ALIVE

"Coordination, Organisation and Model Driven Approaches for Dynamic, Flexible, Robust Software and Services Engineering"

Javier Vazquez
Universitat Politècnica de Catalunya
jvazquez@lsi.upc.edu

Project Parameters

- IST Framework 7 STREP Project
  - Funded from the 1st Call ICT
- Focus:
  - Design, Deployment and Management of Service-Oriented Applications
  - Organisational and coordination techniques
  - Model-Driven approach
- Project ID: FP7-215890
- Area: Service and Software Architectures, Infrastructures and Engineering
- Costs:
  - Total Cost: 3.772.256,84 Euro
  - Req. Cont: 2.809.213,72 Euro
- Dates:
  - Start: 1st February '08
  - Defined End: 31st Aug '10

ALIVE Project Consortium

- Universitat Politecnica de Catalunya (UPC)
  - Barcelona / Spain
- University of Bath (BATH)
  - Bath / UK
- Trinity College Dublin (TCD)
  - Dublin / Ireland
- University of Aberdeen (UNIABDN)
  - Aberdeen / UK
- Universiteit Utrecht (UU)
  - Utrecht / The Netherlands
- Calico Jack Ltd. (CJ)
  - Dundee / UK
- TMT Factory (TMT)
  - Barcelona / Spain
- Thales Nederland B.V. (THALES)
  - Hengelo / The Netherlands

Partner Roles Summary

- Universitat Politècnica de Catalunya (UPC)
  - Normative aspects of distributed systems / Coordination and Organisation [WP1, WP2 and WP8 lead]
- University of Bath (BATH)
  - Semantic web service description, discovery, brokerage and workflow enactment [WP5 lead]
- Trinity College Dublin (TCD)
  - Methodologies for development of large-scale dynamic systems [WP6 lead]
- University of Aberdeen (UNIABDN)
  - Formal specification and verification of organisations / Academic Industry partnerships [WP4 and WP7 lead]
- Universiteit Utrecht (UU)
  - Formal semantics of organisational models [WP3 lead]
- Calico Jack Ltd. (CJ)
  - Use case: multimodal communication within the entertainment domain
- TMT Factory (TMT)
  - Use case: context aware information services for citizens
- Thales Nederland B.V. (THALES)
  - Use case: dynamic re-structuring of distributed emergency management services

Context

- New generations of networked service applications should be able to:
  - *communicate and reconfigure* at runtime
  - *adapt* to their environment
  - *dynamically combine* sets of building block services into new applications
- This requires profound changes in the way software systems are designed, deployed and managed…
  - from existing, top-down, "design in isolation" to new approaches based on integrating new functionalities/behaviours into existing running systems

Objectives

- In this context, the objectives of the Alive project are:
  - To *bring together the leading edge methods* from Coordination Technology and Organizational theory *with new technologies* on Model Driven design *to create a framework for software and services engineering* addressing the new reality of "live", open systems of active services.
  - To *close the gap* between theoretical approaches and existing web services technologies

