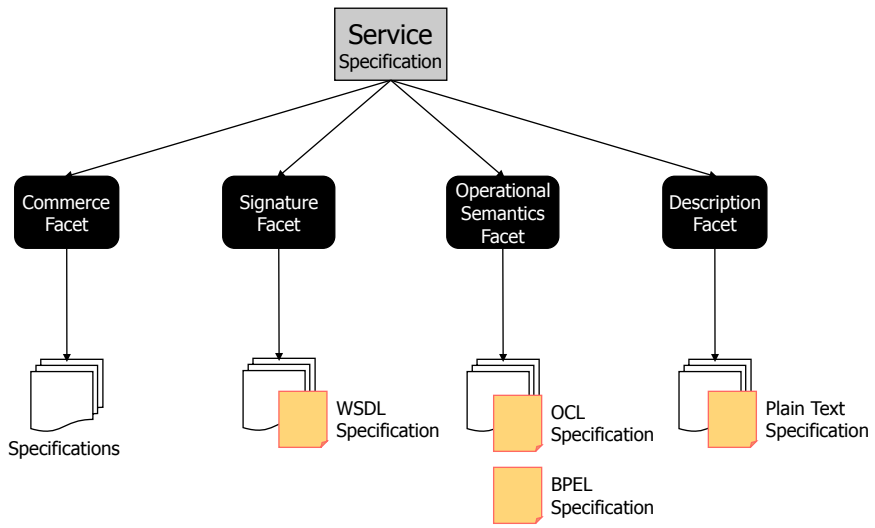
Existing service specification formats

- For any service property, there is a choice of notation with which to describe it, e.g.:
  - Service Signature
    - WSDL, UML, WSDL-S, …
  - Operation Semantics
    - OWL-S, UML/OCL, WSMO, WSDL-S, …
  - Behavioural Specification
    - UML, OWL-S, OpenModel, BPEL4WS, WSCI, …
  - QoS
    - UDDIe, SWSQL, WSQL, E QoS, WSLA, WSML, …

SeCSE's specification solution

- Specification based on service *Facets*
  - Primary aim is to organise specifications that address different properties of a service
  - Projections over one or more service properties
  - Provide flexibility for service providers
  - Maintain compatibility with existing standards
  - Present information to support the other techniques developed within SeCSE
  - Support 3rd party specification mechanisms so compatibility with other approaches is maintained.

Example



Facet support for discovery

- Meta-data used to:
  - describe the set of facets that exist for any service
  - The languages used to specify each facet property
- New facets can be added as they become available
- New facet types can be defined and facets can accommodate any new notation that is encodable using XML

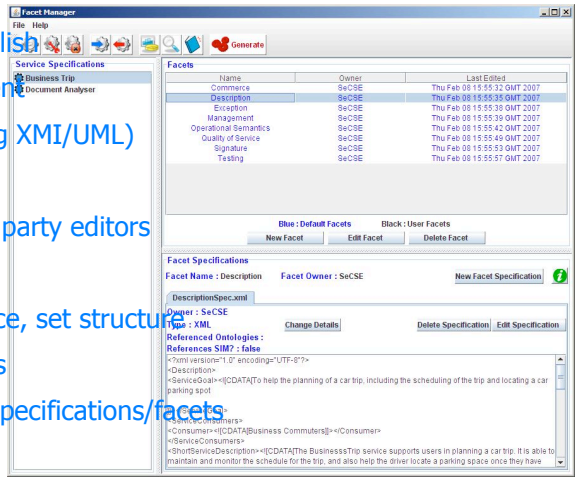
SeCSE facets

- A number of facet types exist, e.g.:
  - Signature – operation signatures and bindings (a subset of WSDL)
  - Operational Semantics – dependencies between operation invocations
  - Exception – descriptions of service failure behaviour
  - Commerce – commercial aspects about service usage, including SLAs
  - Description – structured natural language primarily intended to support requirements-based service discovery
  - QoS – assertions of non-functional properties

