Conference Management System
A Case Study in SODA

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Ambra Molesini

{andrea.omicini, ambra.molesini}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008

# Distributed Paper Review [Zambonelli et al., 2003, Mathews and Jacobs, 1996, Ciancarini et al., 1996]

- The *Program Chair* (PC-Chair) publishes a call for papers
- Authors submit papers
- A number of scientists (called *PC Members*) review the papers and give marks
- To ensure fairness, the reviewers must be anonymous, expert, and must be willing to do the review
- Also, each paper should receive a minimum number of reviews from different scientists
- All accepted papers will be published on a book

Small Conference
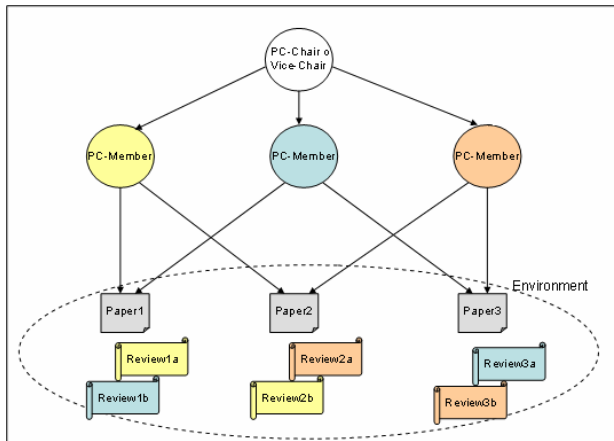
Big Conference
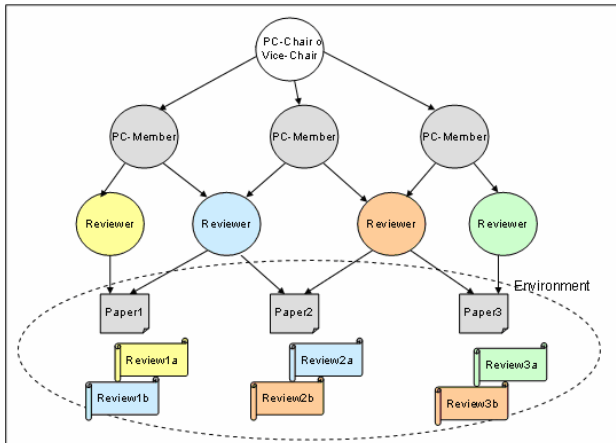
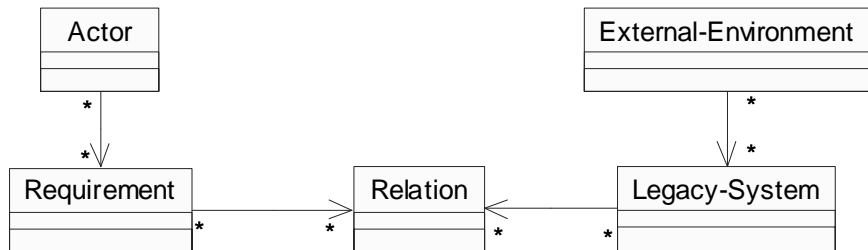# A case study: why Agents?

- It is a typical case of distributed workflow management
  - There are actions to do on common documents
  - According to specific rules
- Each of the human actors involved in the process
  - Could be supported by a personal agents
  - Helping him to submit documents, filling in, respect deadlines, etc.

Requirement Analysis Meta-model

Requirement Analysis: Actors and Requirements

- $(C)AR_t$:

Actor	Requirement
Organisation	ManageStartUp
	ManageSubmission
	ManagePartitioning
	ManageAssignment
	ManageReview

Requirement Analysis: Requirements

- $(C)Re_t$:

Requirement	Description
ManageStartUp	creating call for paper and defining the rules of the organisation
ManageSubmission	managing user registration and paper submissions
ManagePartitioning	partitioning of papers based of the structure of conference
ManageAssignment	managing the assignment process according to the rules of the organisation
ManageReview	managing the review process and sending reviews to authors

Requirement Analysis: External Environment and Legacy-System

- $(C)EELS_t$:

External-Environment	Legacy-System
External	WebServer

- $(C)LS_t$:

Legacy-System	Description
WebServer	it is the container for the web application of the conference

Requirement Analysis: Relations

- $(C)Rel_t$:

Relation	Description
Web	access to the web in order to retrieve or storage some information

- $(C)RLS_t$:

Legacy-System	Relation
WebServer	Web

Requirement Analysis: Relations

- $(C)RR_t$:

Requirement	Relation
ManageStartUp	Web
ManageSubmission	Web
ManagePartitioning	Web
ManageAssignment	Web
ManageReview	Web

Zooming

Now we in-zoom the ManagePartitioning requirement

- $(C)Z_t$

Layer C	Layer C+1
ManagePartitioning	UpdateStartUp
	ManageSubCommetee
	ManageClassification
	PartitionPapers

Requirement Analysis: Requirements

- $(C+1)Re_t$:

Requirement	Description
UpdateStartUp	It could be necessary to update the structure and the rules of the organisation in order to manage the great number of paper submitted
ManageSubCommetee	If it is necessary the sub-commettes will be created
ManageClassification	classification of the papers in base ok key words suggested by authors
PartitionPapers	partitioning of papers in order to accomplish at the organisation's rules

Requirement Analysis: Relations

- $(C + 1)RR_t$:

Requirement	Relation
UpdateStartUp	+ Web
ManageSubCommittee	+ Web
ManageClassification	+ Web
PartitionPapers	+ Web

Analysis Meta-model
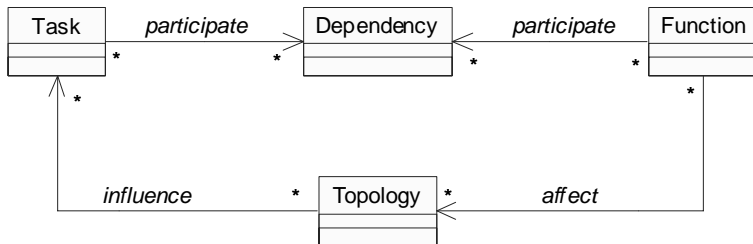
From Requirement Analysis to Analysis

- $(C)RRT_t$:

Requirement	Task
ManageStartUp	start up
ManageSubmission	submission
ManagePartitioning	paper partitioning
ManageAssignment	assignment papers
ManageReview	review process

From Requirement Analysis to Analysis

- $(C + 1)RRT_t$:

Requirement	Task
UpdateStartUp	modifying startup
ManageSubCommetee	create sub-commettees
	Vice-Chair elections
ManageClassification	papers classification
PartitionPapers	partition papers

From Requirement Analysis to Analysis

- $(C)RRF_t$:

Requirement	Function
ManageStartUp	management process
	management user
ManageSubmission	management user
	management paper
ManagePartitioning	management partitioning
	management paper
ManageAssignment	management assignment
	management paper
ManageReview	management review
	management paper

From Requirement Analysis to Analysis

- $(C)RLSF_t$:

Legacy-System	Function
WebServer	webSite

- $(C)RLST_t$:

Legacy-System	Topology

- $(C)RRD_t$:

Relation	Dependency
Web	webAccess

Analysis: Tasks

- $(C) T_t$

Task	Description
start up	insertion of the setup information
submission	submission of paper
paper partitioning	partitioning of the set of papers
assignment papers	assignment papers to PC-members
review process	creation and submission of the reviews

Analysis: Zoom

- Zooming Table: $(C)Z_t$

Layer C	Layer C+1
paper partitioning	modifying startup
	create sub-commettees
	Vice-Chair elections
	papers classification
	partition papers
	new organisation
	classification
	partition
	election

Analysis: Tasks at C+1

- $(C + 1)RRT_t$:

Task	Description
modifying startup	update the structure and the rules of the organisation
create sub-commettees	creating of sub-commettes
Vice-Chair elections	for each sub-commette it is necessary to elect the Vice-Chair
papers classification	classification of papers in base of key words
partition papers	partitioning papers in base of their classification

Analysis: Functions

- $(C)F_t$

Function	Description
management user	managing users'information
management review	managing reviews'information
management paper	managing papers'information
management assignment	managing assignments'information
management partitioning	managing partitioning's information
management process	managing start-up's information
webSite	web interface of the conference