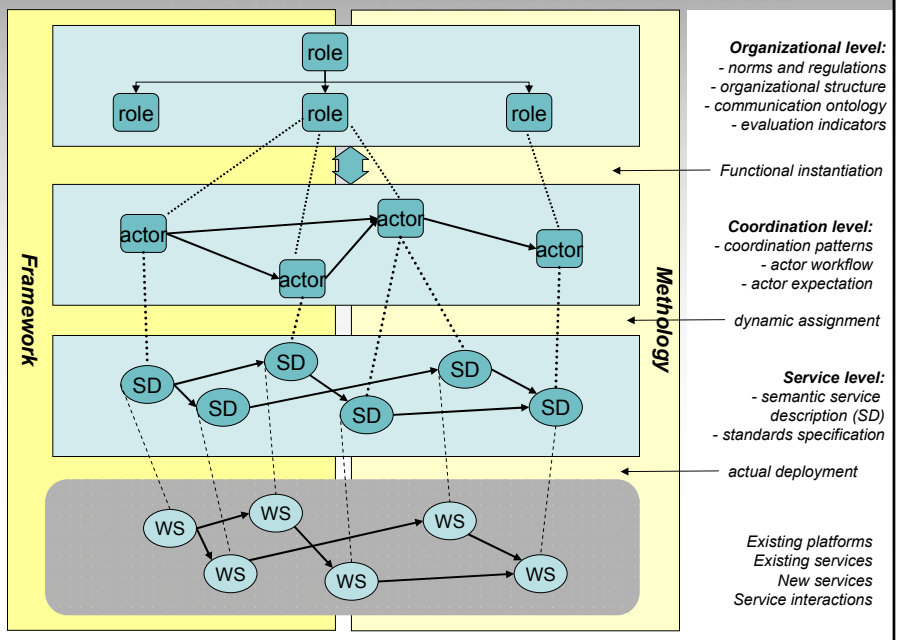
The ALIVE approach

- Splitting the design process in three separate layers
  - **Service layer**: augments service models to make components aware of their social context
  - **Coordination layer**: specifying patterns of interaction
  - **Organisational layer**: specifying organisational rules that govern interaction

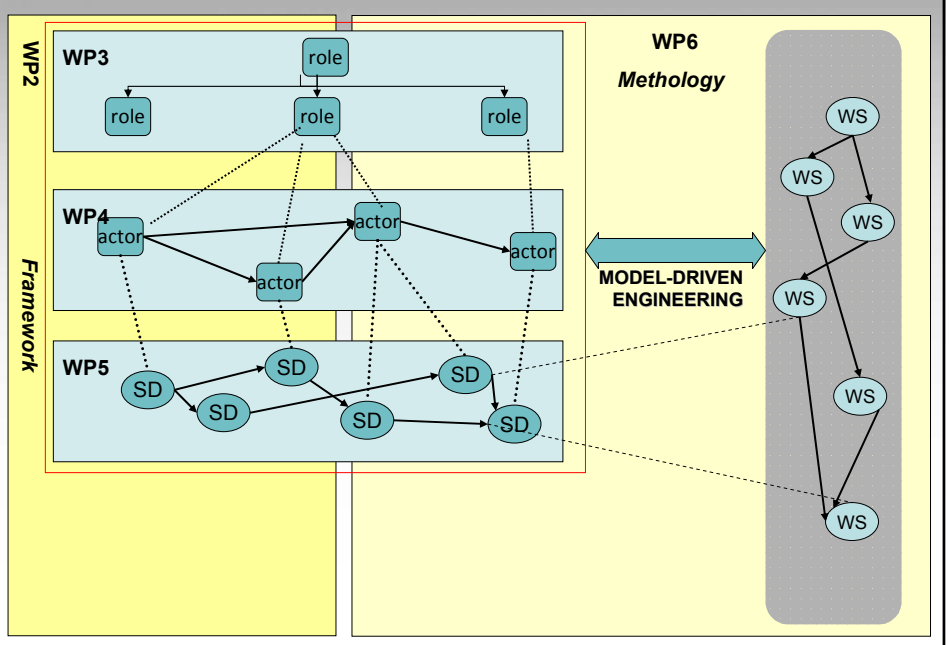
Core motivations for the approach

- New service engineering approaches can benefit from the adaptation coordination and organisation mechanisms often seen in human and other societies.
- Such mechanisms provide
  - *Robust descriptions* of distributed systems
  - Account for the *individual autonomous nature* of service providers/consumers
  - Define a wide range on strategies and mechanisms with *known properties*
- Project Meme:
  - The strategies used today to organise the vastly complex interdependencies found in human, social, economic behaviour will be essential to structuring future software systems