Our focus here

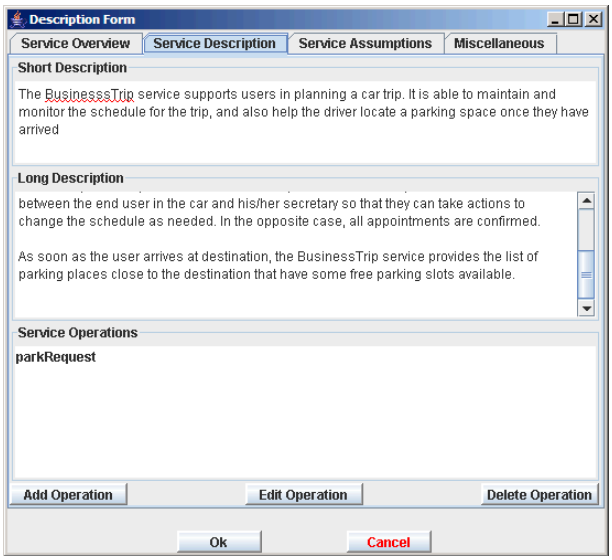
SeCSE Specification Support Tool

- Facet Management
  - Create, edit and publish
- Specification Management
  - XML-based (including XMI/UML)
- Editor Management
  - Integration with 3rd party editors
- Facet Forms
  - Specification guidance, set structure
- Consistency Mechanisms
  - Consistency across specifications/facets



The Description Facet

- To support service discovery, the description facet uses a simple form permitting the capture of
  - Service goal
  - Target service consumers
  - Service description
  - Description of operations
  - Service rationale
  - Business assumptions
  - Technical assumptions
  - Pre and post-conditions
  - Miscellaneous



The QoS Facet

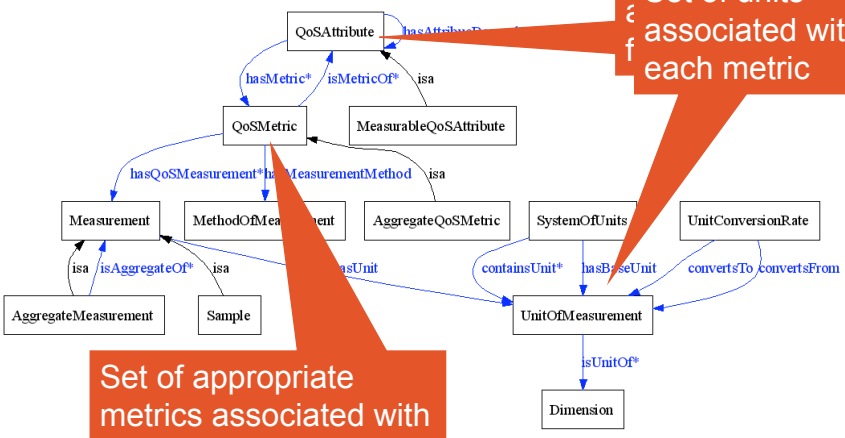
- NFRs may be important to a service consumer.
  - E.g. Service *availability* may be important for on-demand applications
- The QoS facet is designed to allow providers to represent their services' non-functional properties in a way that supports service discovery.

Service QoS discovery challenges

- Must be machine-readable
  - How to represent non-functional properties in a machine-readable form?
- Typically > 1 metric for any given NFR
  - How to insulate the service consumer from having to express their requirement query using the same metrics and units that the service provider uses in their published service specification?

SeCSE QoS solution

- Exploit facets' ability to reference an ontology
- Defined an NFR OWL ontology called QoSOnt