Analysis: Dependencies

- $(C)D_t$

Dependency	Description
start up information	access of all the information about start up process
user information	access to all the users' information
paper information	access to all the papers' information
partitioning information	access to all the information about partitioning process
submission information	access to all the information about submission process
assignment information	access to all the information about assignment process. A reviewer cannot be the author of the papers that are assigned to him
review information	access to all the information about review process
webAccess	access to the web site of the conference

Analysis: Dependencies at C+1

- $(C + 1)D_t$

Dependency	Description
new organisation	organisation is changed
election	start the election of vice-chairs
classification	it is necessary to start the classification of papers
partition	it is necessary to start the partitioning of papers

Analysis: Tasks and Dependencies

- $(C)TD_t$

Task	Dependency
start up	start up information
submission	submission information user information paper information webAccess
paper partitioning	start up information partitioning information paper information user information webAccess
assignment papers	assignment information paper information user information webAccess
review process	review information paper information webAccess

Analysis: Tasks and Dependencies at C+1

- $(C+1)TD_t$:

Task	Dependency
modify startup	+start up information +webAccess new organisation classification
create sub-commetees	+ user information +webAccess election
Vice-Chairs elections	+user information +webAccess election classification
paper classification	classification + paper information +webAccess partition
partition paper	partition + paper information +webAccess + partitioning information

Analysis: Function and Dependencies

- $(C)FD_t$

Function	Dependency
management user	user information submission information assignment information
management review	review information
management paper	paper information submission information assignment information review information partitioning information
management assignment	assignment information
management partitioning	partitioning information
management process	start up information
webSite	webAccess

Analysis: Topologies

- $(C)Top_t$

Topology	Description
place	this is the locus where the functions are allocated

- $(C)TTop_t$

Task	Topology
start up	place
submission	place
paper partitioning	place
assignment papers	place
review process	place

Analysis: Topologies and Functions

- $(C)FTop_t$

Function	Topology
management user	place
management review	place
management paper	place
management assignment	place
management partitioning	place
management process	place
webSite	place

Analysis: Topologies at C+1

- $(C + 1)\,TTop_t$

Task	Topology
modify start up	+place
create sub-commettes	+place
Vice-Chair elections	+place
paper classification	+place
partition paper	+place

Architectural Design Meta-model
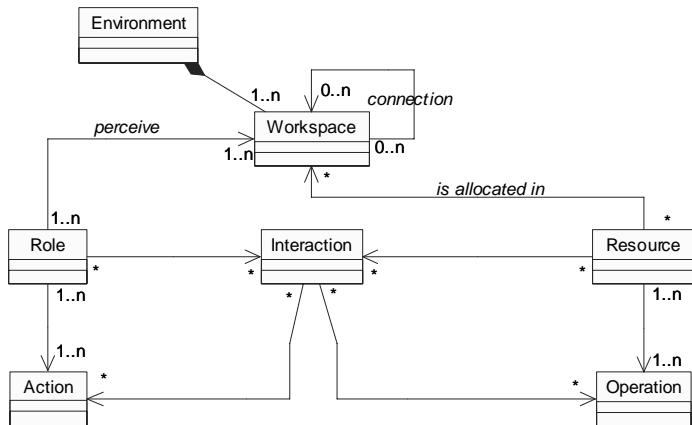
From Analysis to Architectural Design

- $(C) TRT_t$:

Role	Task
PC-Chair	paper partitioning
	start-up
	assignment papers
Author	submission
PC-member	review process

From Analysis to Architectural Design

- $(C)\, TRF_t$:

Resource	Function
People DB	management user
Paper DB	management paper
	management review
	management partitioning
	management assignment
Process DB	management process
WebService	webSite

From Analysis to Architectural Design at C+1

- $(C + 1)TRT_t$:

Role	Task
ManagerStartUp	modify start up
Sub-Commette	create sub-commettees
	Vice-chair elections
Partitioner	papers classification
	partition papers

From Analysis to Architectural Design

- $(C)\,TID_t$

Dependency	Interaction
start up information	
user information	User-Rule
paper information	Author-Rule
partitioning information	Match-Rule
submission information	Deadline-Rule
assignment information	AutRev-Rule
	Review-Rule
review information	Author-Rule
webAccess	Access-Rule

