Session VI_a: **Service Creation**

Chair: Joe Sventek, *Hewlett-Packard*

Rapid Service Development on a TINA-Based Service Deployment Platform

Jack Verhoosel
Martin Wibbels
John-Luc Bakker
Harold Batteram

Outline

- Introduction
- MESH platform overview
- Rapid Service development
- Conclusion
- Future work

MESH objectives

- To accelerate the introduction of advanced telematics services for the ESH.
- To study and explore new telematics services and possibilities.
- To design, implement and validate a service platform for the ESH.

Application domains

- Tele-consultation in the health sector.
- Teamwork between university lectures.
- Tele-learning for students.
- Tele-meeting in a distributed organization.

MESH partners

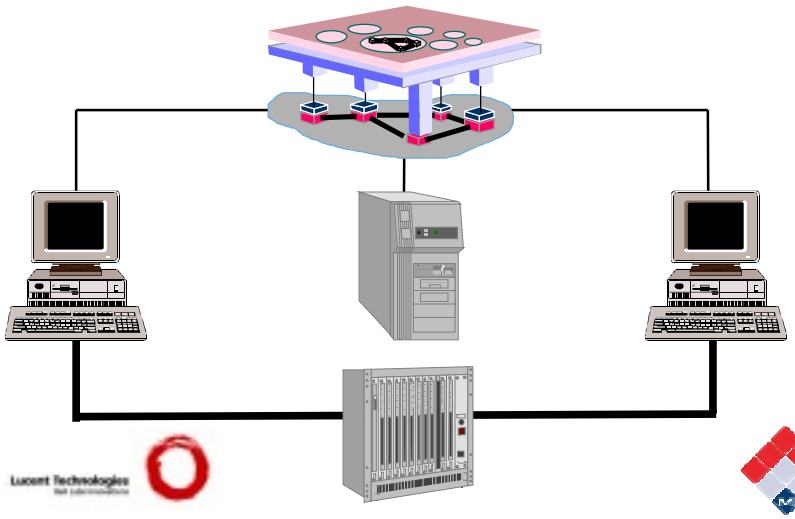
- Lucent Technologies
- SURFnet
- KPN Research
- Telematics Institute
- Centre for Telematics and Information Technology (CTIT)

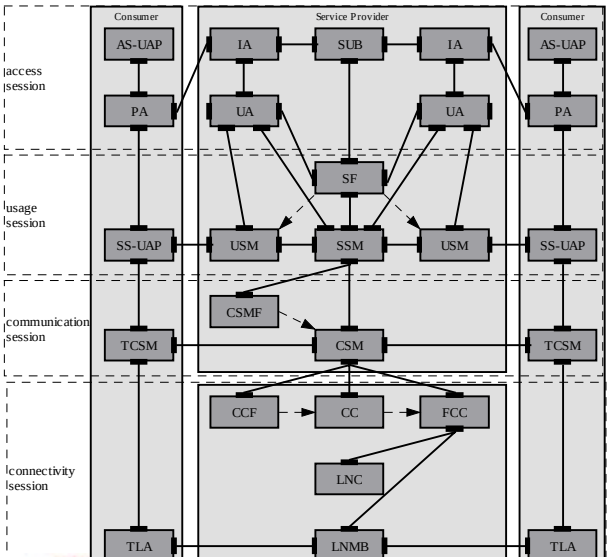
(http://www.mesh.nl/extern/english.htm)

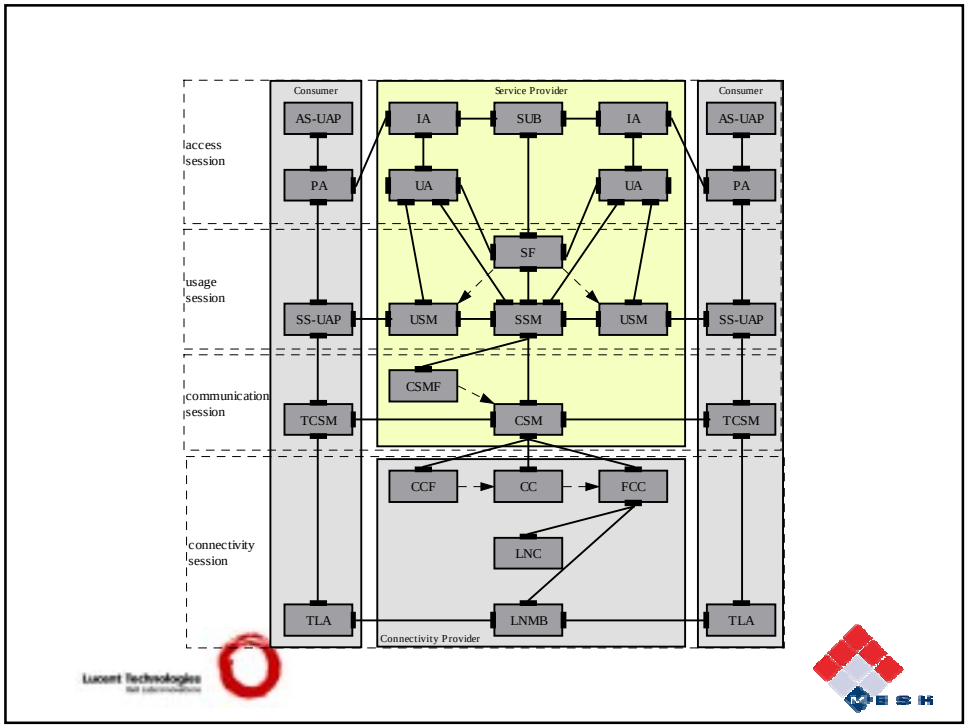
MESH users

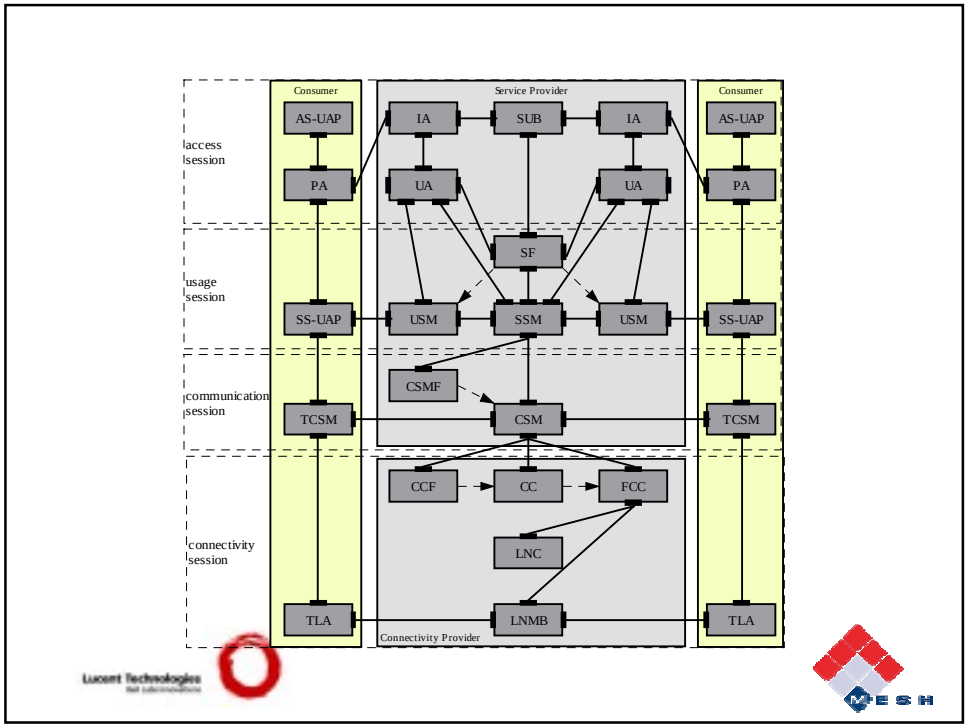
- Academic Hospital University Amsterdam
- Roessingh Research and Development
- Delft University of Technology
- University of Twente