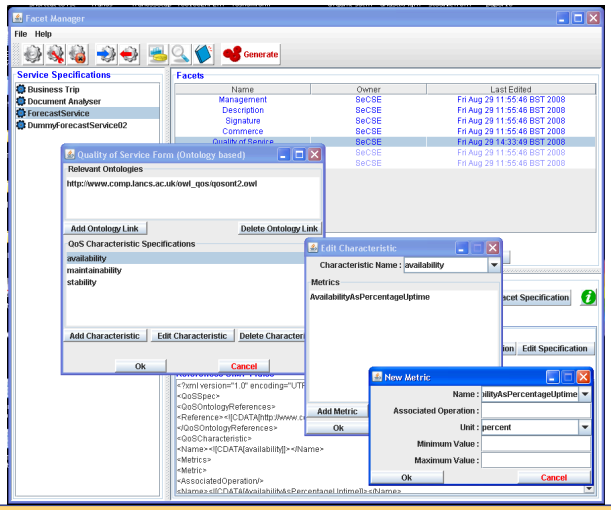


Set of units associated with each metric

Set of appropriate metrics associated with each QoS attribute

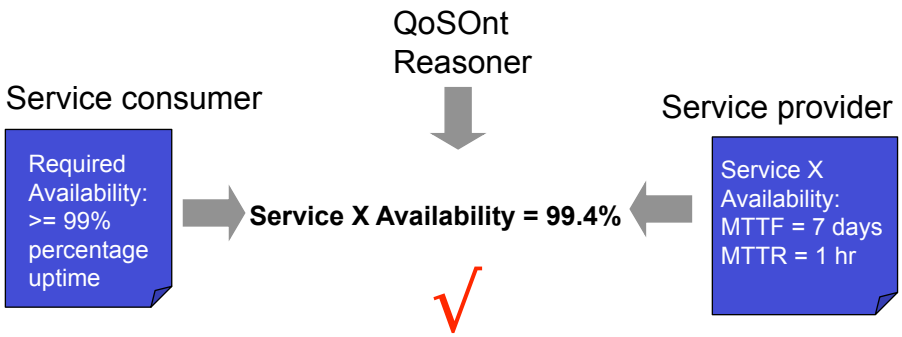
Using QoSOnt (1)

- The SST uses QoSOnt to guide the service provider when specifying the NF properties of their service



Using QoSOnt (2)

- More significantly, QoSOnt has an associated reasoner web service that will resolve metric and unit mismatches transparently



Service-level agreement

- QoSOnt also has a role in SLA negotiation
- Derived an SLA template based on WS-Agreement
- Uses QoSOnt to concretise the WS-Agreement guarantee term

```
<wsag:GuaranteeTerm Name="AvailabilityGuarantee" Obligated="ServiceProvider">
        <wsag:ServiceScope ServiceName="xTripService"/>
        <wsag:ServiceLevelObjective>
            <wsag:KPITarget>
                <wsag:KPIName
OntMetricConcept="http://www.comp.lancs.ac.uk/owl_qos/
qosont2.owl#AvailabilityAsPercentageUptime">Availability</wsag:KPIName>
                    <wsag:Target OntUnitConcept="http://www.comp.lancs.ac.uk/
owl_qos/qosont2.owl#percent">98.9</wsag:Target>
                </wsag:KPITarget>
            </wsag:ServiceLevelObjective>
            <wsag:BusinessValueList/>
        </wsag:GuaranteeTerm>
```

SeCSE
Service Centric System Engineering

Discovering Services

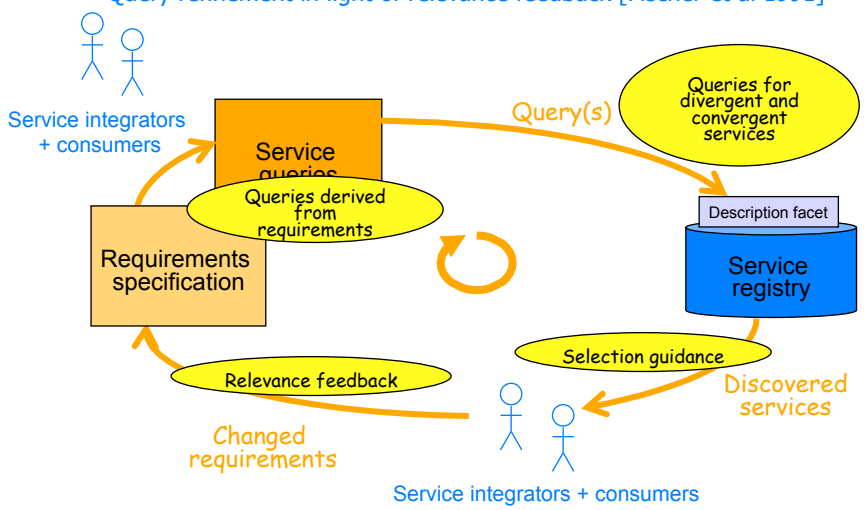
Neil Maiden

The problem

- Two assumptions
  - Service specifications - large numbers of services with multiple versions, expressed in English and WSDL
  - Requirement queries - structured natural language, measurable quality criteria, no ontologies
- Two challenges to discover services when
  1. Semantic, ontological and granularity mismatches exist
  2. Queries are incomplete and ambiguous

