Service modeling

IC SOC Panel

December 16, 2003

Panelists

- Fabio Casati, HP Labs, USA
- Jean-Jacques Dubray, Attachmate, USA
- Mike Papazoglou, Univ. of Tilburg, NL
- Barbara Pernici, Politecnico di Milano, I

Why service modeling?

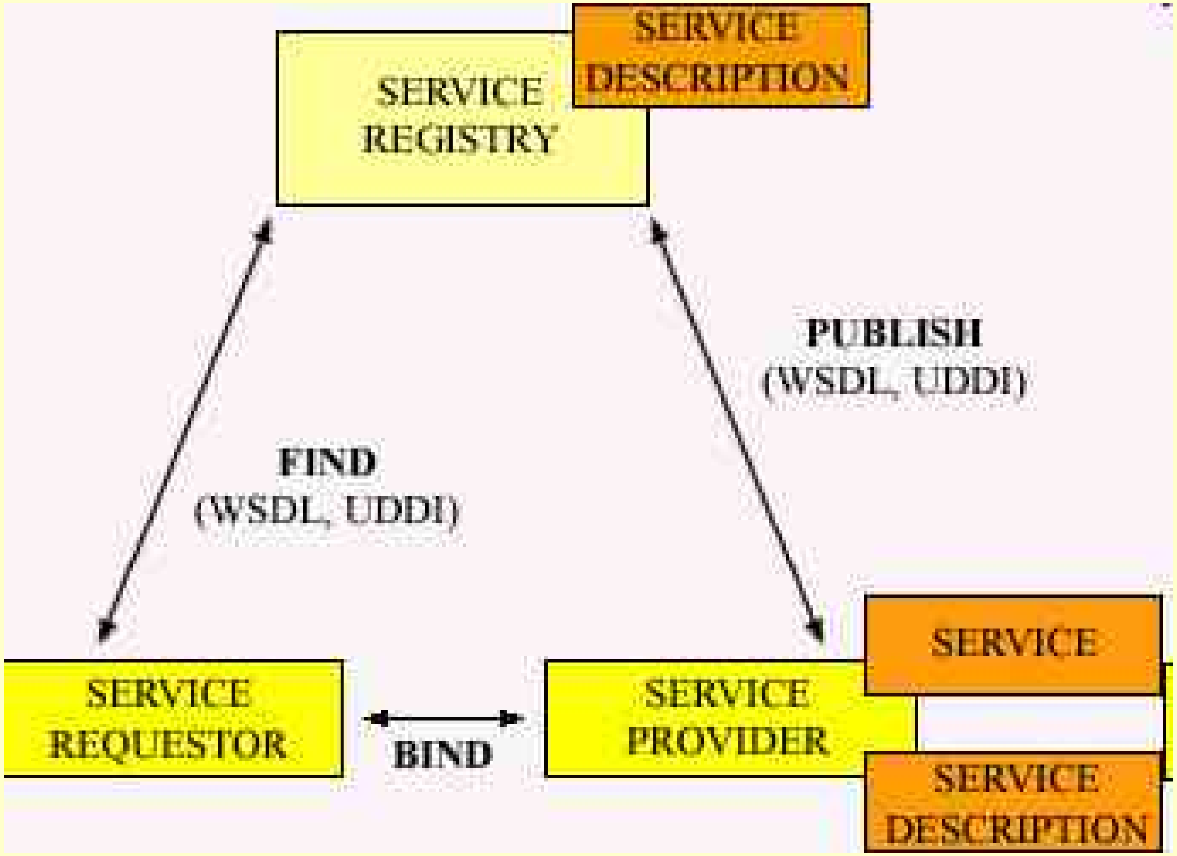
- To express what services can do
- To represent services for customers (requirements, high level specifications)
- To perform analysis (resources)
- For coming to an agreement (contracts)

Service modeling

- Basics
  - “stack”

Message model	SOAP	*RPC-like, XML*
Service model	WSDL	*Service interface, XML, stateless*
Registry	UDDI	*Yellow pages*

Service Oriented Architecture



Service models

- “advanced”
  - Composition
  - Additional features (transactions, security, policy,…)
  - Evolution (stateful)
  - Resources
  - Context (multichannel, mobile)

Modeling goals

- Process modeling
- Dynamic composition
  - Search
  - Adaptation
- Design
  - Methodology
  - Evolution
- Management
  - Unstable resources

Questions to panelists

- Old wine in new bottles?
  - E.g. cfr Van Der Aalst, Dumas, Ter Hofstede, Euromicro 2003
- Why so many XML-based proposals?
- Is flow representation (not) enough?

Perspectives

- Conceptual modeling and requirements (Papazoglou)
- Business process modeling (Dubray)
- Coordination and composition models (Casati)
- Retrieval (Pernici)

Modeling of services and Retrieval

B. Pernici

Themes

- Repositories
- Search
  - When to search? (design time vs run time)
  - Level of granularity? (fine or coarse graines services)
- How to invoke?

