WP1: Connecting Systems

Emergent Middleware:
Rethinking Interoperability for Complex Systems

Gordon Blair and Paul Grace

Computing Department,

Lancaster University, UK

Goals of WP1: A reminder

To provide an overall architecture for CONNECT with emphasis on the theory and practice behind our approach to eternal networked systems:

- A point of integration for CONNECT;
- Close relationship with work packages 2 to 5
  - Defines and documents the common architectural principles behind CONNECT;
  - Integrates all contributions coming from the specialized work packages into a consistent system prototype which in turn supports the experimentation in WP6;
- A central resource in dissemination and standardization.

Associated Tasks

- The definition and refinement of a technology-independent and eternal architectural framework based on the concept of *emergent connectors* or CONNECTors **(task 1.1) – LANCS (1-42);**
- A focus on the major role of *semantics* in enabling the CONNECT approach to automated runtime correlation of systems **(task 1.2) – Docomo (1-42);**
- The development of key underlying systems principles and techniques to support the development of a practical, efficient and self-sustaining CONNECT prototype capturing the above-mentioned architectural framework **(task 1.3) – LANCS (7-42).**

Interoperability in Complex Systems

- Emergence of multi-faceted '*Systems of Systems*'
  - Work together to meet the global aims of associated applications and services
- Characteristics
  - Extreme heterogeneity
    - Technology, network, software, hardware
  - Dynamic composition
    - Which systems interact with which is unknown until runtime
- Interoperability
  - *"Ability of two or more systems to exchange information and to use the information"*

A Few Words from Danny Cohen

- In the beginning ARPA created ARPANET.
- And the ARPANET was without form and void.
- And darkness was upon the deep.
- And the spirit of ARPA moved upon the face of the network and ARPA said, 'Let there be a protocol,' and there was a protocol. And ARPA saw that it was good.
- And ARPA said, 'Let there be more protocols,' and it was so. And ARPA saw that it was good.
- And ARPA said, 'Let there be more networks,' and it was so.

Gordon’s distributed systems version

- In the beginning there was small scale experimentation.
- And the experiments were without abstraction or openness.
- And darkness was upon the deep.
- And the spirit of the OMG moved upon the face of distributed systems and said, 'Let there be a middleware standard,' and there was a standard. And OMG saw that it was good.
- And Microsoft said, 'Let there be more standards,' and it was so. And Microsoft saw that it was good.
- And the community said, 'Let there be more networks and of course also mobility, ubiquity and cloud computing for good measure,' and it was so.....

Motivating Scenario

Service Discovery Protocol Heterogeneity

Challenges of Discovery Heterogeneity

Six challenges that must be resolved to achieve SDP interoperability:

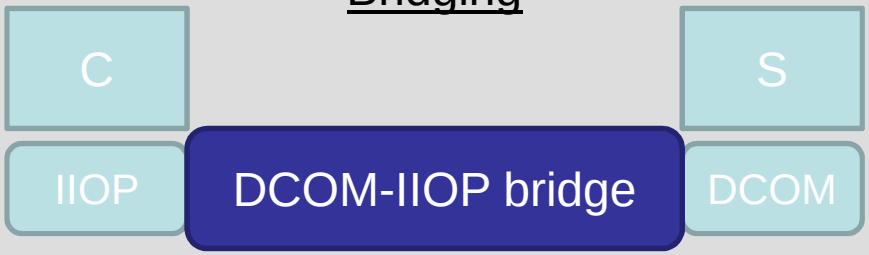
1. Heterogeneous description of the behaviour
   - Different description languages to describe services e.g. WSDL, SLP templates, …
2. Heterogeneous systems architecture
   - e.g. Protocols that use structured networks of directories versus unstructured
3. Heterogeneous protocol behaviour
   - Active/passive protocols e.g. lookup request and respond, versus periodic announcement
4. Heterogeneous message format
   - Each discovery protocol uses a different message format
5. Heterogeneous networks
   - Lookup requests and responses are routed over different networks e.g. IP multicast, P2P (Pastry, Gnutella), MANETs
6. Heterogeneous non-functional properties
   - Security, privacy, location mechanisms in SDPs

Middleware Approaches to Interoperability

The Middleware Approach

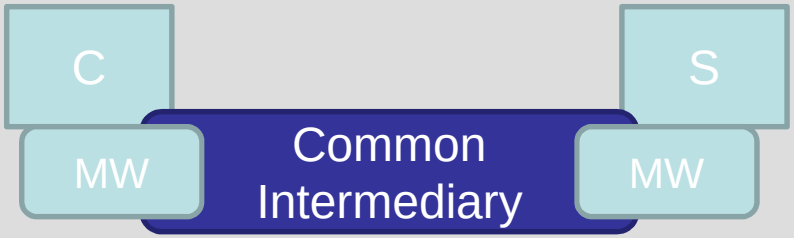
No interoperation with other standards and technologies