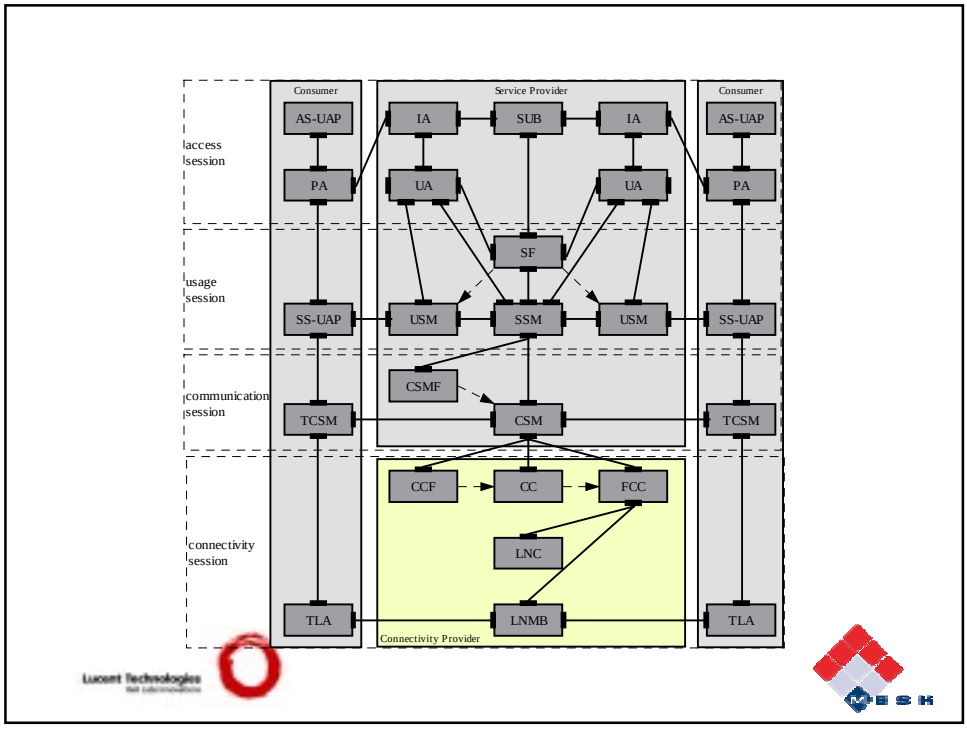
MESH overview

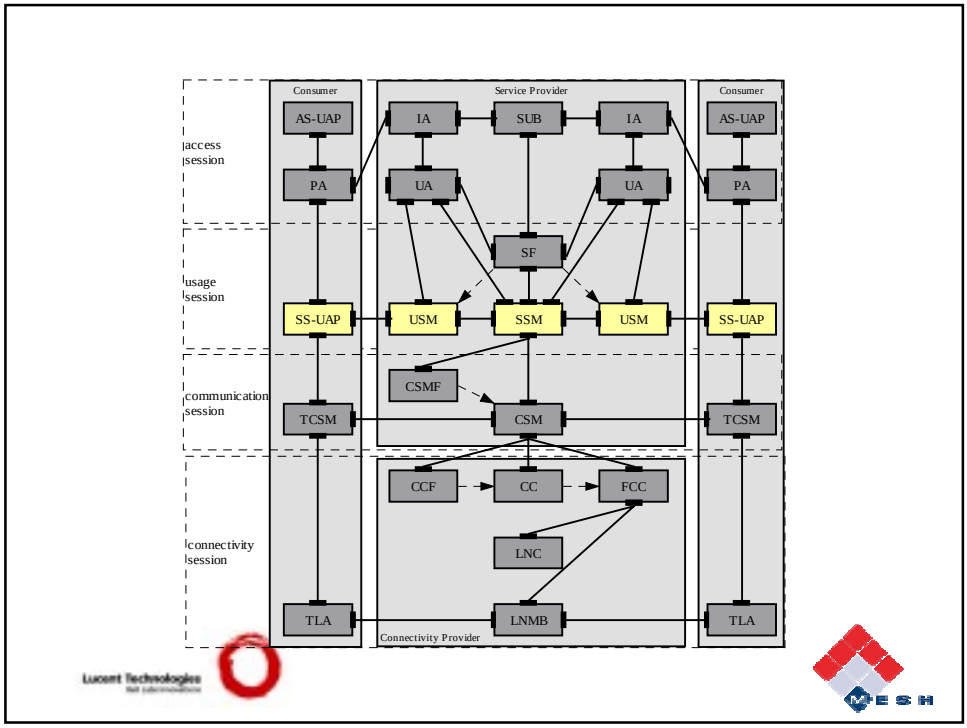





Consumer
Service Provider
Consumer
AS-UAP
IA
SUB
IA
AS-UAP
access session
PA
UA
UA
PA
SF
usage session
SS-UAP
USM
SSM
USM
SS-UAP
CSMF
communication session
TCSM
CSM
TCSM
CCF
CC
FCC
connectivity session
LNC
TLA
LNMB
TLA
Connectivity Provider
Lucent Technologies
MESH

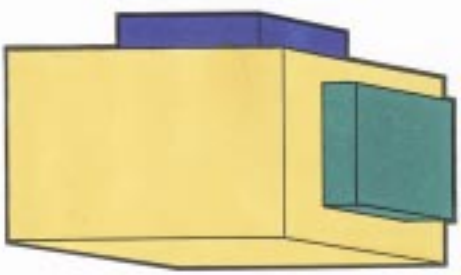

Consumer
Service Provider
Consumer
AS-UAP
IA
SUB
IA
AS-UAP
access session
PA
UA
UA
PA
SF
usage session
SS-UAP
USM
SSM
USM
SS-UAP
CSMF
communication session
TCSM
CSM
TCSM
CCF
CC
FCC
connectivity session
LNC
TLA
LNMB
TLA
Connectivity Provider
Lucent Technologies
MESH


Consumer
Service Provider
Consumer
AS-UAP
IA
SUB
IA
AS-UAP
access session
PA
UA
UA
PA
SF
usage session
SS-UAP
USM
SSM
USM
SS-UAP
CSMF
communication session
TCSM
CSM
TCSM
CCF
CC
FCC
connectivity session
LNC
TLA
LNMB
TLA
Connectivity Provider
Lucent Technologies
MESH


Consumer
Service Provider
Consumer
AS-UAP
IA
SUB
IA
AS-UAP
access session
PA
UA
UA
PA
SF
usage session
SS-UAP
USM
SSM
USM
SS-UAP
CSMF
communication session
TCSM
CSM
TCSM
CCF
CC
FCC
connectivity session
LNC
TLA
LNMB
TLA
Connectivity Provider
Lucent Technologies
MESH

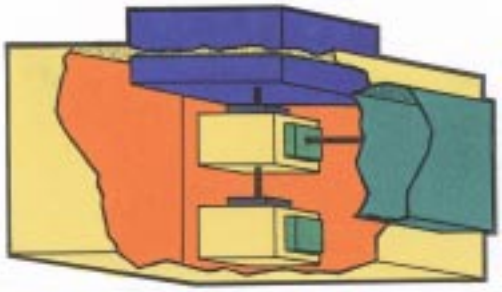
Distributed Software Component Framework

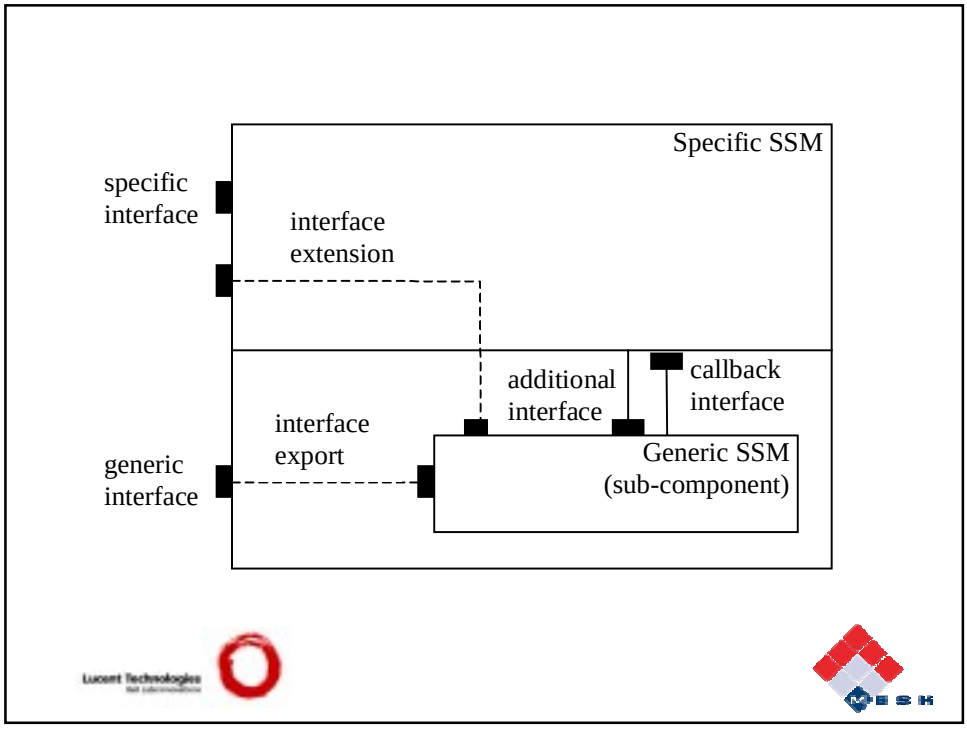
- Encapsulates distr. programming logic
- High level software building blocks
- Interaction through IDL-interfaces