From Analysis to Architectural Design at C+1

- $(C+1)\,TID_t$

Dependency	Interaction
new organisation	Org-Rule
election	Vice-Rule
classification	Class-Rule
partition	Part-Rule

- $(C)\,TTopW_t$

Topology	Workspace
place	Wplace

Architectural Design: zoom

- Zooming Table: $(C)Z_t$

Layer C	Layer C+1
PC-Chair	ManagerStartUp, Sub-Commettee
	Partitioner, ...
	Vice-Rule, Org-Rule
	Part-Rule, Class-Rule
	change information
	read paper information
	modify paper information
	define vice-chair

Architectural Design: Actions

- $(C)A_t$

Action	Description
login	user authentication
send paper	user compiles form and sends his paper
publish deadline	user generates/modifies deadline
partition	user splits papers according to key words
assignment	user assigns papers
read paper	user reads papers
download paper	user download paper from the web
write review	user writes the review
. . .	. . .

Architectural Design: Actions at C+1

- $(C+1)A_t$

Action	Description
change information	changing an information in start up process
read paper information	reading information about a paper
modify paper information	modifying an information about a paper
define vice-chair	election of Vice-Chair
...	...

Architectural Design: Actions and Roles

- $(C)RA_t$

Role	Action
PC-Chair	login
	publish deadline
	partition
	assignment
	...
Author	login
	send paper
	...
PC-member	login
	read paper
	write review
	download paper
	...

Architectural Design: Actions and Roles at C+1

- $(C+1)RA_t$

Role	Action
ManagerStartUp	+ login change information . . .
SubCommette	+ login define vice-chair . . .
Partitioner	+login + partitions read paper information modify paper information . . .

Architectural Design: Operations

- $(C)O_t$

Operation	Description
store paper	storing paper and its information
get paper	providing paper and its information
store user	storing user information
get user	providing user information
store process	storing process information
get process	providing process information
store assignment	storing assignment information
access web	friendly interface of the application
. . .	. . .

Architectural Design: Operations and Resources

- $(C)RO_t$

Resource	Operation
People DB	store user
	get user
	. . .
Paper DB	store paper
	get paper
	store assignment
	. . .
Process DB	get process
	store process
	. . .
WebService	access Web

Architectural Design: Interactions

- $(C)I_t$

Interaction	Description
Deadline-Rule	send paper is possible if and only if time is minus then deadline submission
User-Rule	get user is possible if the request user is the requester or the requester is the PC-Chair
Author-Rule	author can access and modify only his public paper information
Match-Rule	papers can be partitioned according key words
AutRev-Rule	the PC-member cannot be the author of paper
Review-Rule	the PC-member cannot access to private information about his papers
Access-Rule	the access to the system must be authorised
. . .	. . .

Architectural Design: Interactions at C+1

- $(C + 1)I_t$

Interaction	Description
Org-Rule	if the organisation is changed then start Sub-Commette
Vice-Rule	the Vice-Chair must be an expert of the filed
Class-Rule	a paper can belong only at one class
Part-Rule	papers can be partitioned according their classification

Architectural Design: Interactions and Roles

- $(C)Rol_t$

Role	Interaction
PC-Chair	Deadline-Rule
	User-Rule
	Author-Rule
	Match-Rule
	AutRev-Rule
	Review-Rule
	Access-Rule
	. . .
Author	Deadline-Rule
	User-Rule
	Author-Rule
	Access-Rule
	. . .
PC-member	User-Rule
	Author-Rule
	AutRev-Rule
	Review-Rule
	Access-Rule
	. . .

Architectural Design: Interactions and Roles at C+1

- $(C + 1)Rol_t$

Role	Interaction
ManagerStartUp	+Deadline-Rule
	+User-Rule
	+Access-Rule
	Org-Rule
	. . .
SubCommette	+User-Rule
	+Access-Rule
	Vice-Rule
	. . .
Partitioner	+User-Rule
	+Access-Rule
	+Match-Rule
	Class-Rule
	Part-Rule
	. . .