Bridging



Static deployment; bridge type must be available; significant effort to develop each bridge

Service Buses



Tied to interaction styles (e.g. messaging);

Analysis

- Focus on Communication protocols
  - Applications, semantic interoperability?
  - Cross abstraction interoperability?
    - RMI to P/S to Tuples to ...
- Known protocols aware of each other
  - New protocol to interoperate with at runtime?

Rethinking Interoperability

- State of the art is mostly design-time interoperability
  - A priori knowledge of connected systems
  - Common standards
  - Middleware
  - Mediator/Adaptor/Wrapper

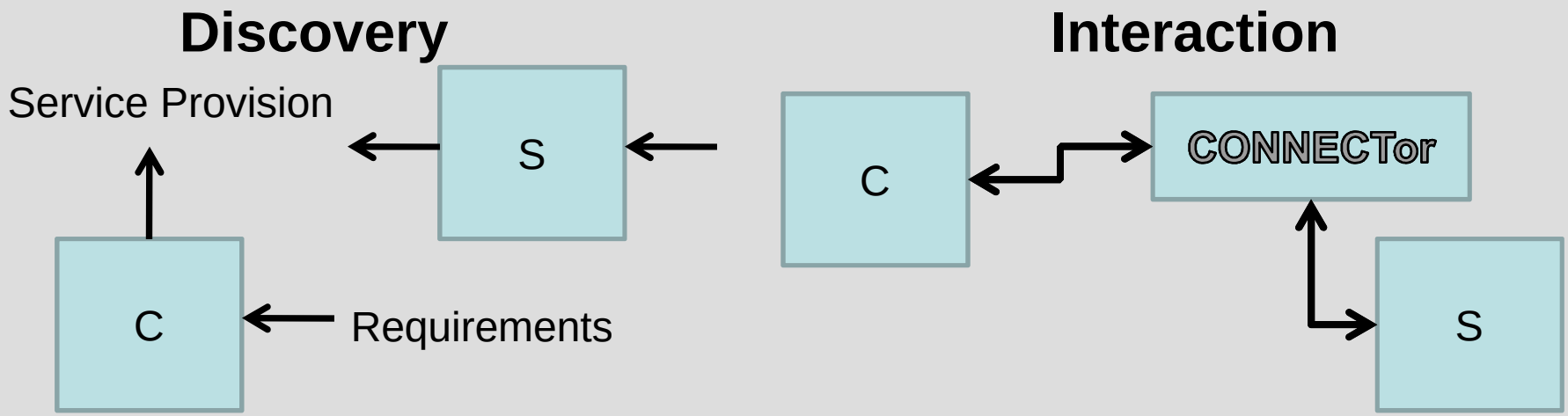
➔ Need for run-time interoperability

  ➔ No a priori knowledge of networked systems

  ➔ **No reference architecture / middleware / technology → "Next years legacy middleware"**

The Connect Approach

- Synthesize CONNECTors between heterogeneous communication endpoints
  - Generate middleware and application protocols to create connections that will overcome the interoperability barrier
    - CONNECTors devised and created at **Run Time**