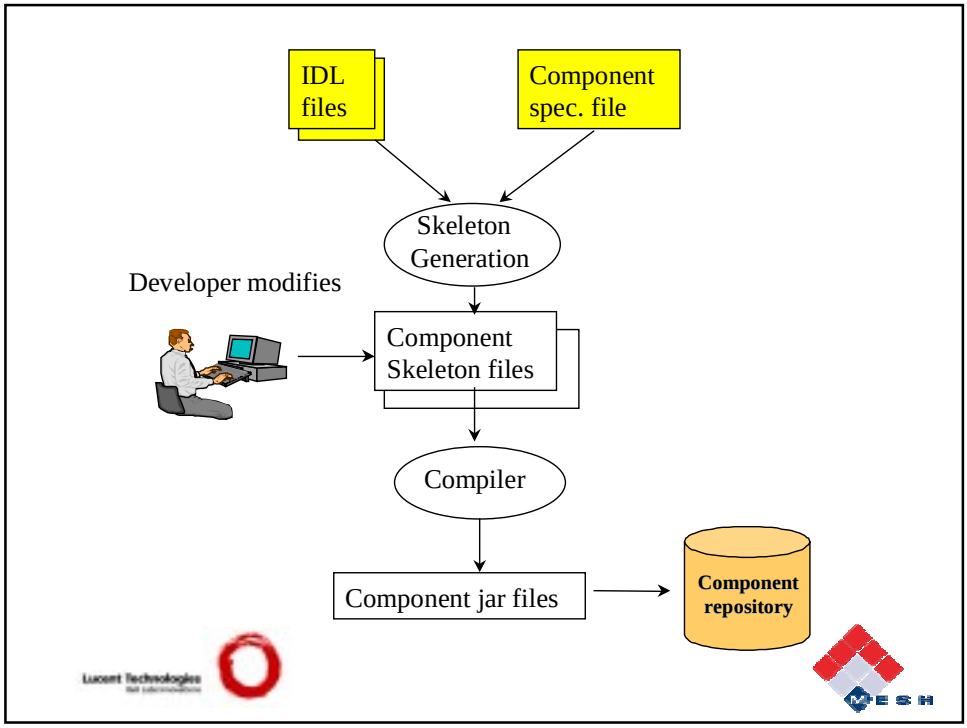


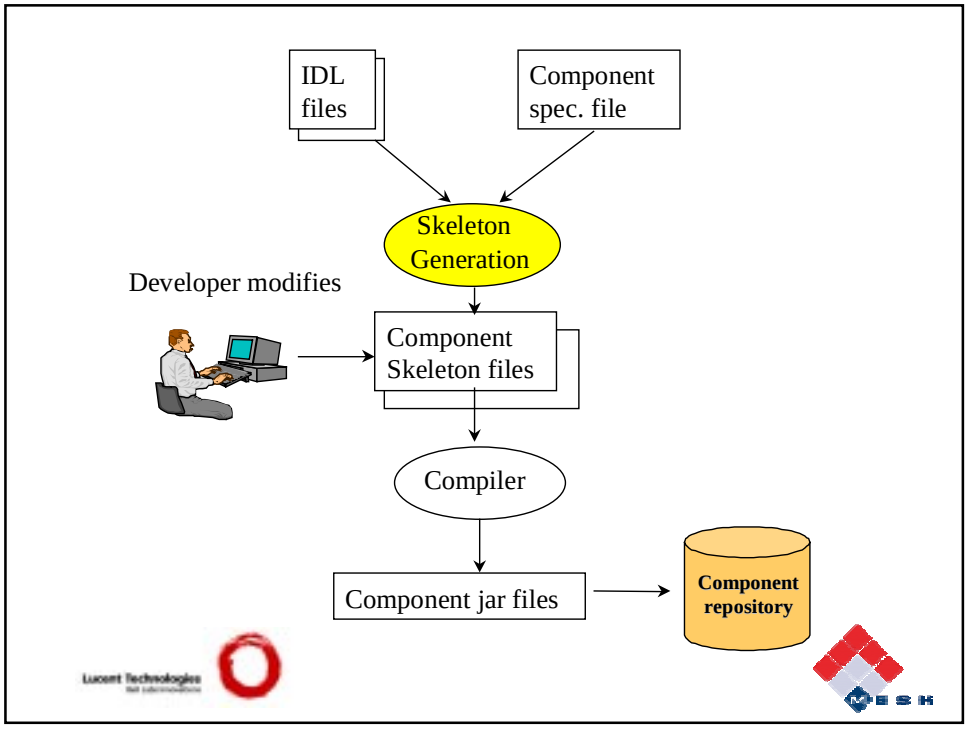
Compound component

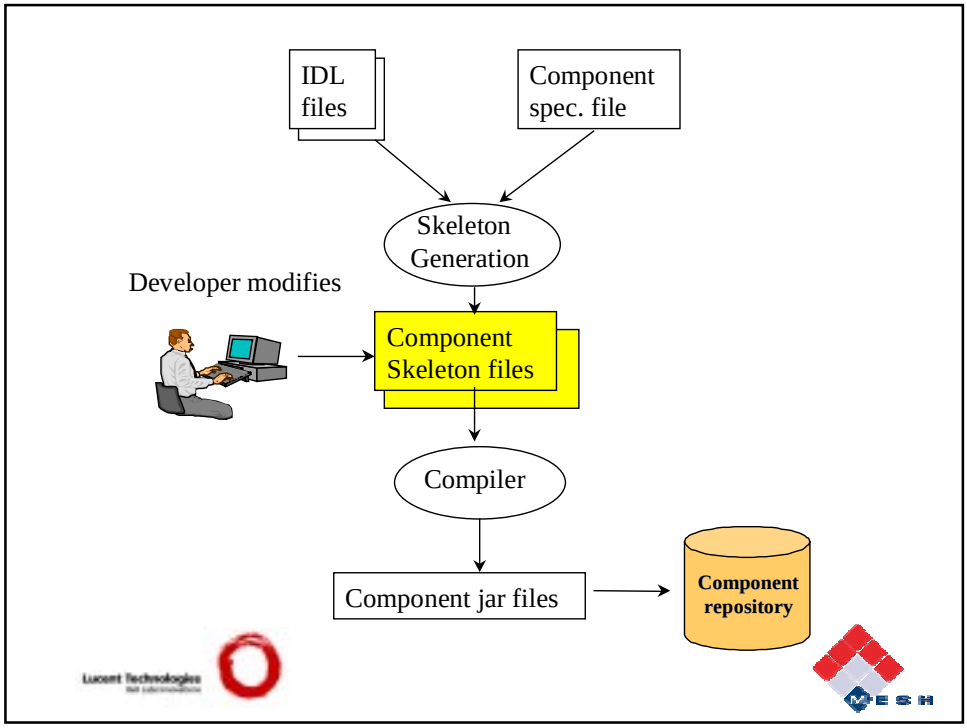
- Specialization through aggregation
- Dynamic composition
- Selective export of interfaces