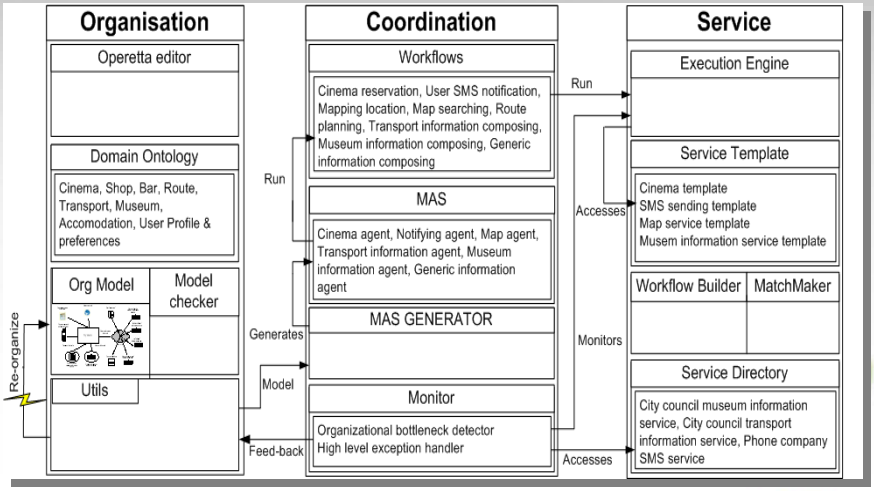
The ALIVE approach



The ALIVE approach



ALIVE Architecture



Summary: Project objectives

O1 – Sound theoretical foundations

- New framework for design, deployment and management of service-oriented systems with sound theoretical foundation for organisational aspects

O2 – New engineering techniques and tools

- Provide concrete modelling languages and their implementations to capture organisational, coordination and autonomy levels

O3 – New design and methodological approaches

- Design methods and tools

O4 - Integration and availability on SOA environments

- Results widely available both during and after the project

Summary: Target Outcomes

R1 – Architecture and Scientific foundations
- Sound theoretical model

O1

R2 – Organisational Level models and tools
- Rules of engagement, organisational actors…

R3 – Coordination Level models and tools
- Coordination patterns, coordination techniques

R4 – Service Level models and tools
- Connection of services to coordination and organisational level concepts

O2

R5 – Design and methodology
- Profiles, patterns and modelling guidelines

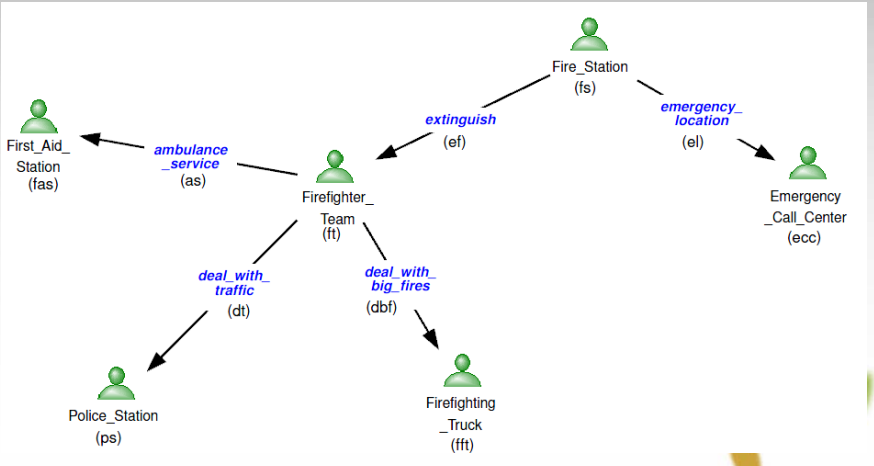
O3

R6 – Integrated tools and systems for easy application
- Tools, libraries and plug-ins