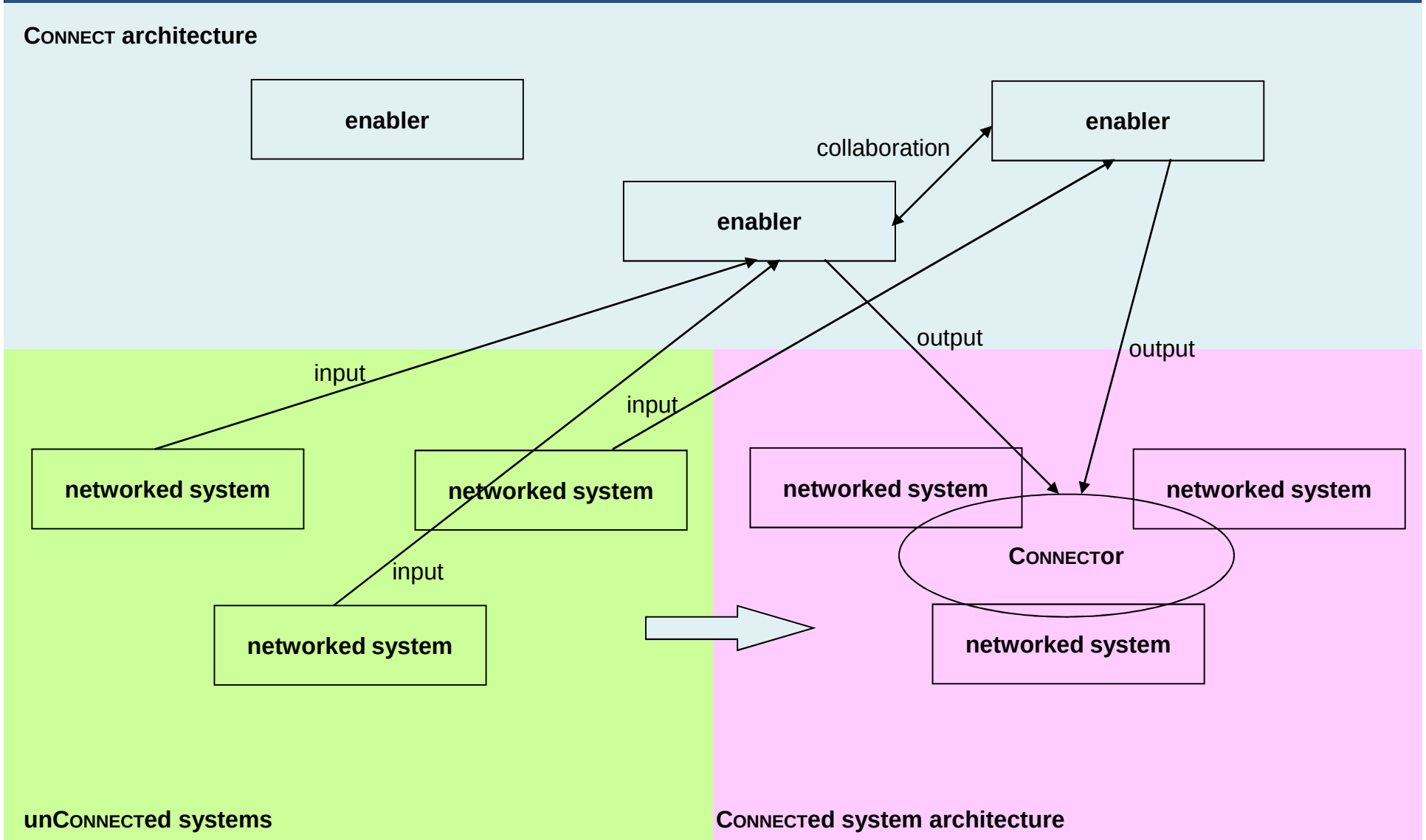
Technical Progress

- Study of interoperability solutions
- Identification of scenarios incl. detailed study of a stadium warning system
- Initial definition of the CONNECT architecture
- Validation of the initial architecture by working through popcorn scenario