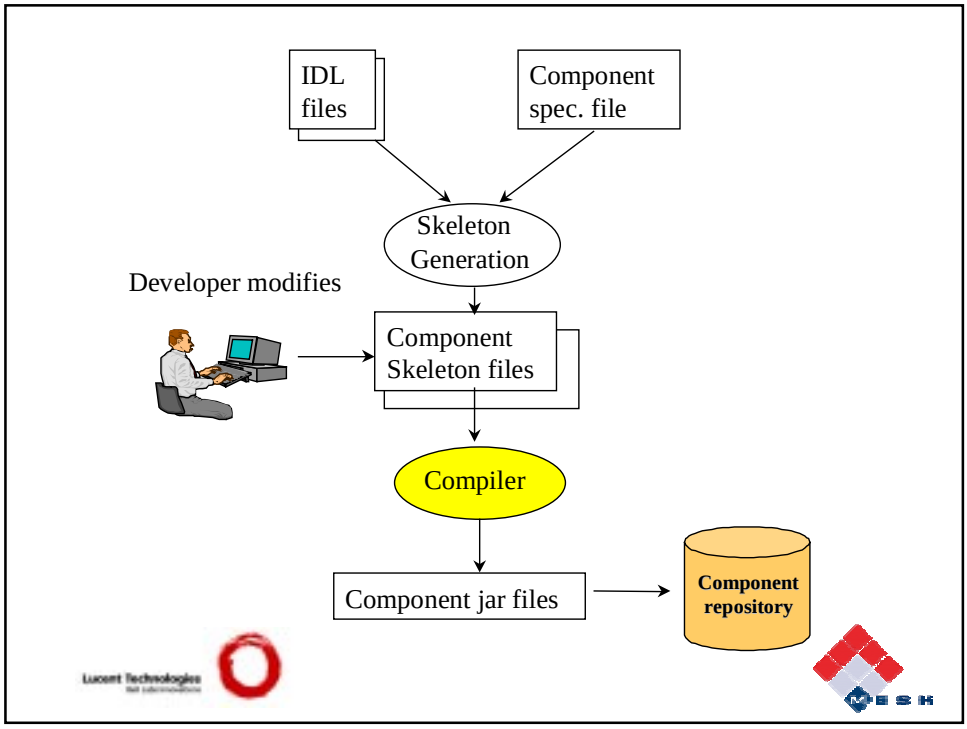


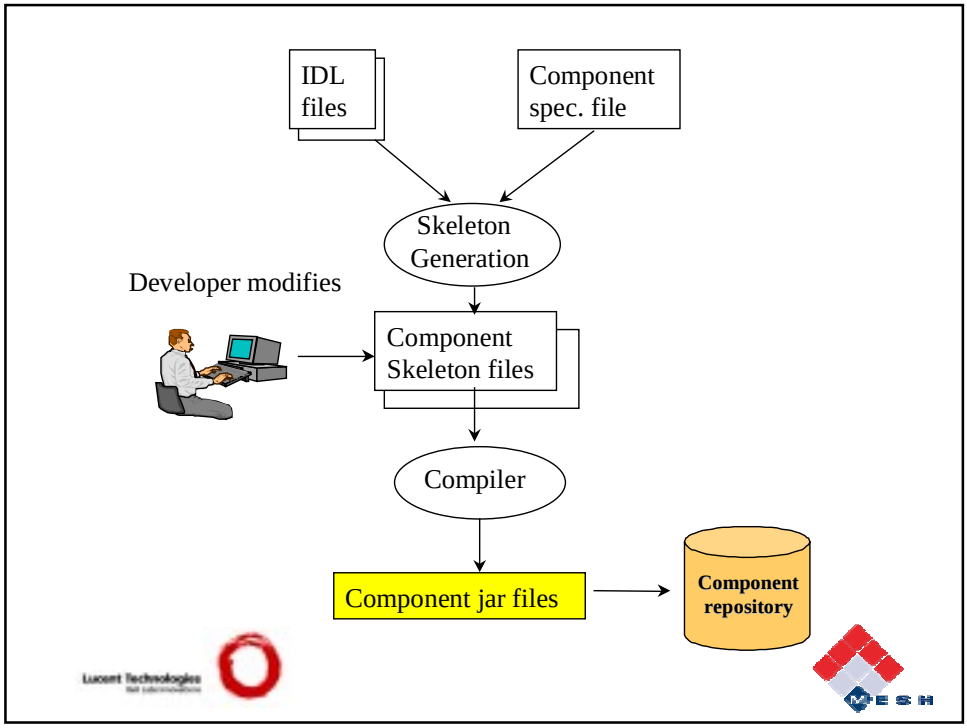

Specific SSM
specific interface
interface extension
additional interface
callback interface
interface export
generic interface
Generic SSM (sub-component)


IDL files
Component spec. file
Skeleton Generation
Developer modifies
Component Skeleton files
Compiler
Component jar files
Component repository

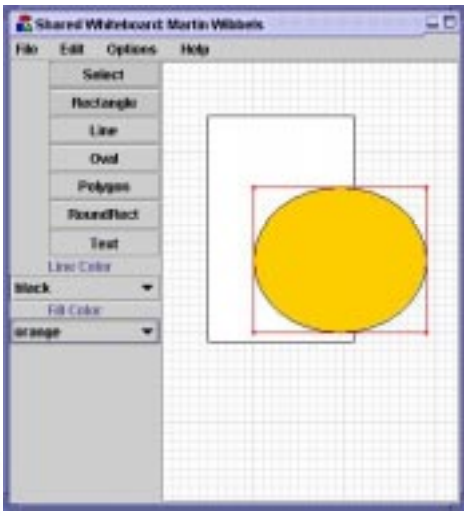

IDL files
Component spec. file
Skeleton Generation
Developer modifies
Component Skeleton files
Compiler
Component jar files
Component repository


IDL files
Component spec. file
Skeleton Generation
Developer modifies
Component Skeleton files
Compiler
Component jar files
Component repository


IDL files
Component spec. file
Skeleton Generation
Developer modifies
Component Skeleton files
Compiler
Component jar files
Component repository

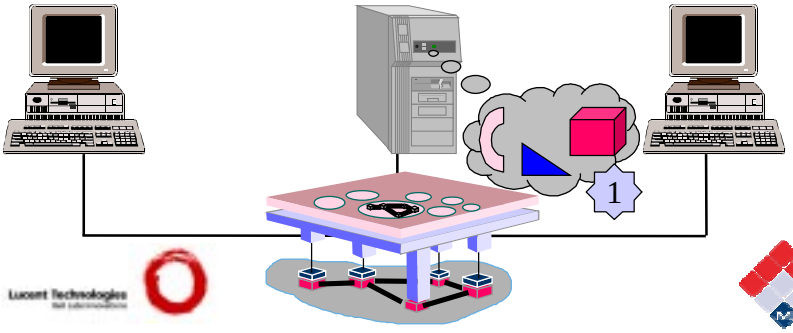

IDL files
Component spec. file
Skeleton Generation
Developer modifies
Component Skeleton files
Compiler
Component jar files
Component repository