R7 – Use case prototypes

O4

Use Case 1: Emergency Scalation Handling



Use Case 1: Emergency Scalation Handling

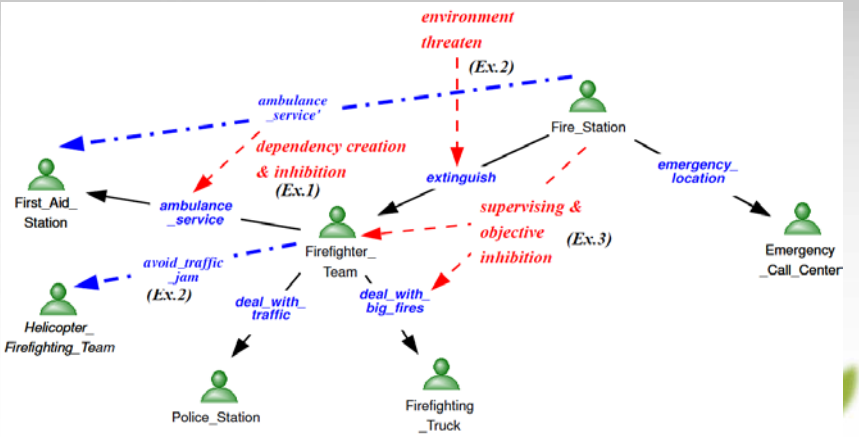
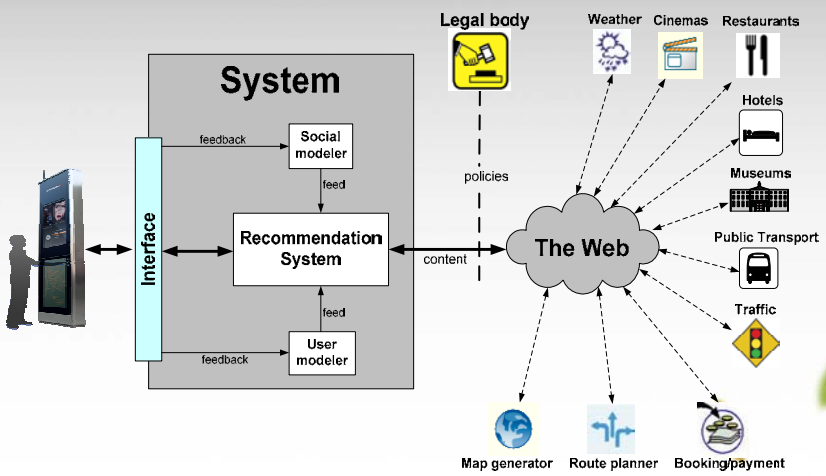
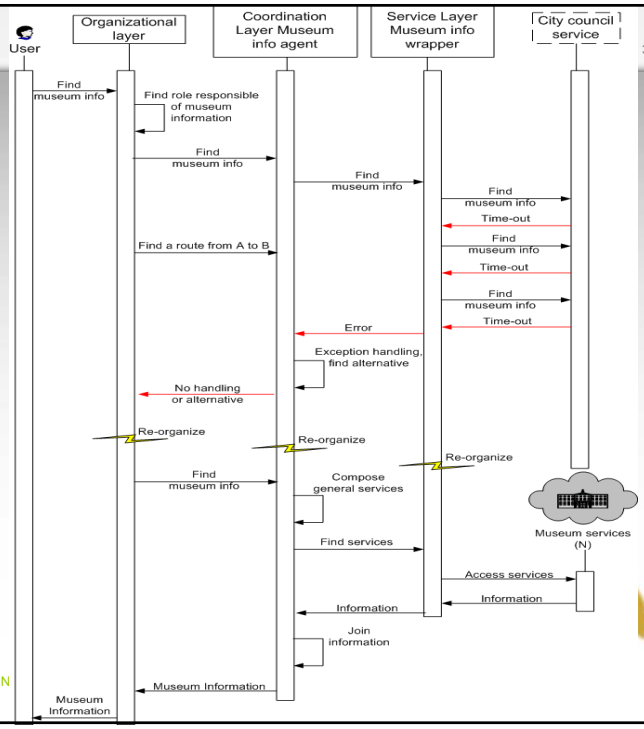


Figure 2.2 Running Example -Changes in Stakeholders' Relationships in Various Situations