Repositories

- Two main streams
  - UDDI
    - providers
    - service categories
    - reference abstract interfaces (t-models)
  - Semantic web services (DAML-S)
    - Service profile
    - Ontology-based

Search

- UDDI
  - Simple retrieval functionalities
- DAML-S
  - Inferencing capability

- Based on
  - On functionalities
  - On quality

Advanced repositories

- Structured queries
- Ontologies of services, not only of elements

- Performance issues (in particular at run time)
- Data quality of repositories contents
  - Accuracy
  - Completeness
  - Credibility, reputation
- Local/specialized repositories

Old wine?

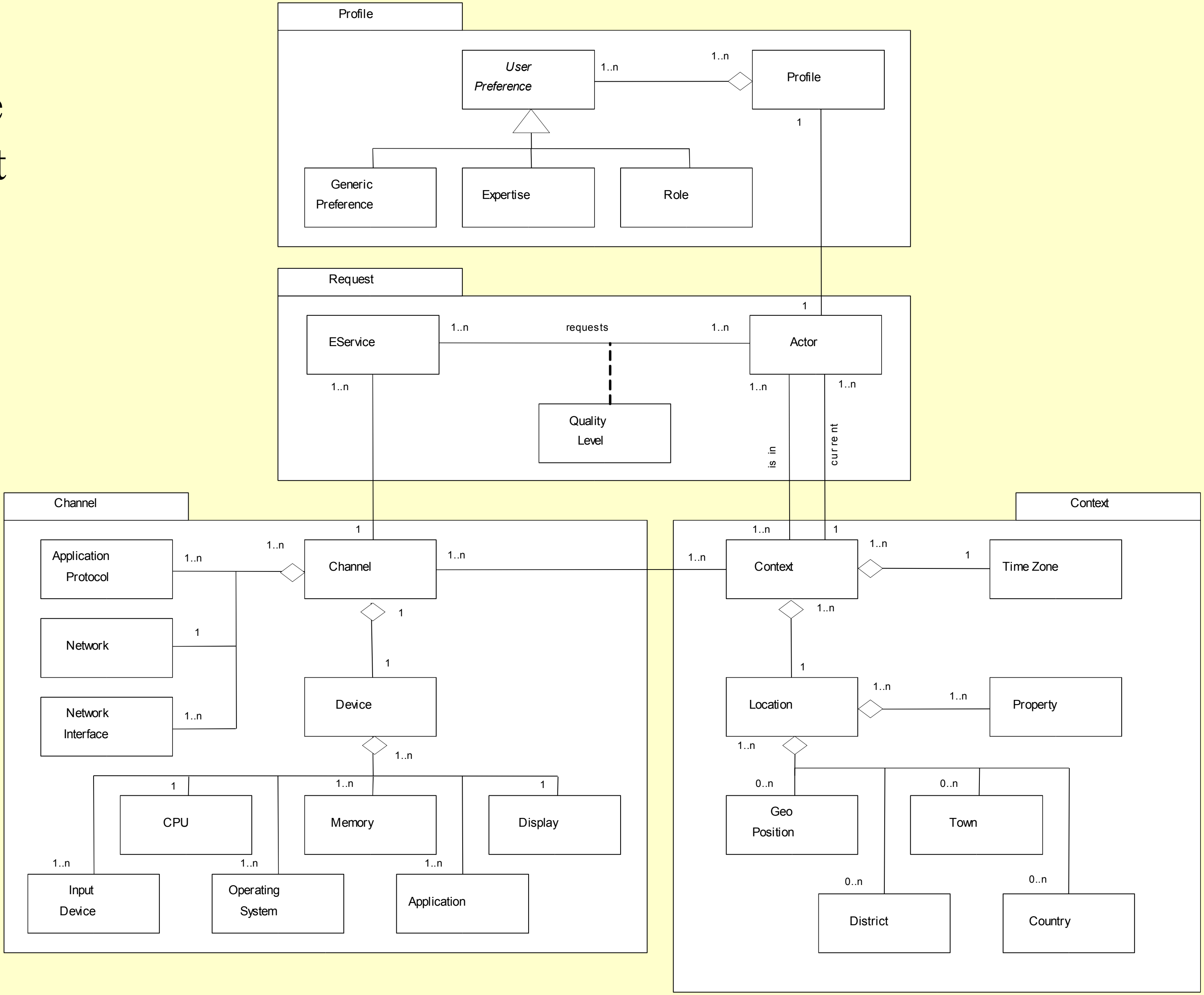
- Component repositories for reuse
  - Versioning
  - specifications, traceability of requirements
- XML-based query languages
- Inferencing mechanisms