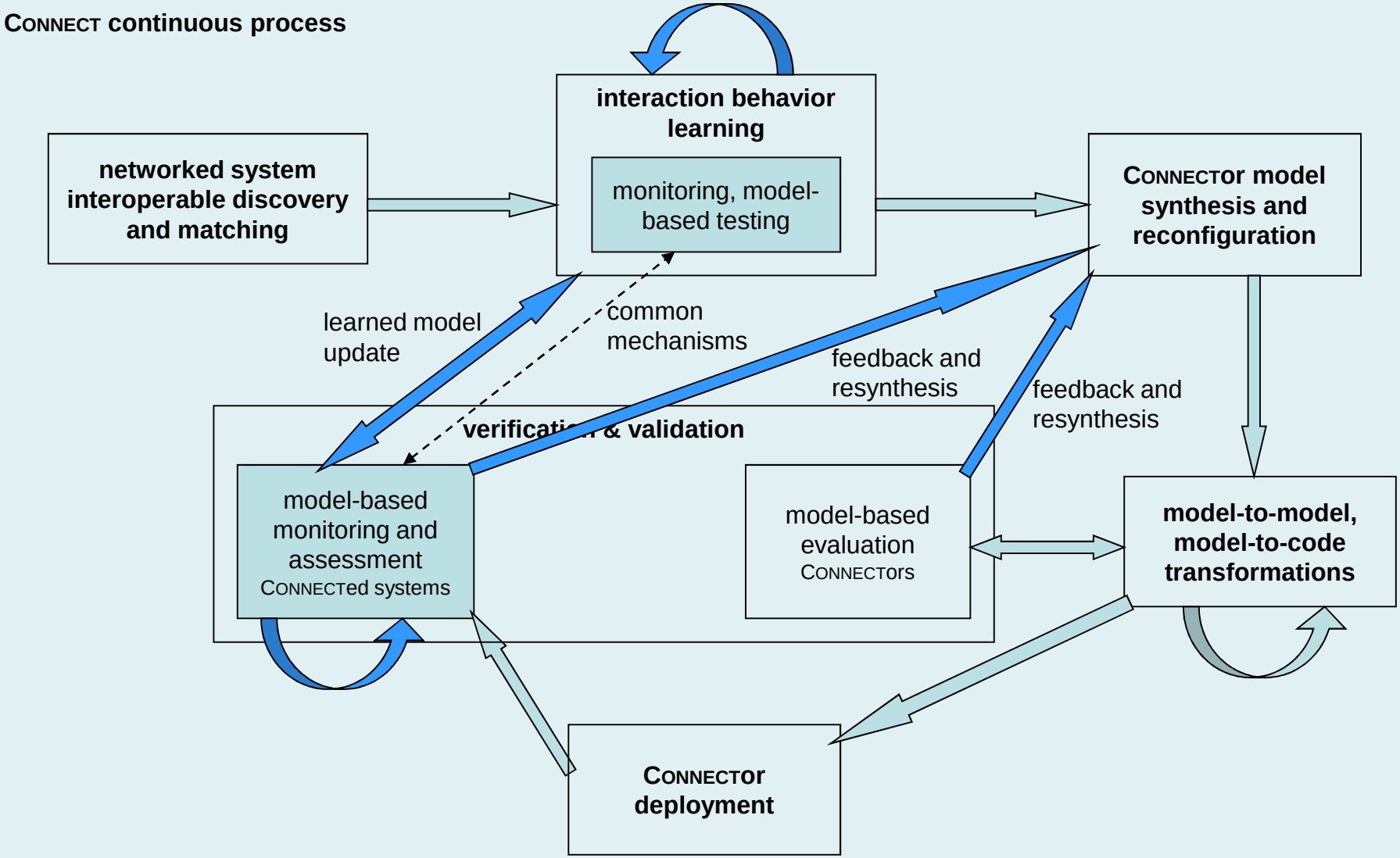
Use cases

- Ubiquitous computing
  - Environmental monitoring, Car-to-car communication, Warning systems
- Daily life support
  - Car parking, Airport boarding pass, Ticketing
- Telco Web2.0 & Cloud computing
  - Online user reputation enabler, Campus life planner, Telco presence and Web2.0

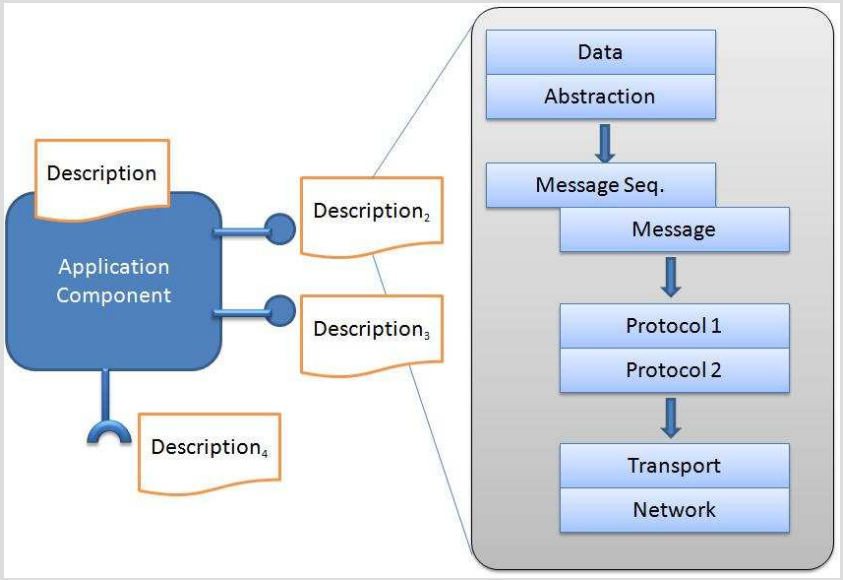
The Connect Architecture: An Overall View