Shared Whiteboard



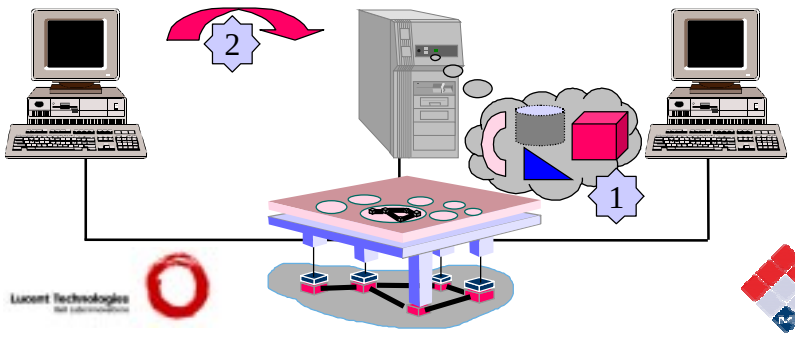
Shared white board service

- Service provider maintains white board state



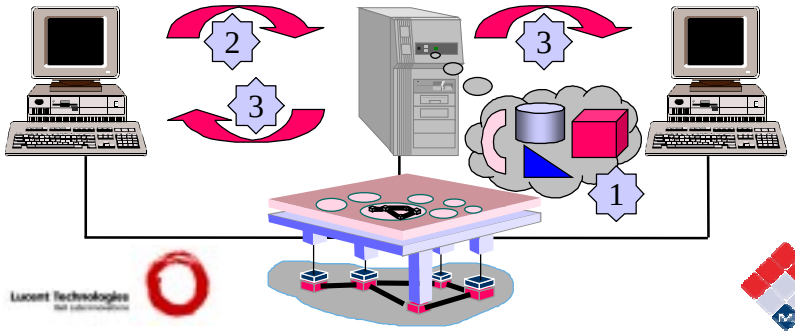
Shared white board service

- Service provider maintains white board state
- Change state of white board



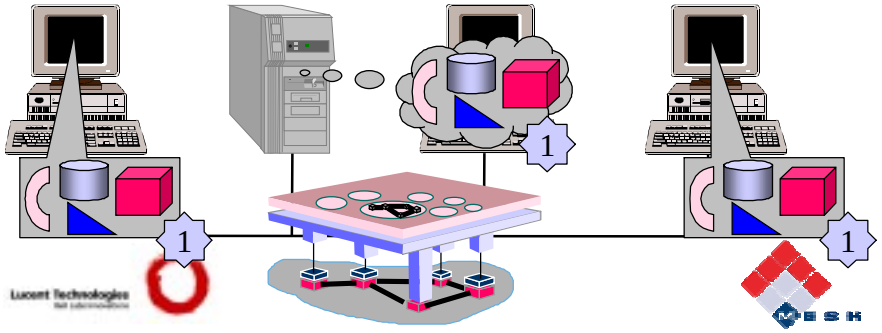
Shared white board service

- Service provider maintains white board state
- Change state of white board
- Distribute new state



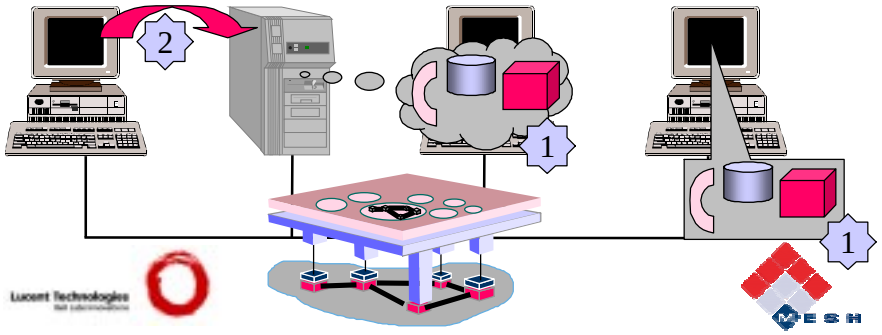
User mobility

- Two users participate in a white board service



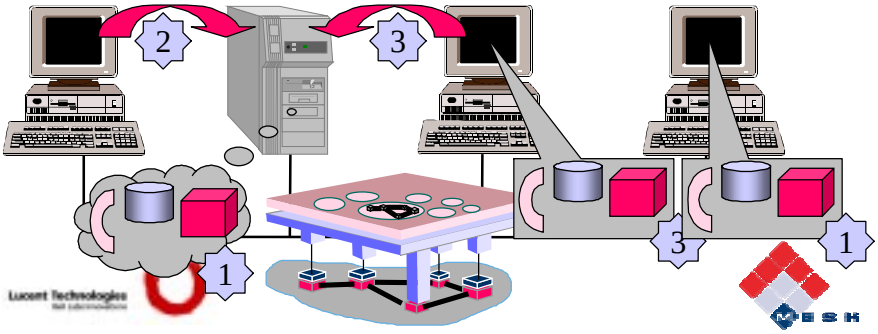
User mobility

- Two users participate in a white board service
- One user suspends the service session

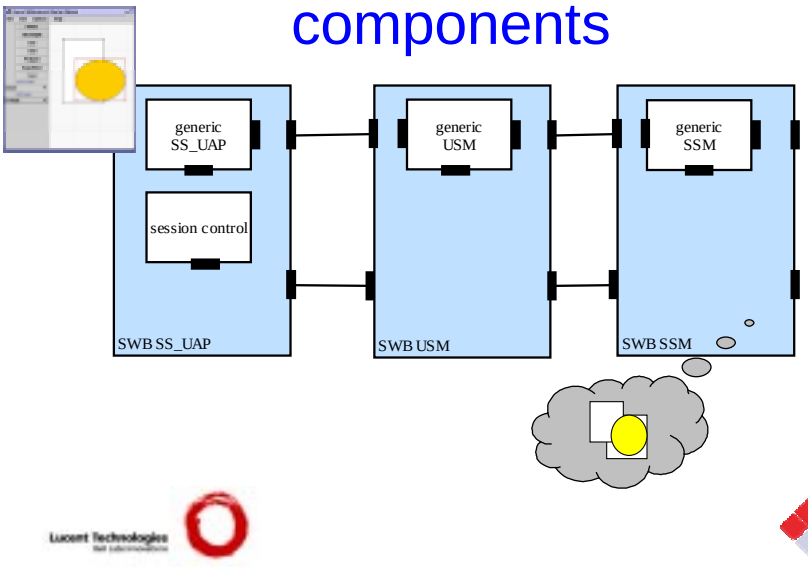


User mobility

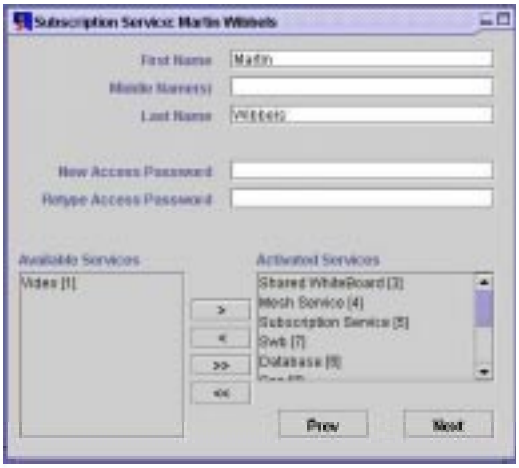
- Two users participate in a white board service
- One user suspends the service session
- The session is resumed from another terminal



Shared Whiteboard components



Subscription Service



Shared Database

Conferencing Service

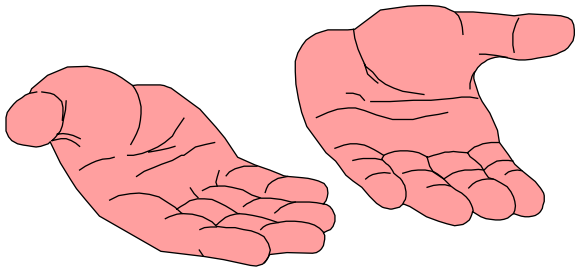
Conclusion

- TINA provides a solid architecture with many generic features for multi party, multi media services.
- Interfaces to extend SS-UAP, USM and SSM should also be standardized.
- The combination of TINA and the DSC framework allow rapid service development.

Future work

- New project FRIENDS.
- Focus on service creation and deployment at a larger scale.
- Based on the TINA and MESH experiences.

Questions?



Framework and Paradigm based Process for TINA Service Creation

and

Engineering of Interworking TINA-based Telecommunications Services

Dr Richard Sinnott

GMD Fokus

Berlin, Germany

Mario Kolberg

University of Strathclyde

Glasgow, Scotland

Overview of Presentation

- Brief introduction to TOSCA
  - what we want to do and who we are
- Service creation in TOSCA
  - paradigms and frameworks
  - TOSCA Process Model Outline
- Description of SDL based Service Engineering
  - outline of Service Federation
- Demo

TOSCA Intentions

Develop tools/techniques for service creation

- should speed up service creation
- should widen service creation audience
- should not be detrimental to service quality

Propose approach based on paradigm tools, OO frameworks and validation techniques

- includes validation of isolated and deployed services, i.e. potentially interworking (later!)

The TOSCA Consortium

Service Creation in TOSCA

Approach based on Paradigm tools

- Movies, Functional Block, ... (more later!)

