

DL Architectures

TEL – DELOS Meeting

Donatella Castelli
CNR-ISTI, Italy

Heiko Schuldt
University of Basel, Switzerland
UNIT, Austria

End-user requirements: some examples

“I would like to be able to search transparently across the large collections of documents held by European libraries” (scalability)

“I need a good response time for this functionality” (performance)

“The search language does not always satisfy my needs, I would like to replace it with a richer one” (easy update)

Libraries requirements: some examples

“We have spent a lot of effort for creating this repository; it has always been managed following our own established rules; our staff is used to these procedures. We are willing to share the content, but we do not want to change the way in which is managed” (independence)

“We got a large collection of audiovisual material, our server has not enough storage capabilities. We would like to store this collection elsewhere.” (distribution)

“We have just digitised and make it accessible on-line a very important collection. We need a system able to support a much larger number of users than before.” (scalability)

Developer requirements: some examples

“I must deliver the system as soon as possible, I would like to be able to add the functionality that will serve the needs of the users in 2010 incrementally” (extensibility)

“I would like to have a mechanism for ‘Lower the barrier for users to remix the output of services’” (composability)

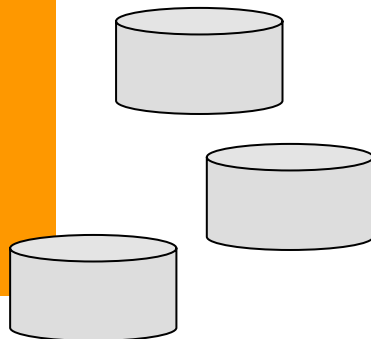
The answer to all these requirements

1. An appropriate software architecture
2. An appropriate middleware

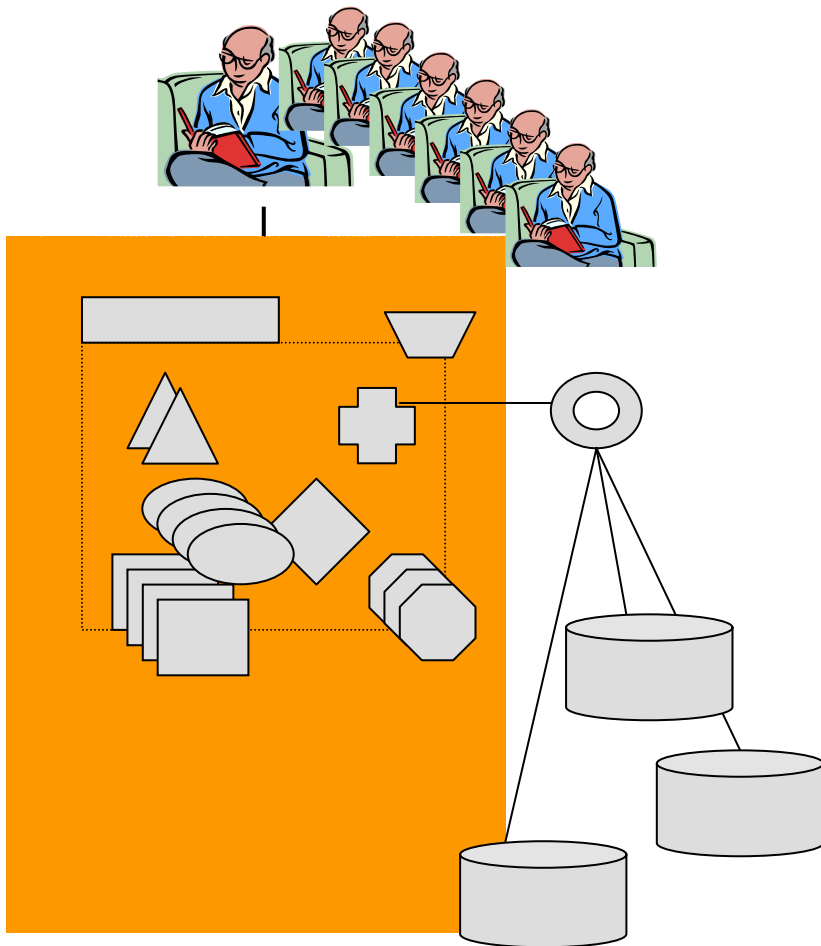
Monolithic Systems vs. Service-Orientation



- Many current DL systems have evolved over time, leading to a monolithic system architecture
- Problems
 - Search over different DLs/collections?
 - Add new functionality?
 - Scalability: support large number of users/documents?