Use Case 2: Multimedia distribution of information and orchestration of services

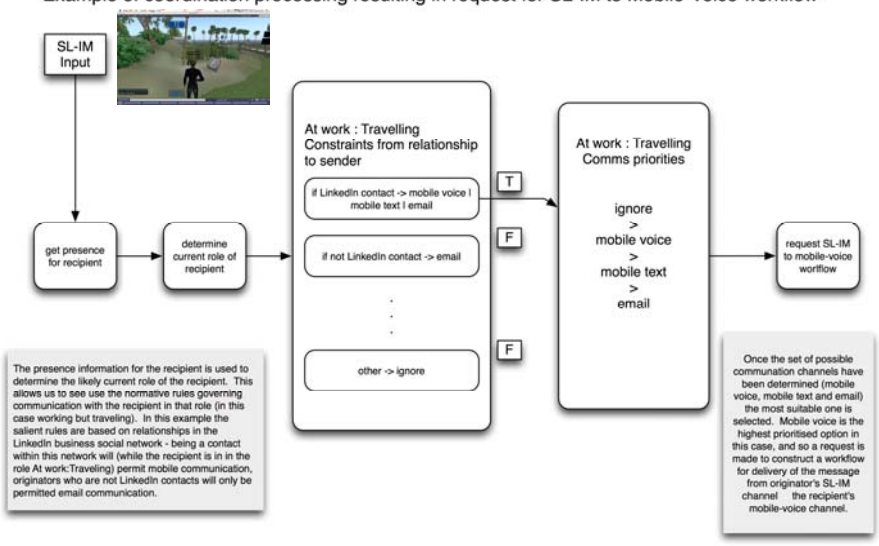




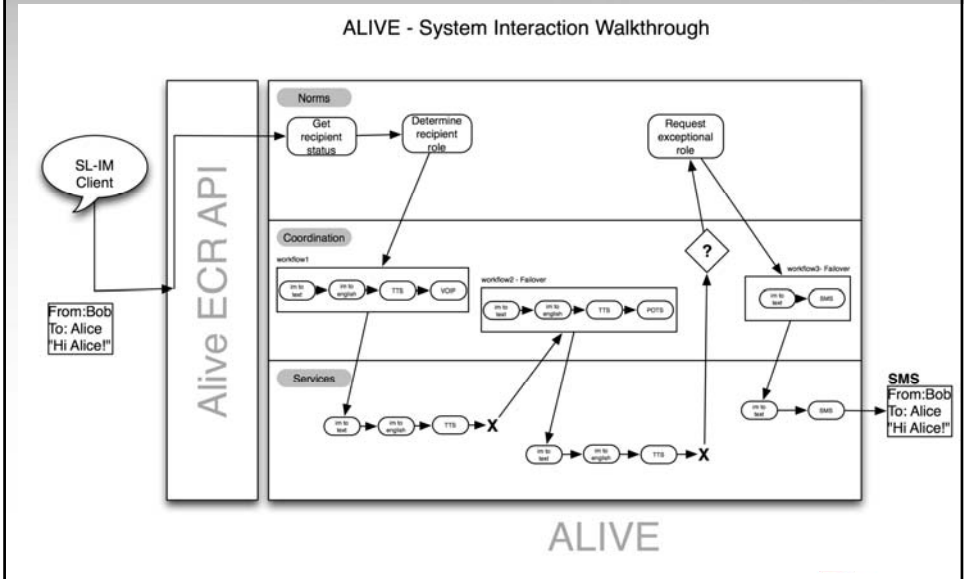
ALIVE EU FUN

Use Case 3: Entertainment Communication Router

Example of coordination processing resulting in request for SL-IM to Mobile-Voice workflow