OO Frameworks

- capture design experience that can be used to create a broad class of similar(?) services
- have specific places (holes) where specific service behaviour can be inserted

Paradigm based Service Creation

Allows for service complexity to be hidden

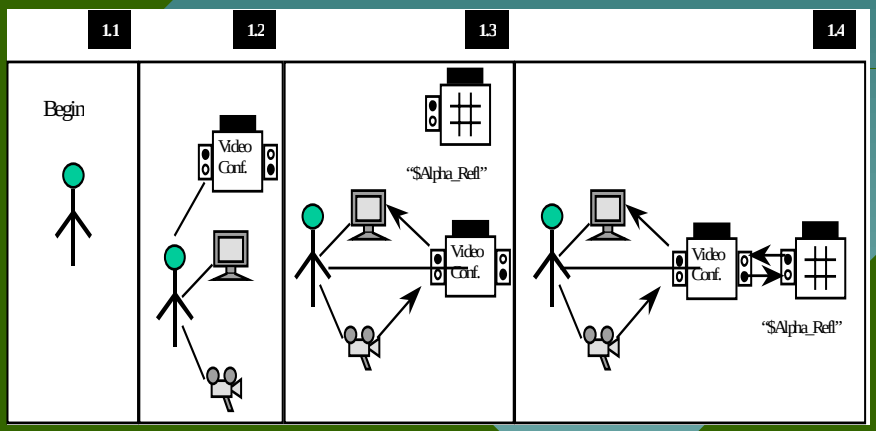
- user of paradigm tool need not know specifics of framework, e.g. whether in C++/CORBA, SDL, ...
- output of tool is specialising code or specification fragments that complete service, i.e. fill in the holes!

TOSCA Paradigm Tools

Functional block paradigm user constructs service through combination of graphical building blocks (more later!)

Movies paradigm user constructs service through sequence of snapshots that highlight expected service behaviour - *ala* a movie sequence

Typical Movie Sequence



Object-Oriented Frameworks

Capture generic and re-useable design that can be applied to create class of services

- includes semantics!!!

Have predefined holes where service creators can add/refine behaviour

TOSCA built frameworks around service session

- start, stop, suspend, resume user/service sessions

Developing OO Frameworks

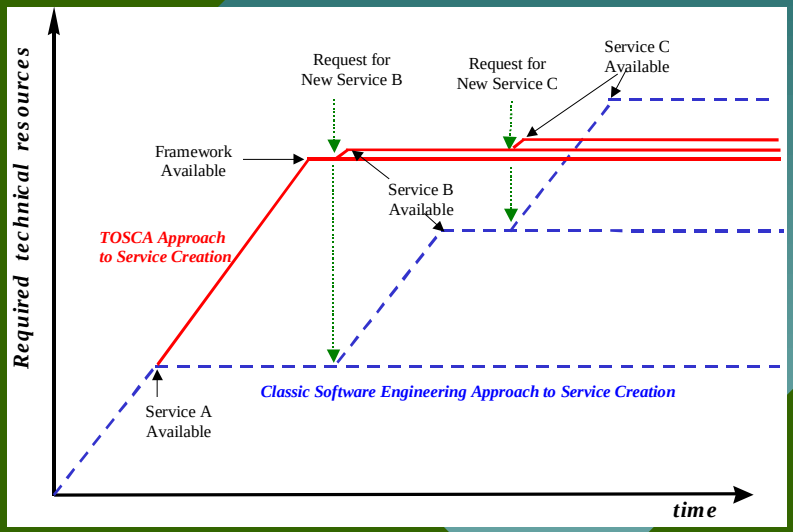
TOSCA has developed OO Frameworks in

- C++/CORBA
- SDL

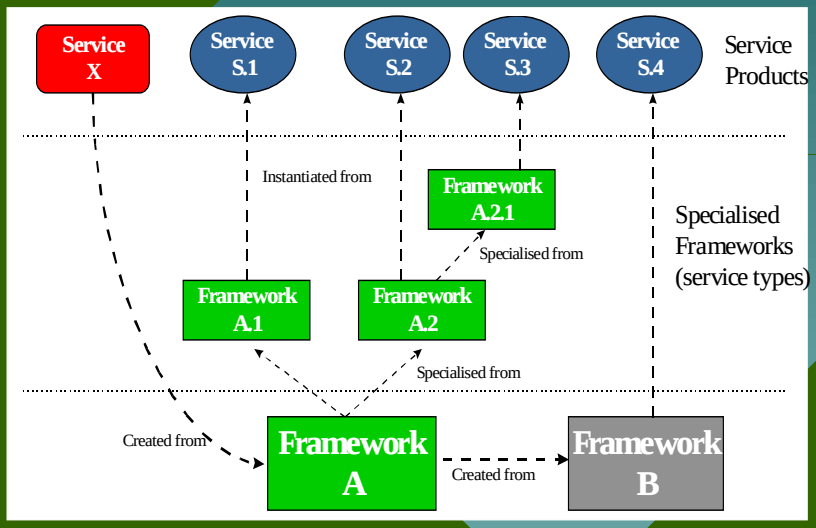
Starting point is ODL/IDL description

More later on framework development, relation between different frameworks and services derived from frameworks!!!