Monolithic Systems vs. Service-Oriented

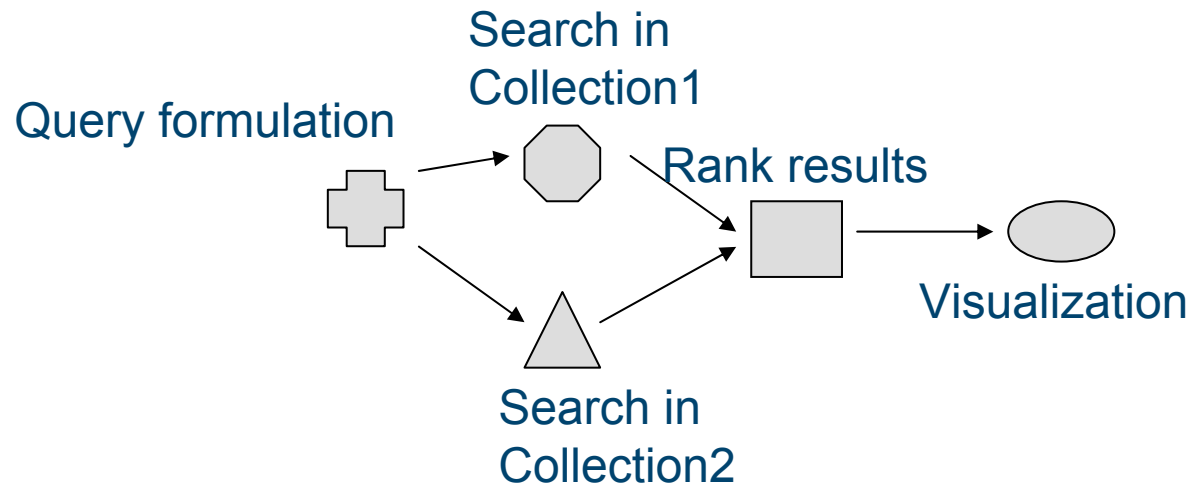


- Provide functionality by means of independent services
 - Search over different DLs/collections
 - Add new functionality
 - Scalability: support large number of users/documents (replication of services or content)
 - Distribution

Service Composition



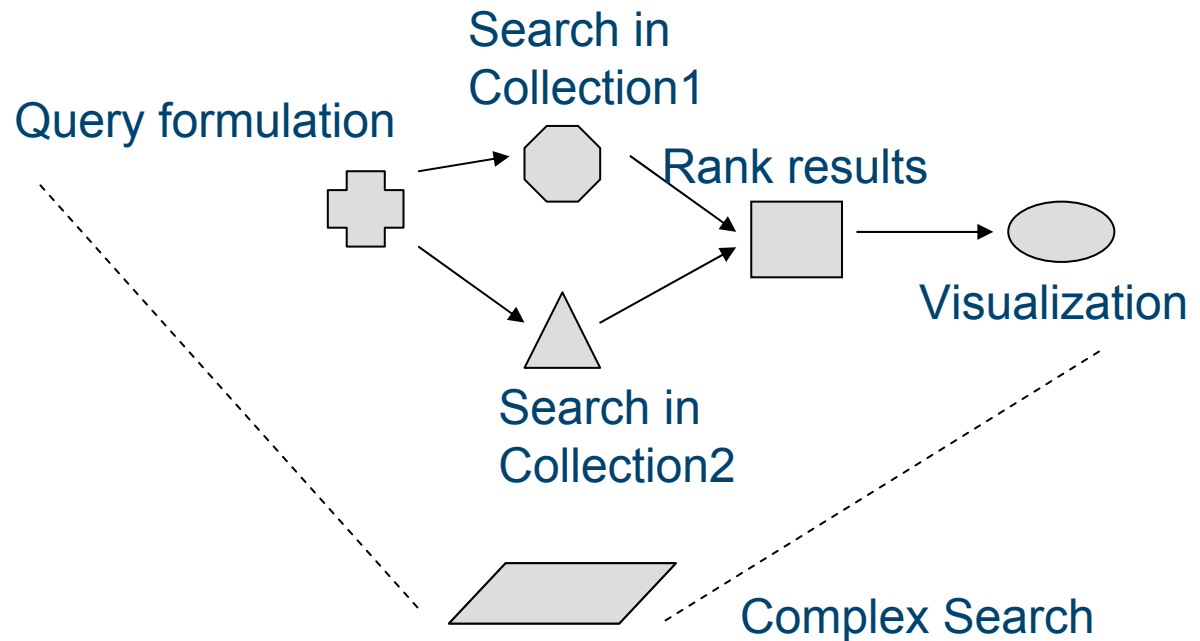
- Existing services can be combined to new services