Architectural Design: Interactions and Resources

- $(C)Rel_t$

Resource	Interaction
People DB	User-Rule
	Match-Rule
	. . .
Paper DB	Author-Rule
	AutRev-Rule
	Review-Rule
	. . .
Process DB	Deadline-Rule
	. . .
WebService	

Architectural Design: Workspaces

- $(C)W_t$:

Workspace	Description
Wplace	this is the workspace where the resources are be allocated

- $(C)WRe_t$:

Workspace	Resource
Wplace	People DB
	Paper DB
	Process DB
	WebService

Architectural Design: Workspaces and Roles

- $(C)WC_t$:

Workspace	Connection
Wplace	

- $(C)WRo_t$:

Role	Workspace
PC-Chair	Wplace
Author	Wplace
PC-member	Wplace

Architectural Design: Workspaces and Roles at C+1

- $(C + 1)WRo_t$:

Role	Workspace
ManagerStartUp	+Wplace
Sub-Commette	+Wplace
Partitioner	+Wplace

Detailed Design Phase Meta-model
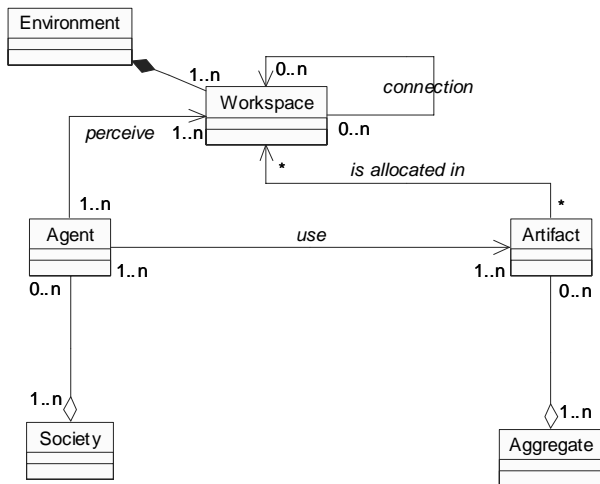
Design Views

- Potentially, in this step our system could be composed by all the layers detected in the previously steps.
- However, the deliverable of the Detailed Design step will be composed of only one layer
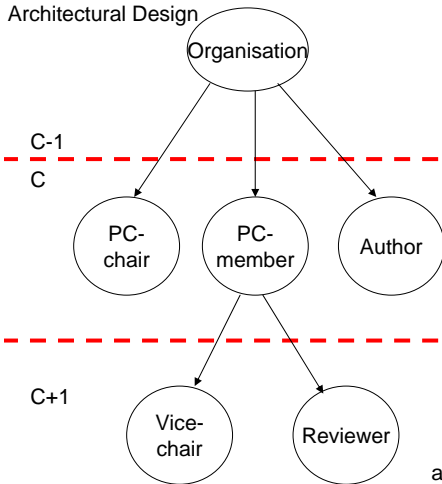- So, for each entity, we choose the appropriate layer of representation:
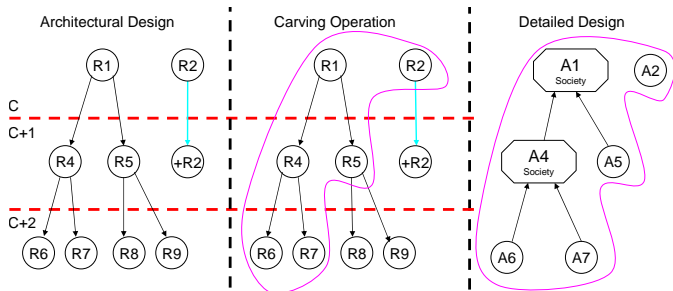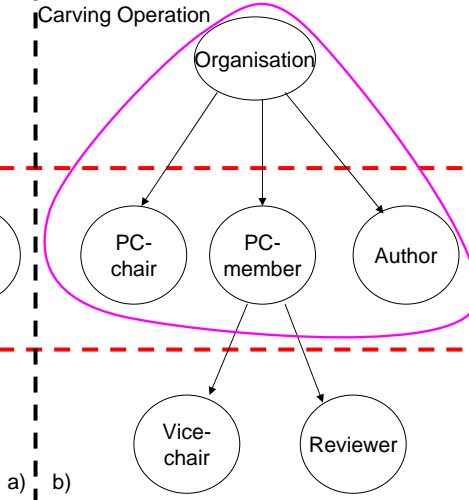
Design views in the case study

- This presentation shows only a sketch version of the case study because the complete version requires a lot of times for a detailed explanation.
- So, for sake of simplicity, in this step of methodology I choose the level C for each entities.