Use Case 3: Entertainment Communication Router



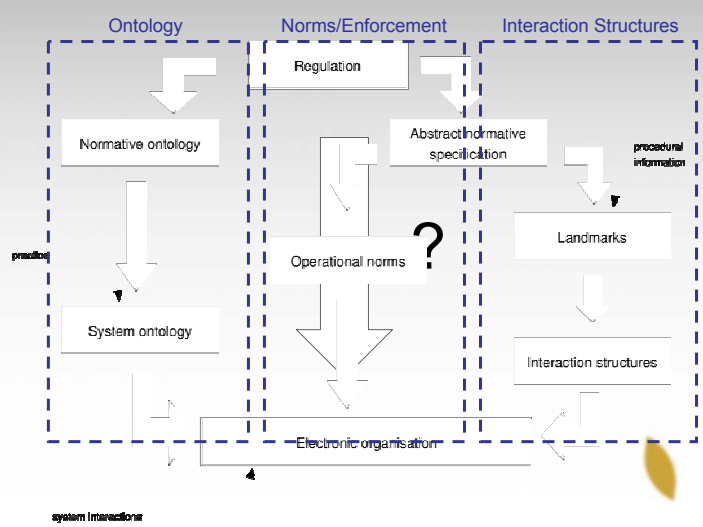
Benefits of the ALIVE method

- Mapping human organisations to service-based solutions
- Development layers allows for:
  - Traceability (why is something done in this way on this level?)
  - Adaptivity (moving up in abstraction to solve problems at a specific level)

Change and adaptation

- 3 levels:
  - Changes in system functionalities — Service
    - e.g., services that become unavailable or are not used correctly
  - Changes in environmental conditions — Coordination
    - e.g., changes (sensed symptoms) that can lead to potential failure during the achievement of objectives
  - Changes in stakeholders needs — Organisation
    - e.g., changes in laws and norms that regiment particular organisational protocols and responsibilities

From abstract regulations to implementation

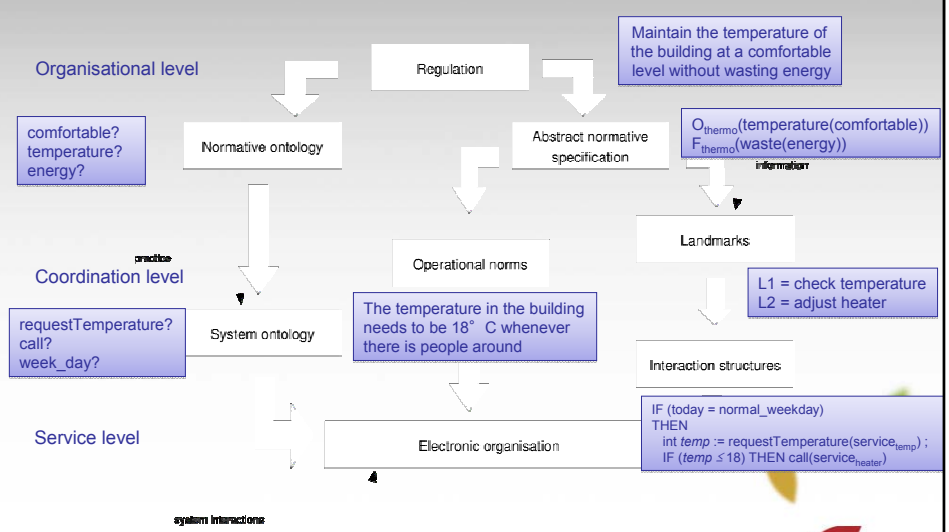


Simple example

(example by Virginia Dignum and Huib Aldewereld)

- Thermostat organisation:
  - Declarative regulation: *maintain the temperature of the building at a comfortable level without wasting energy*
- What does that mean in operational terms?
- Which services to call to adhere to this norm?
  - Service to get the day of week
  - Service to get the time of day
  - Service to get the temperature of each room
  - Service to translate ° F to ° C
  - Service to regulate the heater/air-conditioner

Example, continued



Deploying a regulation

- Operationalization
  - How to check for violations?
  - How to enforce? (what sanctions to use?)
  - How to cope with violations? (repairs needed?)
- Contextualization
  - Bringing the regulation into context
  - E.g., *personal info* in the context of a hospital vs. *personal info* in the context of severe criminality registers

UPC research in ALIVE

- Formal model of normative adaptation
  - From norms to regulations
  - Norm Metamodel
- Formalisation of the connection between (fixed) abstract norms and (dynamic) context-depended norms
- Formalisation of the connection between abstract terms and actions in the domain with lñow-level events and service end-points
- Normative influence at the coordination level
  - How norms and organisational landmarks influence the actor execution
  - How norms and organisational landmarks influence coordination and distributed planning?
- Generalisation of the Agentified services platform from CONTRACT
  - Validation in TMT use scenario

http://www.lsi.upc.es/~jvazquez