- But
  - Beyond just interfaces
  - Context representation
  - Fit for use

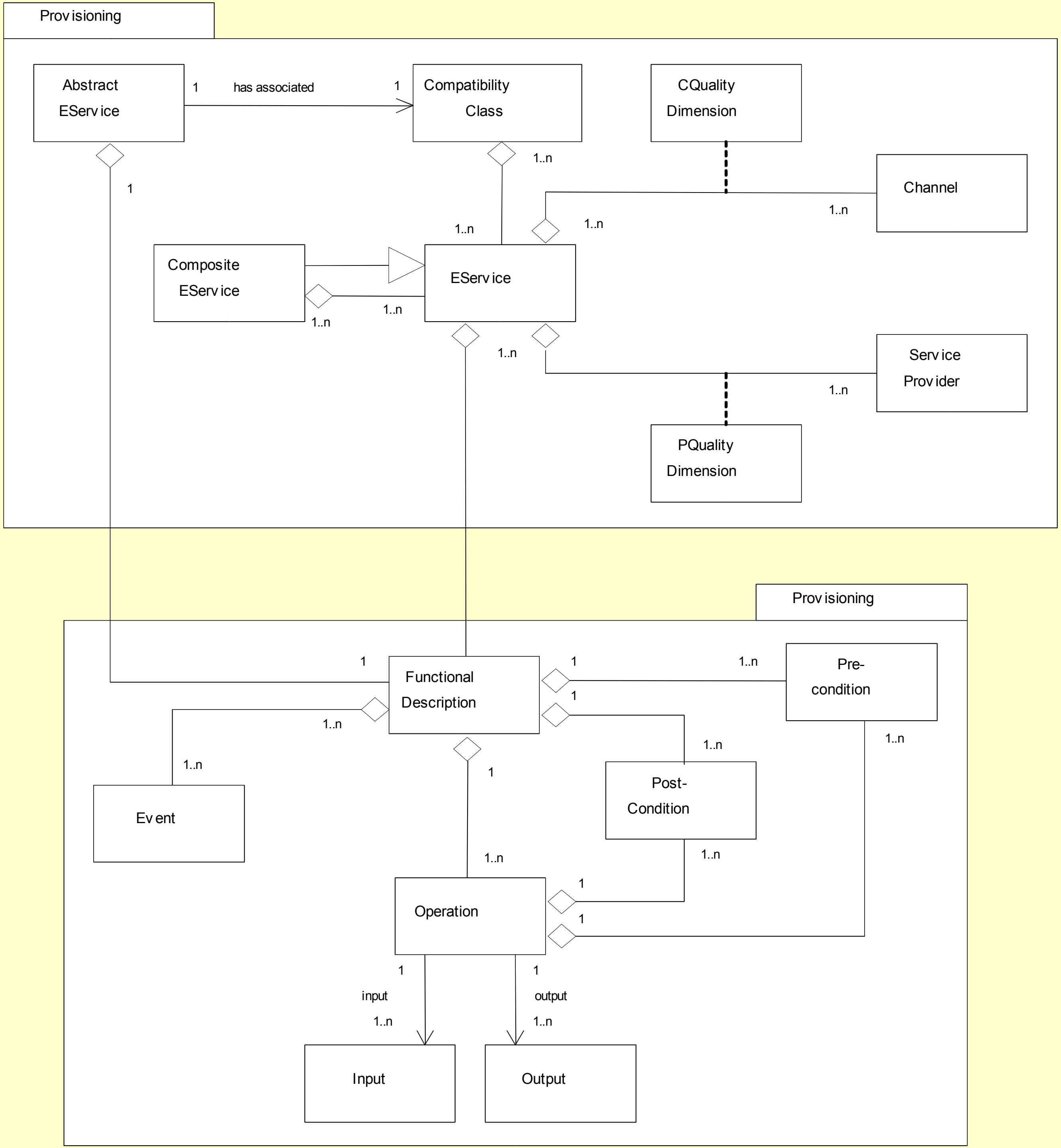
Search results

- “best” fit vs fit for use
- Functional fitness
  - Similarity of interfaces (e.g. VISPO, CAiSE’03)
  - Similarity of behaviour (e.g. Casati et al, CAiSE’03)
- Quality fitness (e.g. MAIS)
  - Request context
  - Channel for provisioning (web vs e-mail, desktop vs PDA, network connection)
  - provider

MAIS service request



MAIS service description



Invocation

- Adaptation
  - Functional (wrappers, mediators)
  - Quality levels
- Negotiation, contracts (based on requestor and service profiles, quality levels)
- Variable channels
  - Multichannel (time-context dependent), e.g. priority msg delivery in emergencies, variable channel depending on the personal schedule
  - Mobile: location awareness, variable resources (e.g. network connection)

Other panel questions

- XML?
  - Mainly for interface description (used at design time for selection, at run time for invocation)
- Flow representation?
  - Not easy to query
  - Only external view
  - For behavioural modeling