Architectural Design

Carving Operation

Organisation

C-1
C

PC-chair

PC-member

Author

C+1

Vice-chair

Reviewer

Organisation

PC-chair

PC-member

Author

Vice-chair

Reviewer

a) | b)

From Architectural Design to Detailed Design

- $(C)MAR_t$:

Agent	Role
PC-Chair Agent	PC-Chair
Author Agent	Author
PC-Member Agent	PC-Member

- $(C)MArR_t$:

(Environmental) Artifact	Resource
Paper Artifact	Paper DB
People Artifact	People DB
Process Artifact	Process DB
Web Artifact	WebService

From Architectural Design to Detailed Design

- $(C)MArI_t$:

Interaction	(Social) Artifact
Deadline-Rule	StartUp Artifact
User-Rule	User-Rule Artifact
Access-Rule	User-Rule Artifact
Author-Rule	Partition Artifact
Match-Rule	Partition Artifact
AutRev-Rule	Assignment Artifact
Review-Rule	Review Artifact
. . .	. . .

Detailed Design: Agents and Societies

- $(C)AA_t$:

Agent	(Individual) Artifact
PC-Chair Agent	PC-Chair Artifact
Author Agent	Author Artifact
PC-Member Agent	PC-Member Artifact

- $(C)SA_t$:

Society	Agent
Org	PC-Chair Agent
	Author Agent
	PC-Member Agent

- $(L)SAr_t$:

Society	(Social)Artifact
Org	StartUp Artifact
	User-Rule Artifact
	Partitioning Artifact
	Assignment Artifact
	Review Artifact

Detailed Design: Environments

- $(C)AUI_t$:

Artifact	Usage Interface
PC-Chair Artifact	read start up information
	modify start up information
	login
	partition
	assignment
	. . .
Author Artifact	login
	send paper
	. . .
PC-Member Artifact	login
	read paper
	write review
	download paper
	. . .
People Artifact	store user
	get user
	. . .

Detailed Design: Environments

- $(C)AUI_t$:

Artifact	Usage Interface
Paper Artifact	store paper
	get paper
	store assignment
	. . .
Process Artifact	get process
	store process
	. . .
Web Artifact	access Web
	. . .
StartUp Artifact	deadline extension
	update rule
	read rule
	. . .

Detailed Design: Environments

- $(C)AUI_t$:

Artifact	Usage Interface
User-Rule Artifact	get user modify user . . .
Partition Artifact	partition paper access classification . . .
Assignment Artifact	check authors check reviewer . . .
Review Artifact	check access to review information . . .

Detailed Design: Environments

- $(C)AggA_t$:

Aggregate	Artifact

- $(C)WA_t$:

Workspace	Artifact
Wplace	PC-Chair Artifact, Author Artifact
	PC-Member Artifact, People Artifact
	Process Artifact, Web Artifact
	StartUp Artifact, User-Rule Artifact
	Partition Artifact, Assignment Artifact
	Review Artifact, Paper Artifact

Bibliography I

📑 Ciancarini, P., Nierstrasz, O., and Tolksdorf, R. (1996).
A case study in coordination: Conference management on Internet.
`ftp://ftp.cs.unibo.it/pub/cianca/coordina.ps.gz`.

📑 Mathews, G. J. and Jacobs, B. E. (1996).
Electronic management of the peer review process.
In *5th Intern. WWW conference on Computer networks and ISDN systems*, pages 1523–1538. Elsevier Science Publishers.

📑 Zambonelli, F., Jennings, N. R., and Wooldridge, M. J. (2003).
Developing multiagent systems: The Gaia methodology.
ACM Transactions on Software Engineering and Methodology (TOSEM), 12(3):317–370.

Conference Management System
A Case Study in SODA

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Ambra Molesini

{andrea.omicini, ambra.molesini}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008