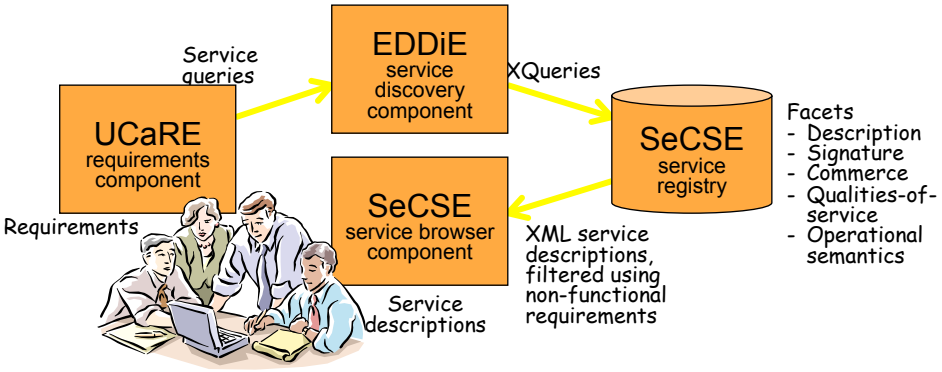
Requirements-based discovery process

- Iterative and incremental service discovery process
  - Query refinement in light of relevance feedback [Fischer et al 1991]

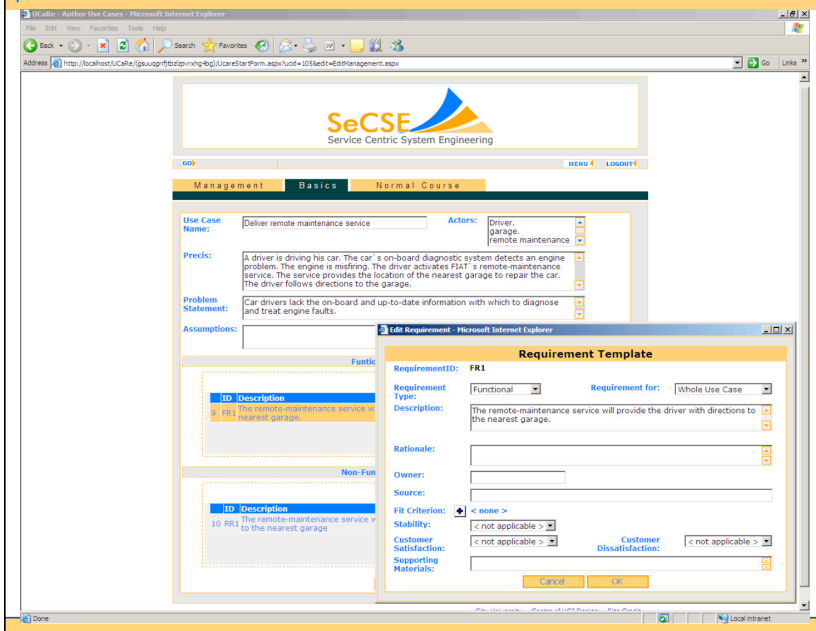


Four Key Modules

- Web-based application
  - .NET implementation using a three-layer model



- Seamless integration of requirements processes and service discovery



Use case-based specification of requirements

Edit Measurable Fit Criterion - Microsoft Internet Explorer provided by Wanadoo

Edit Measurable Fit Criterion

ID	QoSCharacteristic	Dimension	MFCDescription	
C1	Time To Complete	MeanTimeToComplete	The system shall respond to an event in less than 5 Seconds	[Del]

Defining Criterion

ReqType: PR

ReqDescription: The remote-maintenance service shall diagnose all faults within 5 minutes of their detection.

Criterion:

Select a Characteristic: Transactional Throughput, Delay, Time To Complete

Specify Selected Characteristic:

Select a Dimension: MeanTimeToComplete

Description: The system shall respond to an event in less than [maxvalue] [Unit of time]

Unit: Second

MaxValue: 5 MinValue Period Of Time

Test:

Context:

Edit Fit Criteria... Add Fit Criteria...