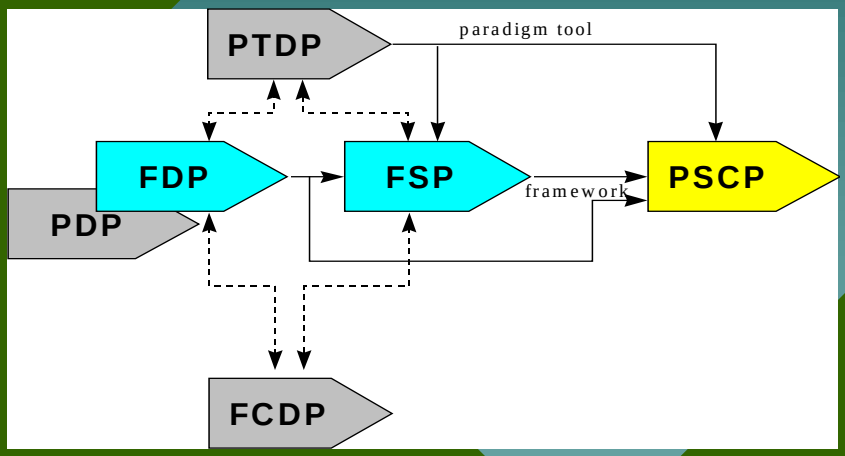
Framework based Service Creation



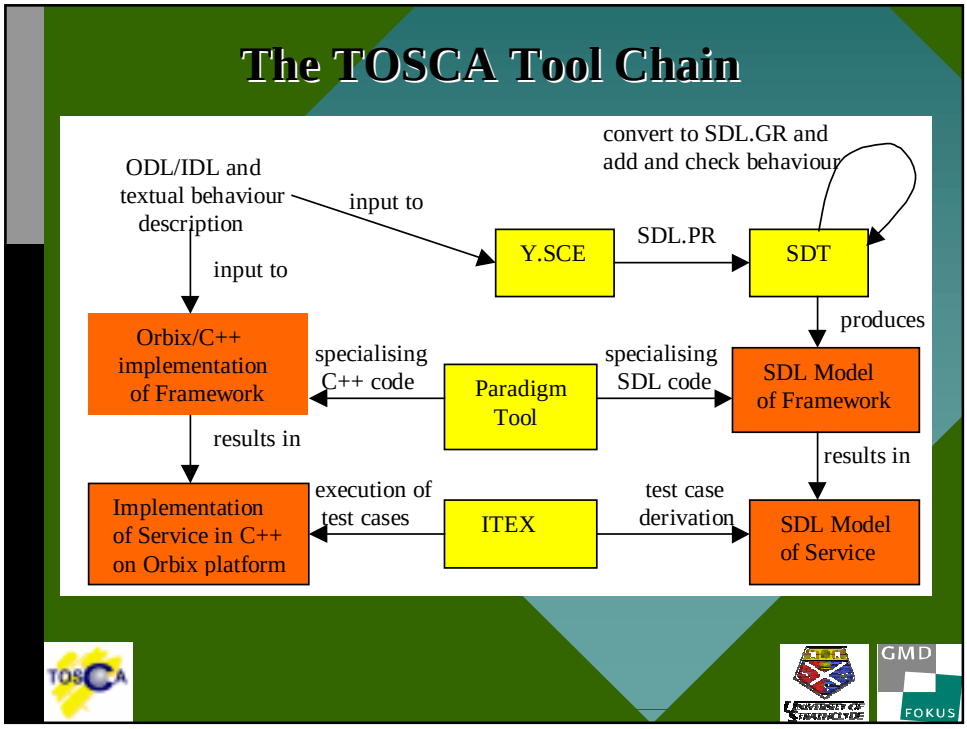
Framework based Service Creation

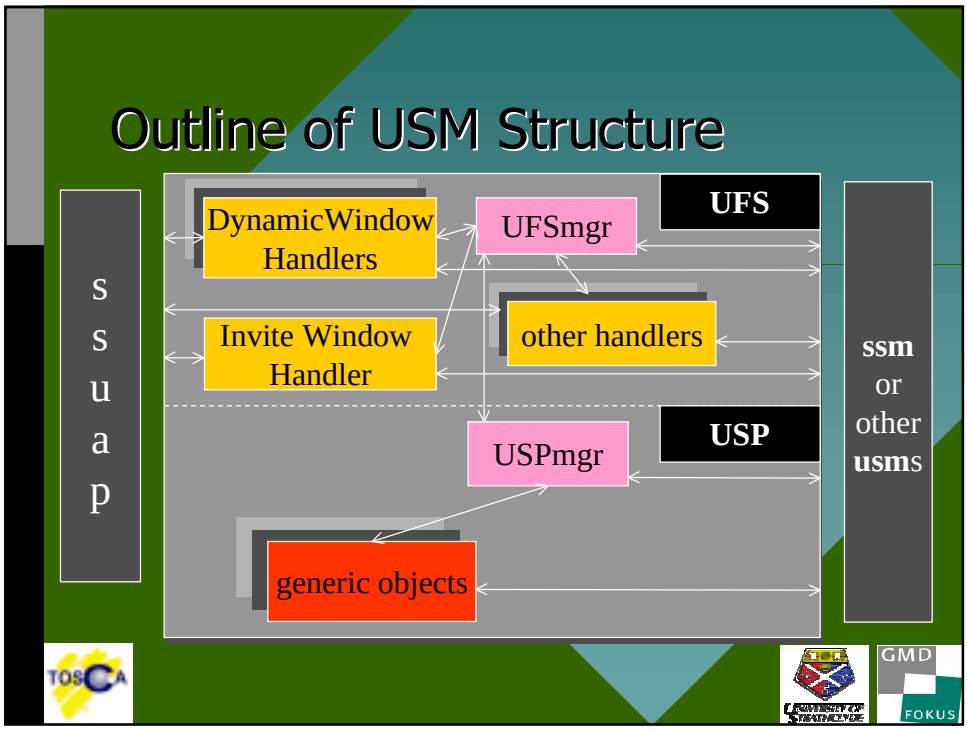


TOSCA Process Model



Over to Mario!


The TOSCA Tool Chain
ODL/IDL and textual behaviour description
input to
Y.SCE
SDL.PR
SDT
convert to SDL.GR and add and check behaviour
input to
produces
Orbix/C++ implementation of Framework
specialising C++ code
Paradigm Tool
specialising SDL code
SDL Model of Framework
results in
results in
Implementation of Service in C++ on Orbix platform
execution of test cases
ITEX
test case derivation
SDL Model of Service
TOSCA
GMD FOKUS


Outline of USM Structure
ssuap
DynamicWindow Handlers
UFSmgr
UFS
Invite Window Handler
other handlers
USP
USPmgr
generic objects
ssm or other usms
TOSCA
GMD FOKUS

IDL for USM manager

```
interface i_UFSmgr : i_CO_lifecycle
{
  void suspendSessionRequest();
  void terminateSessionRequest();
  void requestObject(inout NamedObject obj);
  oneway void ufsstart();
  oneway void ufsstop();
  oneway void ufssuspend();
  oneway void ufsresume();
};
```

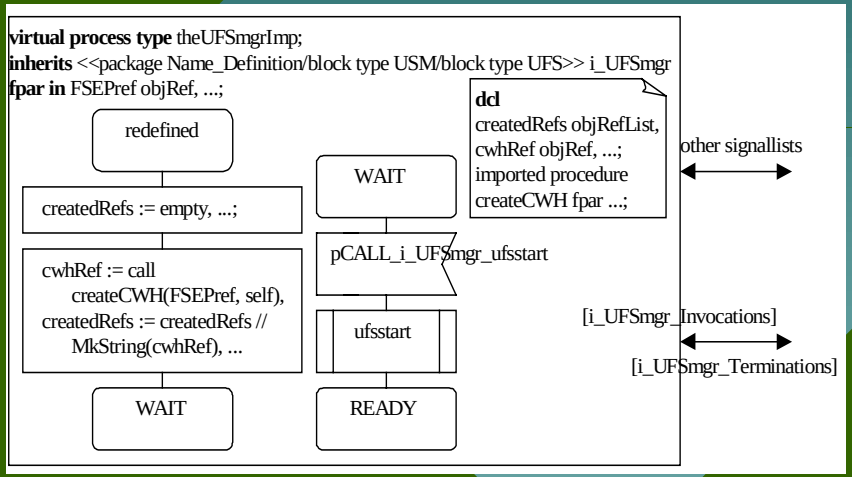
ODL to SDL mapping

ODL Structure	SDL Mapping
Group type	Block type
Object type	Block type
Interface type	Process type
Object Reference	Pid
Oneway (asynchronous) Operation	Signal prefixed with pCALL_
Operation (synchronous)	Signal pair. The first signal is prefixed with pCALL_, the second signal prefixed with pREPLY_ or pRAISE_ (if exception raised)
Exception	Signal prefixed with pRAISE_
Basic IDL types, e.g. long, char, float,…	Syntype
Enum	Newtype with corresponding literals
Typedef	Syntype
Struct	Newtype with corresponding structure
Constant	Synonym

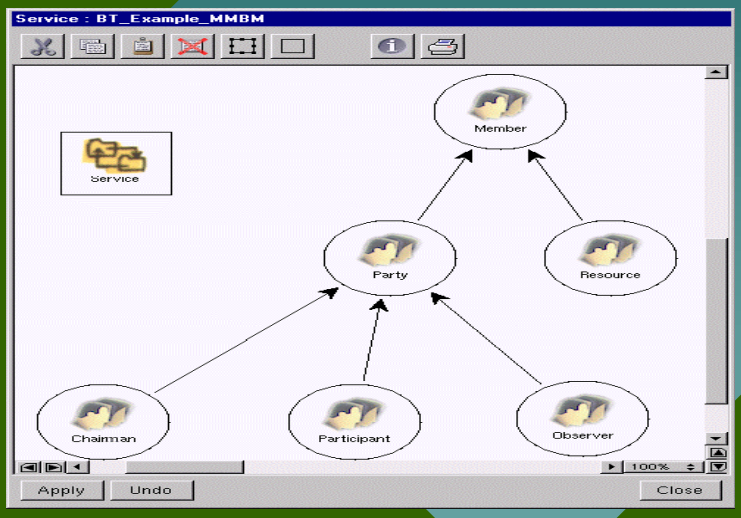
Example of the mapping

Package Name_Interface
signal pCALL_i_UFSmgr_suspendSessionRequest;
signal pCALL_i_UFSmgr_terminateSessionRequest;
signal pCALL_i_UFSmgr_ufsstart;
signal pCALL_i_UFSmgr_requestObject(NamedObject);
// and associated pREPLY_ signals – but not for ufsstart (oneway)
signallist i_UFSmgr_INVOCATIONS =
pCALL_i_UFSmgr_suspendSessionRequest,
pCALL_i_UFSmgr_terminateSessionRequest,
pCALL_i_UFSmgr_ufsstart, pCALL_i_UFSmgr_requestObject..;
signallist i_UFSmgr_TERMINATIONS =
pREPLY_i_UFSmgr_suspendSessionRequest,
pREPLY_i_UFSmgr_terminateSessionRequest,
pREPLY_i_UFSmgr_requestObject... ;

