

WS-TOPOLOGY: TOPOLOGY CONFIGURATION AND COORDINATION FOR SERVICE ORIENTATED ARCHITECTURES



Author: Ronan Barrett - rbarrett@computing.dcu.ie

Supervisor: Dr Claus Pahl

School of Computing, Dublin City University, Dublin, Ireland

Question?

Given a system utilising distributed components, can it's topology configuration be separated from the underlying distributed system or collaboration description to provide for reliable re-configuration and assembly?

What do we do?

This project addresses the lack of flexibility within Service Orientated Architectures (SOA). Current solutions do not allow the alteration of the topology after system development, which inhibits the mutability of such systems. These SOAs do not isolate the topology configuration from the rest of the distributed system. Instead the distributed components refer directly to each other (whether in code or in some collaboration description language), resulting in the hard coding of the topology configuration.

Why do this?

Topology configurations exhibit varying performance for a number of metrics based on the chosen configuration. This implies that having the ability to vary the configuration should provide for improved system evolution and varying levels of scalability, availability and performance.

Objectives

The objective of this project is to provide for improved maintainability/mutability and also to provide flexibility with regard to performance, scalability and availability of SOAs.

How Will We Achieve Our Objectives?

We intend to achieve these goals by separating the topology configuration from the underlying SOA and also from the collaboration description, if any. To achieve this objective a number of deliverables will be produced.

What is a Collaboration?

We consider a collaboration to be a number of Web Services working together to achieve a goal different to their discrete goals. A collaboration is the coming together of a number of elementary Web Services to form a complex Web Service. Collaborations come in two flavours, Orchestrations and Choreographies.

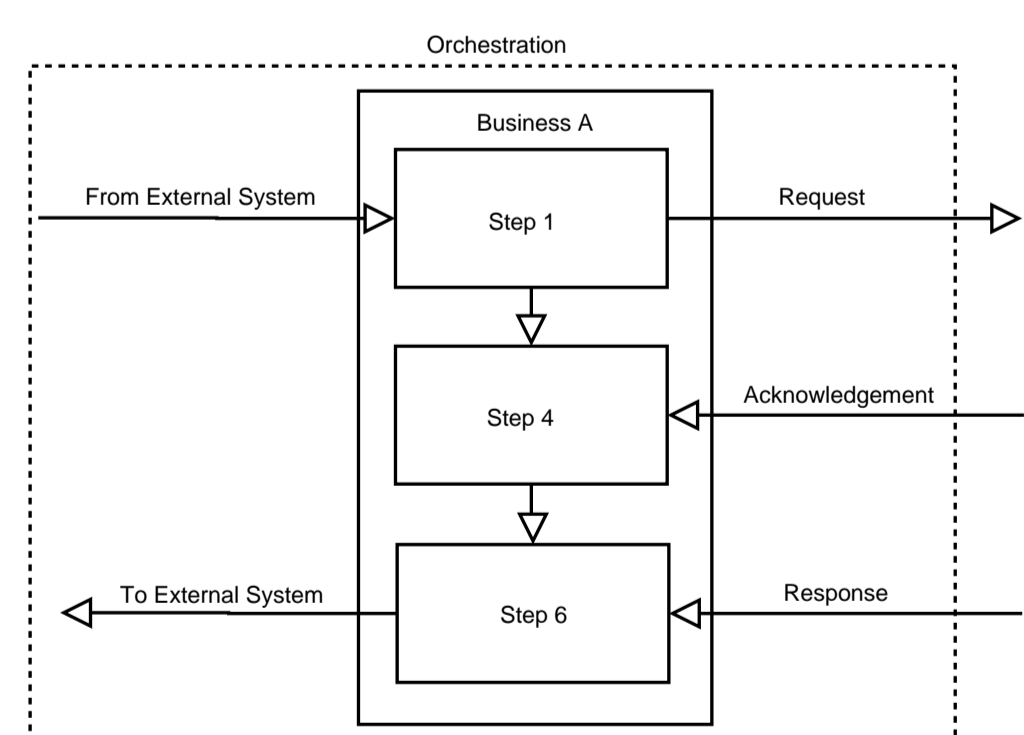


FIGURE 1: Orchestration models the internals of a private process.