Cancel OK

Ontology-based quantification of requirements

Management | Basics | Normal Course

Triggering Event: The car engine misfires. [?]

Preconditions: The remote-maintenance service is available. [?]

Normal Course

Add Actions: <--select an option--> 1

ID	Description		
1	The on-board diagnosis system detects the engine problem.	Edit	[Del]
2	The on-board diagnosis system diagnoses the category of engine problem	Edit	[Del]
3	The on-board diagnosis system informs the driver of the problem	Edit	[Del]
4	The driver activates the remote maintenance service.	Edit	[Del]
5	The advanced diagnostic service sends automotive fault data to the diagnostic services of the parts suppliers.	Edit	[Del]
6	Each diagnostic service of a parts supplier provides diagnoses of using the fault data.	Edit	[Del]
7	The advanced diagnostic service identifies the relevant parts suppliers who are responsible for the problem	Edit	[Del]

...change ordering of actions

Currently no requirements for this action. Please select the '+' symbol to add requirements!

add new Requirement for Action...

Manipulation of use case specifications

Use Case Attributes | Normal Course

Use Case Name:	Deliver remote maintenance service
Actors:	driver. garage. on-board diagnostic system. car. automobile. vehicle. passenger.
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.
Problem Statement:	
Assumptions:	
PreConditions:	The remote-maintenance service is available.
Successful End State:	The driver arrives the garrage.
Unsuccessful End State:	The service doesn`t locate a garage to repair the car.
Triggering Event:	The car engine misfires.

Funtional Requirement(s):

ID	Description	Source
FR8	The remote-maintenance service will provide the driver with directions to the nearest garage.	☑
FR9	The remote-maintenance service shall detect faults with the car`s engine.	☐
FR10	The remote-maintenance service shall diagnose faults with the car`s engine.	☐

Non-Funtional Requirement(s):

ID	Description	Source
RR1	The remote-maintenance service will provide the driver with reliable directions to the nearest garage	☑
RR2	The remote-maintenance service shall correctly diagnose 80% of faults with the car`s engine.	☐

Seamless formulation of service requests

Request

Local Settings:

Registry: ◉ SeCSE Service Registry (Rome) ○ Pattern Registry

Part of Speech: Noun, Verb, Adverb, Adjective

Expansion Type: ☑ Similar Terms (Synonym), Generic Terms (Hypernym), Terms in definition

Selected Use Case Attributes:

Date: 13/03/2006 Author: kos

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.

Other:

Selected Requirement(s):

ID	Description	Source
FR8	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	

Setting Service Requests

QUERY LOGOUT

Discovered Services

Query ID: 434 There are 9 services

View All NF-Requirements

ID	ServiceName	Description	MatchValue			
94	XNavigation	This service helps the driver to find locations or point of interests during a trip. The information are available both in textual and in graphic form depending on the user preference and on the display capabilities.	2.2451	☑	[Matching]	[NFReq]
98	YAgenda	This service provides simple electronic personal agenda functions. Add, check and delete appointments.checkagendaappointment	2.0193	☑	[Matching]	[NFReq]
93	XAgenda	To expose in a secure and safety way the company agenda so that it can be automatically updated on the basis of traffic events like for example the arrival time. The XAGENDA services exposes a set of operations for the secure reading/writing access to the company/personal user agenda.	1.845	☑	[Matching]	[NFReq]
88	xrealtime	Get a standard IntraDay price chart for a security. For detailed information on this operation, argument information, and sample values, go to http://www.xignite.com/xRealTime.asmx Returns a range of ticks for a security.	1.6599	☑	[Matching]	[NFReq]
99	AAgenda	The service provides secure access to the user corporate agenda to manage appointments.	1.5367	☑	[Matching]	[NFReq]

1 2

OK

City University Centre of HCI Design Site Credit

Retrieved services

EditBasics - Microsoft Internet Explorer provided by Wanadoo

Satisfaction of Non-Functional Requirements for the Service

Service ID:94 Service Name: XNavigation

NFReqID	Characteristic	Metric	ReqValue	QoSValue	Satisfaction
AR5	Availability	AvailabilityAsPercentageUptime	90 Percent	78 Percent	Unsatisfied
PR6	Time To Complete	MeanTimeToComplete	5 Second	5 Second	Satisfied
PR6	Transactional Throughput	TransactionalThroughputBenchmark	1800 perSecond	1600 perSecond	Unsatisfied
PR6	Delay	MeanRoundTripTime	5 Minutes	3 Minutes	Satisfied