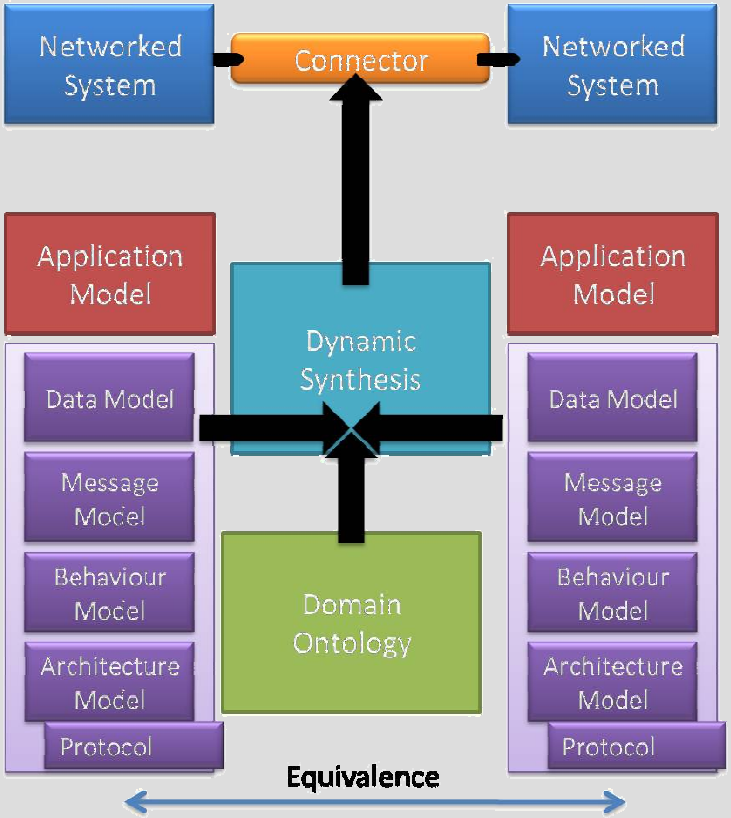


CONNECTed system life-cycle



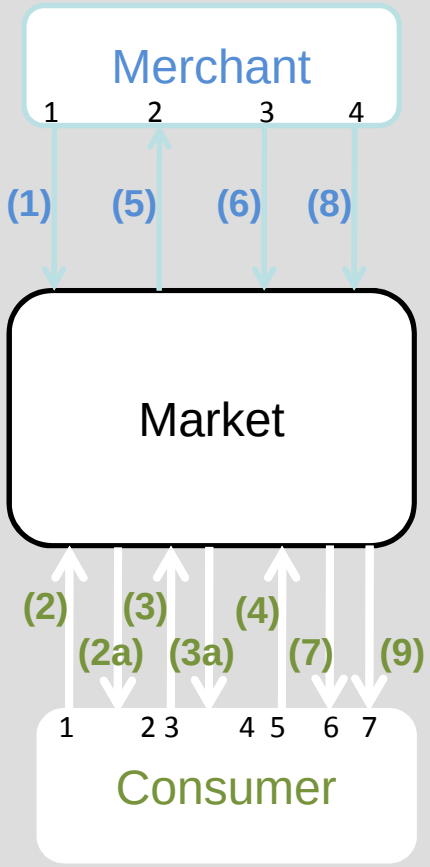
The Role of Self-description





Popcorn Scenario Description

- **Merchant**
  1. Publishes information about his product. (1)
  2. Receives a consumer request (5)
  3. Responds with Yes/No (6)
  4. If yes, then when close enough to the consumer, sends a proximity notification (buzz) and updates his product info (stock has decreased) (8)
- **Consumer**
  1. Browses all products available in the market (2, 2a)
  2. Selects a product (popcorn)
  3. Gets information about all popcorn merchants (3, 3a)
  4. Selects a popcorn merchant
  5. Submits a request to a particular merchant (4)
  6. Receives Yes/No response (7)
  7. If Yes, gets a proximity notification when the merchant is near him. (9)



The Popcorn Scenario: Progress

- Models of Application behaviour
  - Middleware messaging
- Use case
  - Tuple-space to UPnP binding
  - Learning of behaviour
    - Component self-description: WSDL Interfaces
  - Modelling of Applications/Middleware using LTS
    - Determined equivalence & produced model of connector
  - Practical prototype to be completed

Conclusions

- There has been considerable volume of research on interoperability in distributed systems; while progress has been made, the state of the art remains rather patchy, particularly when addressing the complexity of contemporary, highly heterogeneous distributed systems.
- CONNECT aims to :
  - Identify a common framework for Emergent Middleware
    - Covering discovery, interaction and quality of service;
  - Ensure semantically correct discovery and interaction;
  - Automatically synthesize Emergent Middleware.
- These are big challenges that can revolutionize the state of the art in distributed systems in general and middleware more specifically.