Service Composition



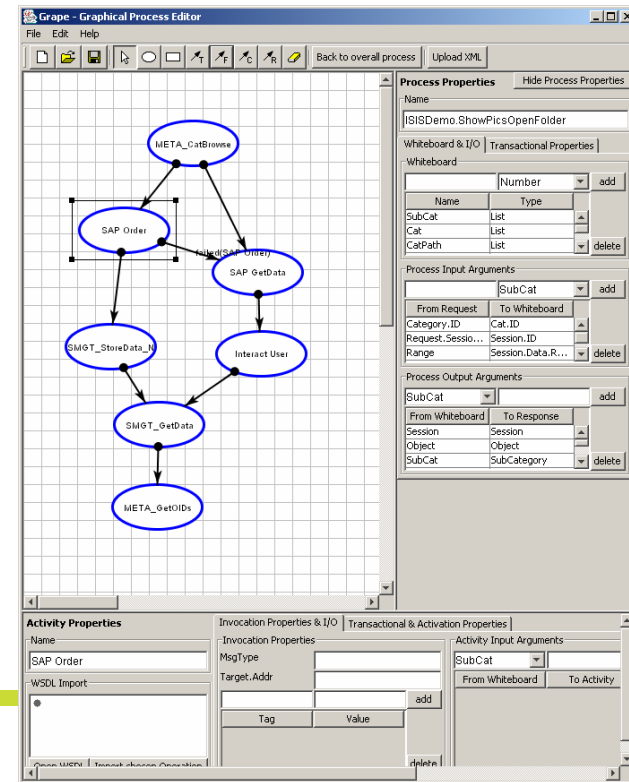
- A composite service is made available again as service



Service Composition



- Users should not have to deal with service composition
- Basic DL functionality: workflow/process support
- If needed: DL user to define her/his own new services

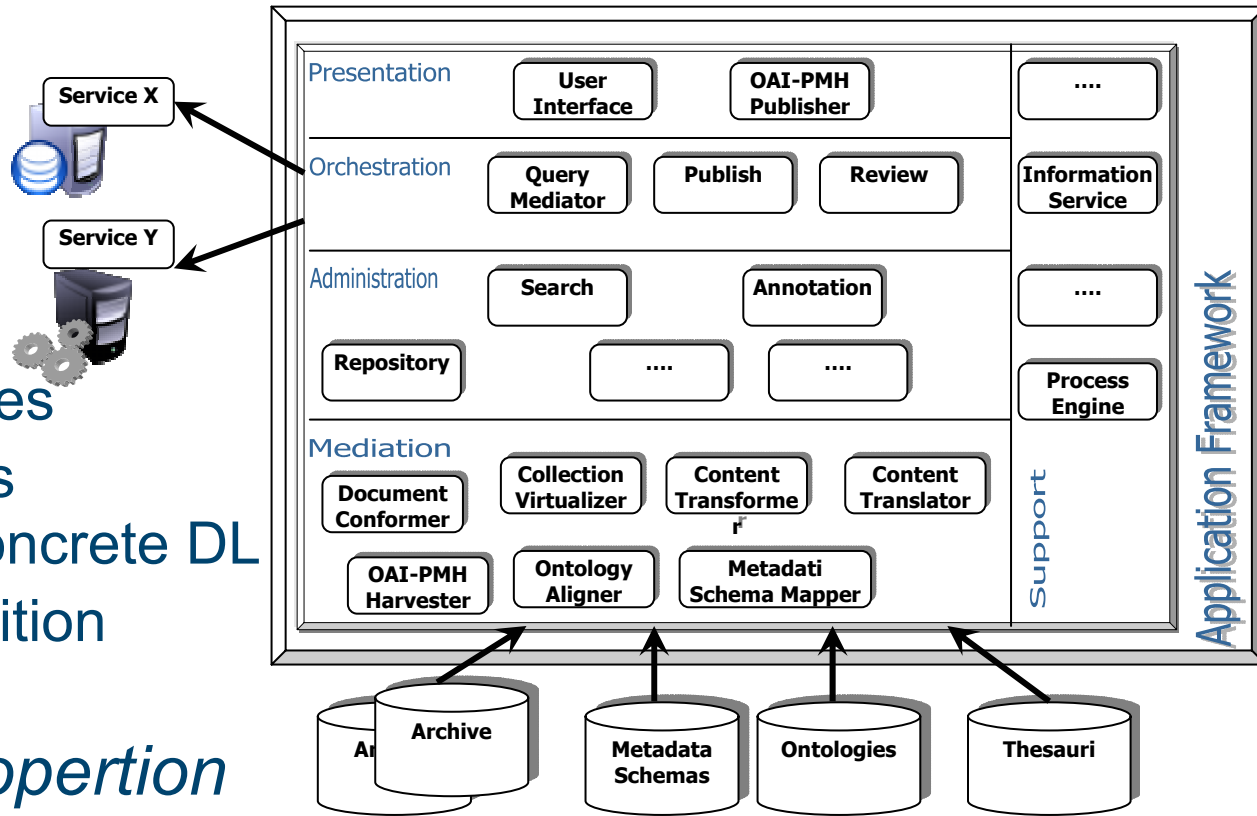


DELOS Contribution

- A Reference Model for Digital Libraries

- Identification of functionality at different levels

- Realized by means of services
- Choose services needed for a concrete DL
- Service composition
- ...



- *TEL-DELOS cooperation for the validation/refinement of the Reference Model*

DELOS Contribution

- Prototype development
 - Integration of DL services provided by DELOS partners
 - *Integration of TEL services?*