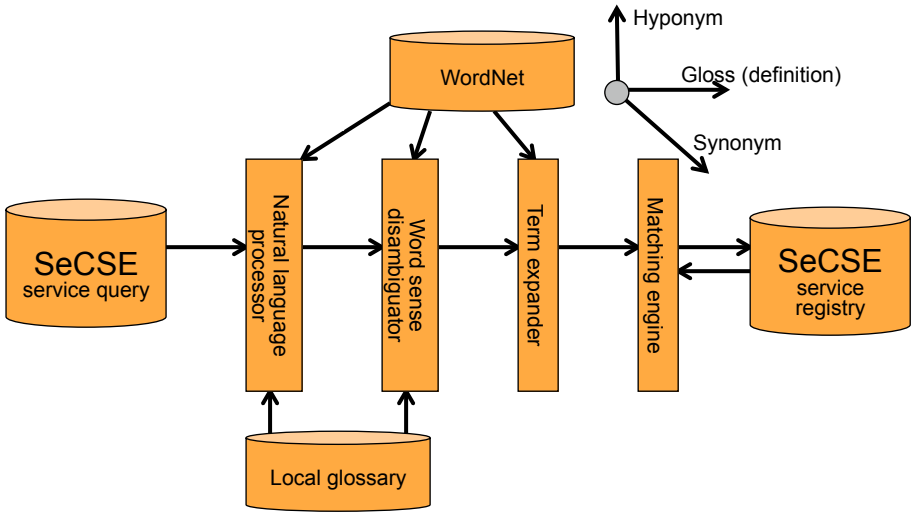
1

OK

Filter services using non-functional requirements compliance

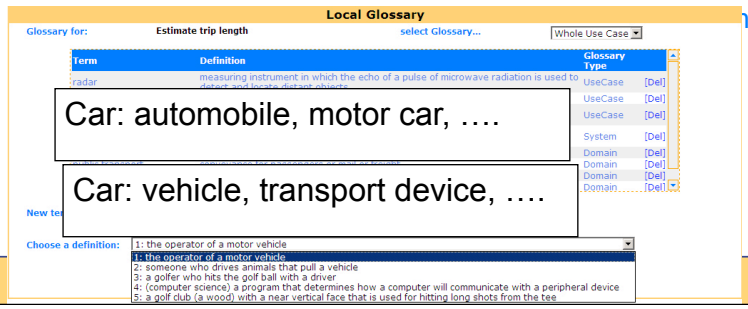
EDDiE's Service Discovery Algorithm

- 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. Selecting sense of term with only 1 possible sense
  3. Selecting senses of synonyms of terms that have already been disambiguated
  4.
  5.
  6.
  7.



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, ….

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

Does It Work?

- Different evaluations have taken place
  - Experiments of EDDiE algorithm to retrieve services
  - Remote uses of UCaRE, EDDiE and Service Browser tools in requirements processes at CA (Computer Associates), KD Software
  - Facilitated requirements workshops in automotive (Fiat, CA, DaimlerChrysler) and UK policing (NPIA)
- Outcomes
  - EDDiE precision and recall adequate but influenced by selected requirements attributes
  - Retrieved services trigger discovery of more novel requirements not discovered using other techniques, when facilitated
  - Un-facilitated generation of requirements more difficult

SeCSE
Service Centric System Engineering

Monitoring Services

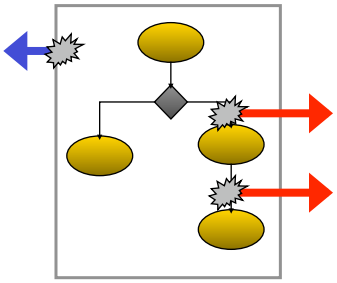
Neil Maiden (on behalf of Luciano Baresi)