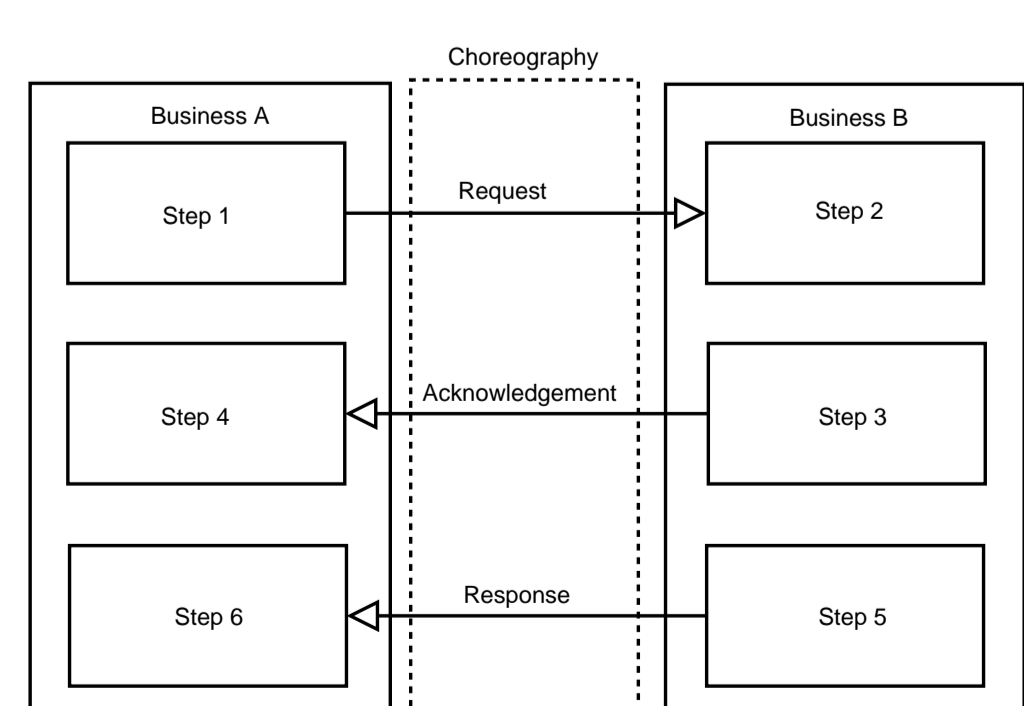


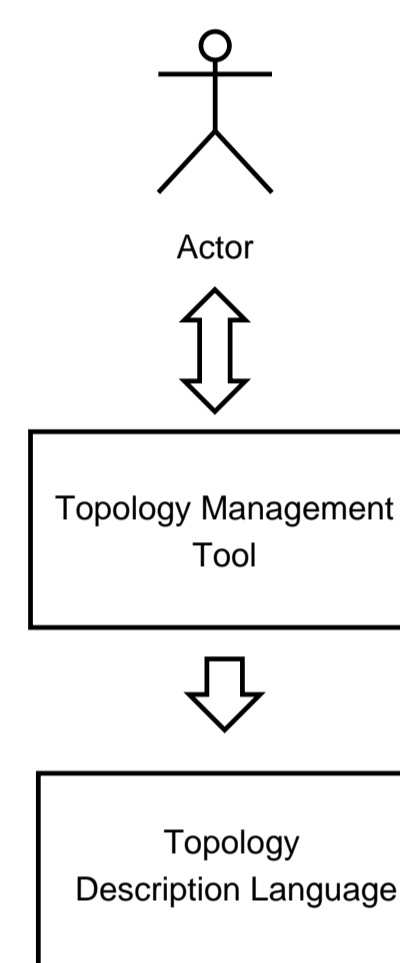
FIGURE 2: Choreography models only public message exchange.

Deliverables

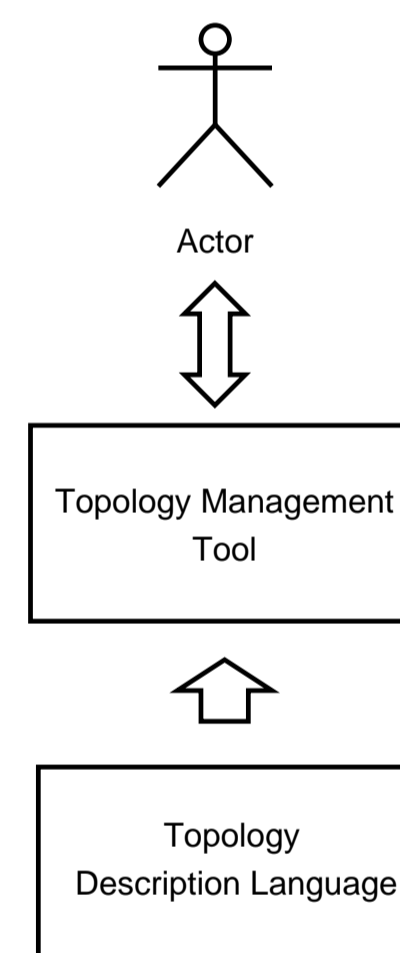
- A process model for breaking up of existing candidate systems.
- A Topology Description Language (TDL) which specifies how the nodes of an SOA are laid out.