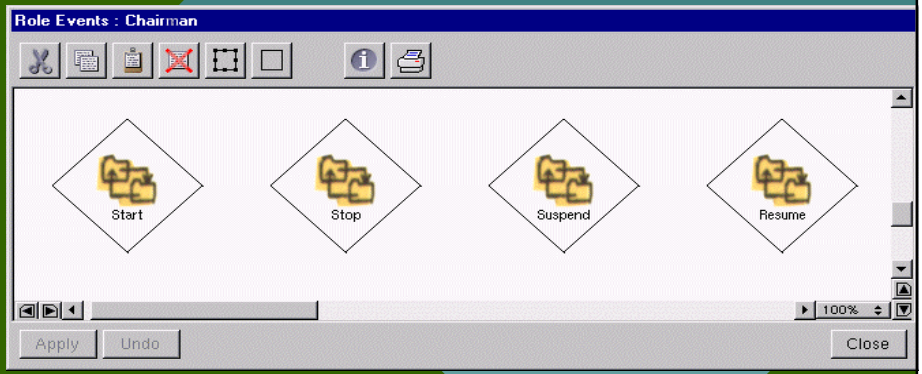
SDL framework



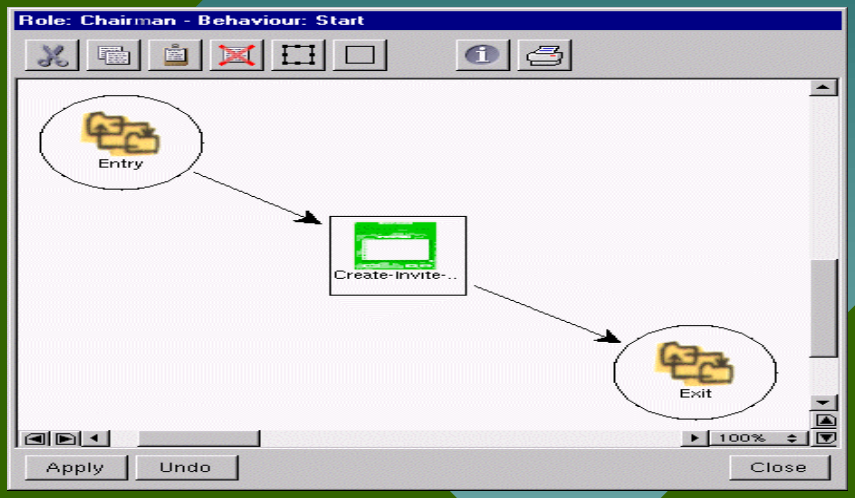
Functional Block Paradigm Usage



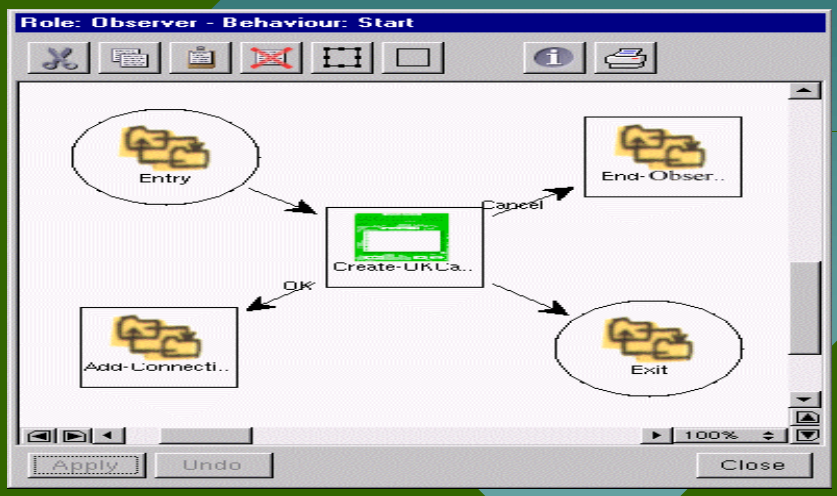
Functional Block Paradigm Usage contd...



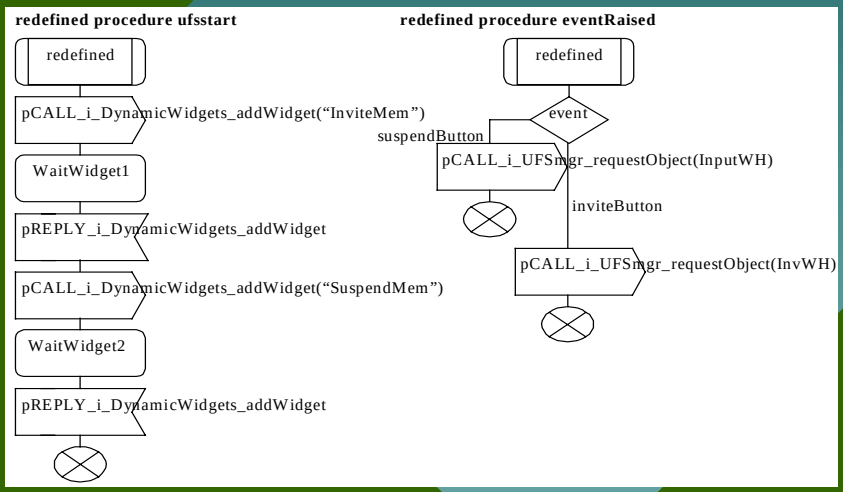
Functional Block Paradigm Usage contd...



Functional Block Paradigm Usage contd...

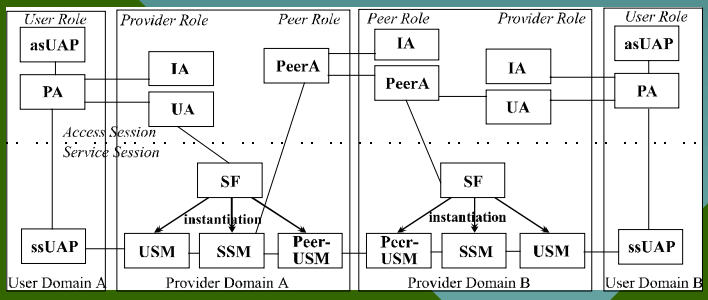


Example of Specialising SDL

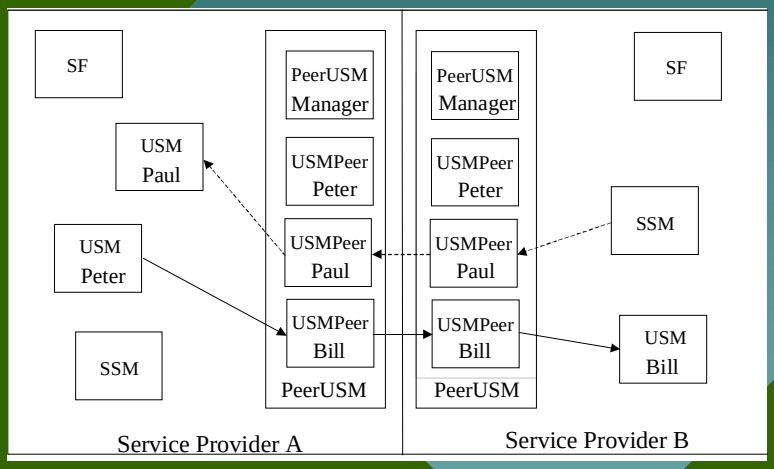


Federation

- domain federation
  - provides environment for two or more providers to transparently offer service across their domains by setting up access session
  - is basis for and governs:
- service federation
  - establishes service session across two or more retailer domains



Basic Design of PeerUSMs



Experienced Interactions

Issue related to federation of same services:

- clash between the role specified in the invitation and the role assigned to the initial user (Board-Meeting, Videophone Service)

Issues related to federation of slightly different services:

- role cardinality
- implemented policies like:
  'When chairman goes, the session goes.'

Conclusions

Frameworks allow semantics to be captured AND re-used

Paradigm tools allow for rapid service creation

- no need to know framework specialisation details
- no need to know programming or specification languages or distributed environments to create services

Conclusions contd

SDL models of services allow for

- animation (look and feel of the services)
- more detailed exploration of service behaviour
- test case generation and execution against service

Service interworking causes service interactions

- federation contract: when are two services compatible?

http://www.teltec.dcu.ie/tosca
TOSCA
GMD
FOKUS