Current approaches

- Standard technology
  - BPEL is the de-facto standard for web services composition
  - Many interesting engines are available (for free)
  - Services can be described in many different ways, e.g. WSDL
- Issues
  - No defensive programming
  - No intertwining of business and supervision logics
  - Many possible supervision policies for the same business process
- Possible deviations
  - Inconsistency with respect to recorded behaviour
  - Inconsistency with respect to expected behaviour
  - Unjustified behaviour

Loose and strict service monitoring

- Loose monitoring
  - Runs in parallel with main execution
- Strict monitoring
  - Intertwined with main execution

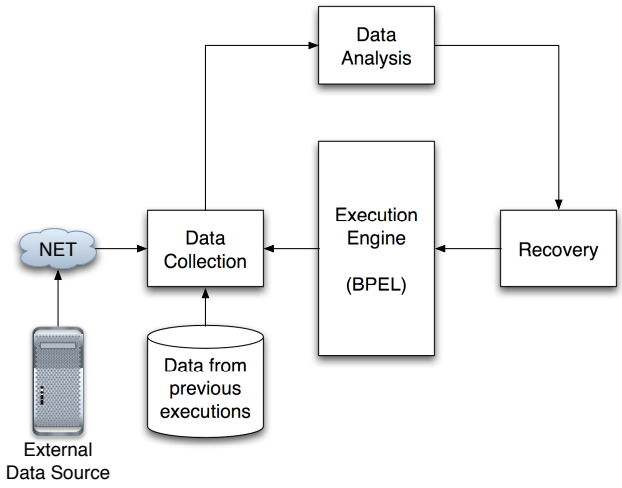


Loose monitoring

- Based on event calculus
  - A first-order formal language for specifying properties of dynamic systems which change over time using predefined predicates, including:

- Happens(e, t, $\Re$(t1,t2)) - occurrence of an event e of instantaneous duration at some time t within the time range $\Re$(t1,t2)
- HoldsAt(f,t) - fluent f holds at time t.
- Initiates(e,f,t) - fluent f starts to hold after the event e at time t.
- Terminates(e,f,t) - fluent f ceases to hold after the event e occurs at time t

Strict monitoring



WSCoL (New Language)

- Mixes JML (lightweight version) and XML technology
- Two main activities
  - Data Collection
    - internal, external, and historical variables
    - Variable aliasing
  - Data Analysis: relationships between data
    - Typical boolean operators (and, or, not, implies, if and only if)
    - Relational operators (<,>,==, <=, >=)
    - Typical mathematical operators (+, -, *, /, %)
    - Quantifiers - forall, exists
    - Data computation - max, min, avg, sum, product
    - Data type specific functions - length, starts-with, etc.

WSReL: Expressing Rules for Recovery Strategies

- Event
  - Monitoring has signaled an error
- Condition
  - Discriminates between different recovery strategies depending, for example, on the extent of the error
  - Uses WSCoL to define the condition
- Action
  - A recovery strategy
  - Made up of different recovery steps
    - Step_A || Step_B || Step_C
  - Each step is made up of a number of atomic recovery actions
    - Action_A && Action_B
- The rules have instance validity
- They do not have access to the process internals

SeCSE
Service Centric System Engineering

Future Trends

Neil Maiden and Pete Sawyer

Future Requirements-Related Trends

- Requirements-driven service composition
  - Functional and quality requirements informing construction of SLAs on atomic services in a SoA
- End-user requirements in dynamic environments
  - Systems integrators cannot be surrogates for emerging end-user needs
- Systems development governance
  - Aligning project governance with service governance
  - Wider engagement with business, legal and social research
- A wider notion of what is a service
  - Human services (Amazon Turk) and mixed services as well as software services

SeCSE Sources

- Information about SeCSE is available from
  - www.secse-project.eu
- SeCSE tools can be downloaded from
  - http://sourceforge.net/projects/secse
- An explanatory video about the SST tool is available at
  - http://www.comp.lancs.ac.uk/~walkerdi/FacetedSpec.mp4
- Access an interactive version of UCaRE at
  - http://achernar.soi.city.ac.uk/ESD/UCaRe/
- SeCSE's development environment is online at
  - http://newton.eng.it/SeCSE
- The QoSont quality-of-service ontology is available at
  - http://www.comp.lancs.ac.uk/owl_qos/qosont2.owl

SeCSE
Service Centric System Engineering

Questions