TDL	Service Topology
WS-CDL/BPEL4WS	Service Flow
UDDI	Service Discovery
	Service Publication
WSDL	Service Description
SOAP	Messaging
HTTP,JMS,SMTP etc.	Transport

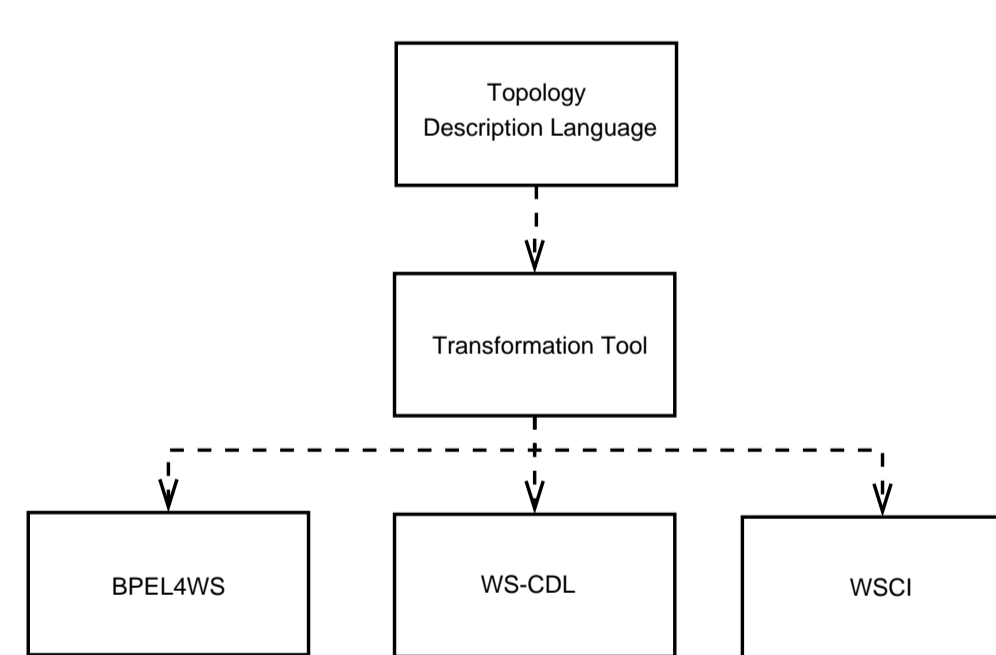
- A Topology Management Tool to create and manipulate the TDL.



- A graphical representation of the topology of the system based on the TDL.



- A Transformation Tool will allow the TDL to be transformed at development time to a lower level collaboration description language.



- Coordination messages required for the reliable execution of the selected topology must also be inserted by the tool into the collaboration description instance created by the transformation tool.
- A Coordination Service will be required at each distributed component to ensure the topology executes reliably.

Some Possible Topologies

Any of the following topology configurations can be applied to a distributed system using our tools and specifications.

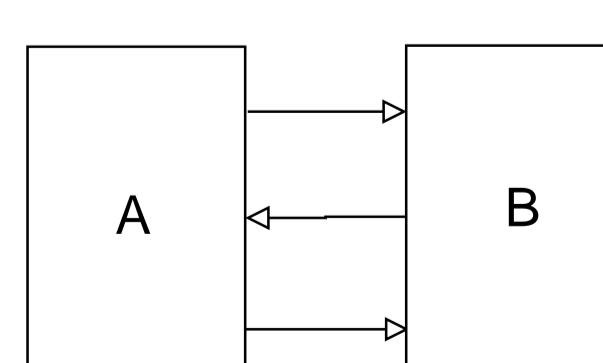


FIGURE 3: Binary or Point-to-Point topology.

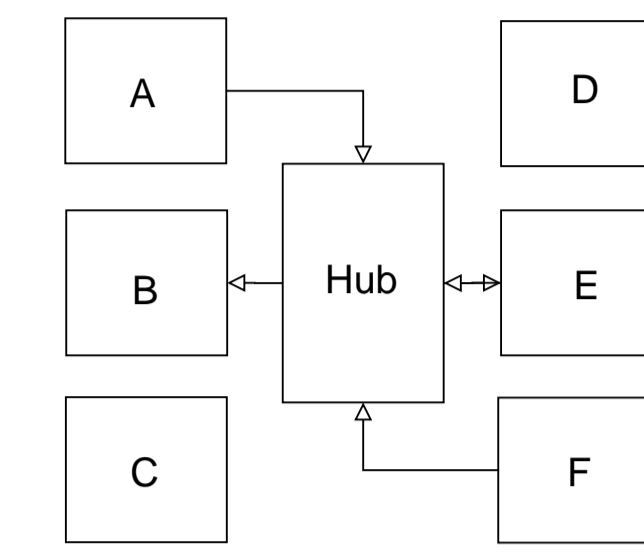


FIGURE 4: Star or Hub and Spoke topology.

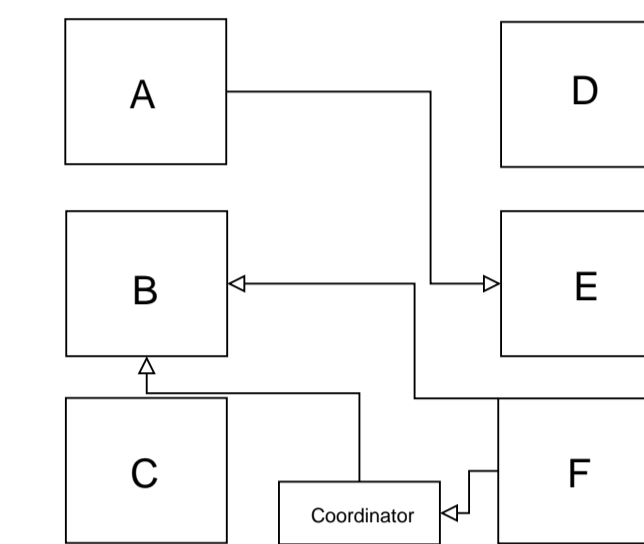


FIGURE 5: Peer-to-Peer topology.

Applications?

We intent to implement a case study to verify our work. This case study will be a Teaching and Learning Environment (TLE) based on the Learning Technology Systems Architecture (LTSA) reference architecture identified by the IEEE as a generic educational system.

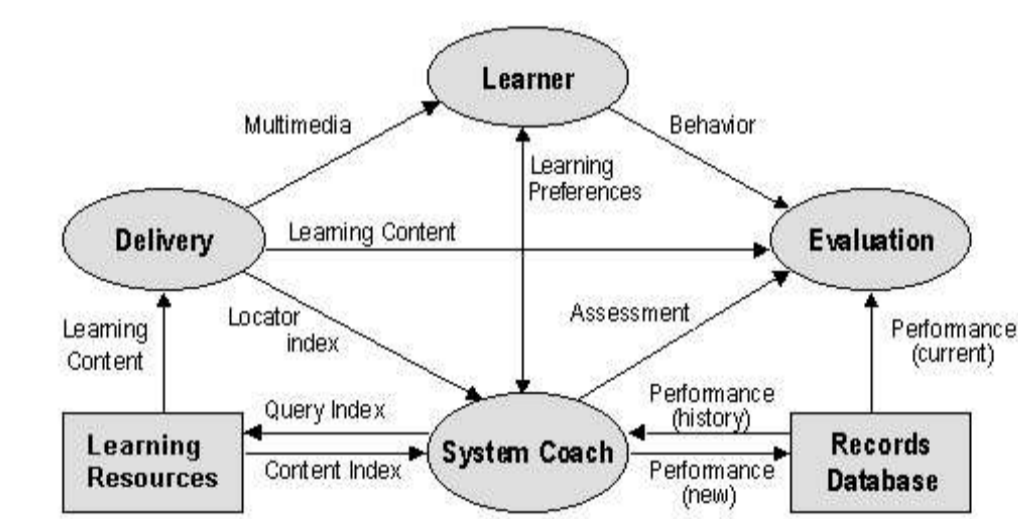


FIGURE 6: LTSA reference architecture.

Results

- Modelled LTSA message flow using Pi-Calculus to verify its SOA characteristics.
- Identified SOA specific characteristics.
- Identified that SOAs are excellent candidates for topology configuration.
- Distinguished between two forms of patterns, architectural patterns and core patterns.
- Identified relevant patterns.
- Documented process model for breaking up candidate systems.

Future Work

- Consolidate our research hypothesis by obtaining further proof of concept.
- Conceptualise and document supporting specifications.
- Build tools to support our work.
- Build case study to prove usefulness of solution.

Acknowledgements

This project is funded by: Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan.



This poster was created with L^AT_EX on Fedora Core 2.

