

Analysis and Design of Large Scale Software II (SENG 401) Concept Mapping Application

<names>, <date>

SENG 401, <Group#>, Winter 2008

Table of Contents

Executive Summary.....	3
1 Conceptual View	3
1.1 Global Analysis	3
1.1.1 Factor Table.....	3
1.1.2 Issue Cards.....	3
1.2 Conceptual Configuration.....	3
1.2.1 Application Level	3
1.2.2 Supervisor Component	3
1.3 Global Evaluation	4
1.4 Resource Budgeting.....	4
2 Module View.....	4
2.1 Global Analysis	4
2.1.1 Factor Table (additional and changed factors).....	4
2.1.2 Issue Cards (additional and changed issue cards).....	4
2.1.2.1 <title>	5
2.2 Central Design Tasks.....	5
2.2.1 Mapping Conceptual Elements to Modules.....	5
2.2.2 Use Dependency Diagrams between Modules	5
2.2.3 Diagrams Assigning Modules to Layers.....	5
2.2.4 Use Dependency Diagram between Layers	6
2.2.5 Global Evaluation	6
2.3 Final Design Task: Interface Design	6
2.3.1 Interface Definition: Security/Approval	7
3 Execution View	7
3.1 Global Analysis	7
3.1.1 Factor Table (additional and changed factors).....	7
3.1.2 Issue Cards (additional and changed issue cards).....	7
3.1.2.1 <title>	7
3.2 Central Design Tasks.....	8
3.2.1 Platform Elements	8
3.2.2 Communication Paths.....	8
3.2.3 First Approximation	8
3.2.4 Refinement	9
3.2.5 Complete Execution Configuration	9
3.2.6 Global Evaluation	10
3.3 Final Design Task.....	10
3.3.1 Resource Allocation	10
3.3.2 Architecture Overview.....	10
4 Code View.....	11
4.1 Global Analysis	11
4.1.1 Factor Table (additional and changed factors).....	11
4.1.2 Issue Cards (additional and changed issue cards).....	11
4.1.2.1 <title>	11
4.2 Central Design Tasks.....	12

4.2.1	Source Components	12
4.2.2	Intermediate Components	12
4.2.3	Deployment Components	12
4.2.4	Relations	13
4.2.5	Global Evaluation	13
4.3	Final Design Tasks	13
4.3.1	Build Procedure	13
4.3.2	Configuration Management	14
A1:	Glossary	14
A2:	Summary of design decisions	14
A3:	<Any other relevant appendices...>	14
	References	14

Executive Summary

...

1 Conceptual View

...

1.1 Global Analysis

...

1.1.1 Factor Table

...

1.1.2 Issue Cards

.....

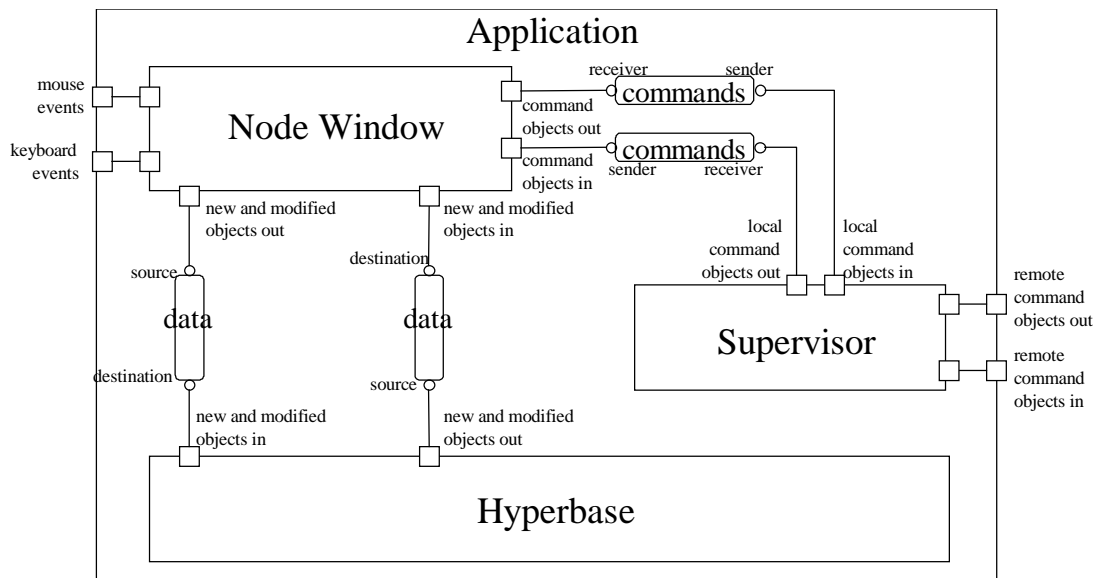
1.2 Conceptual Configuration

...

1.2.1 Application Level

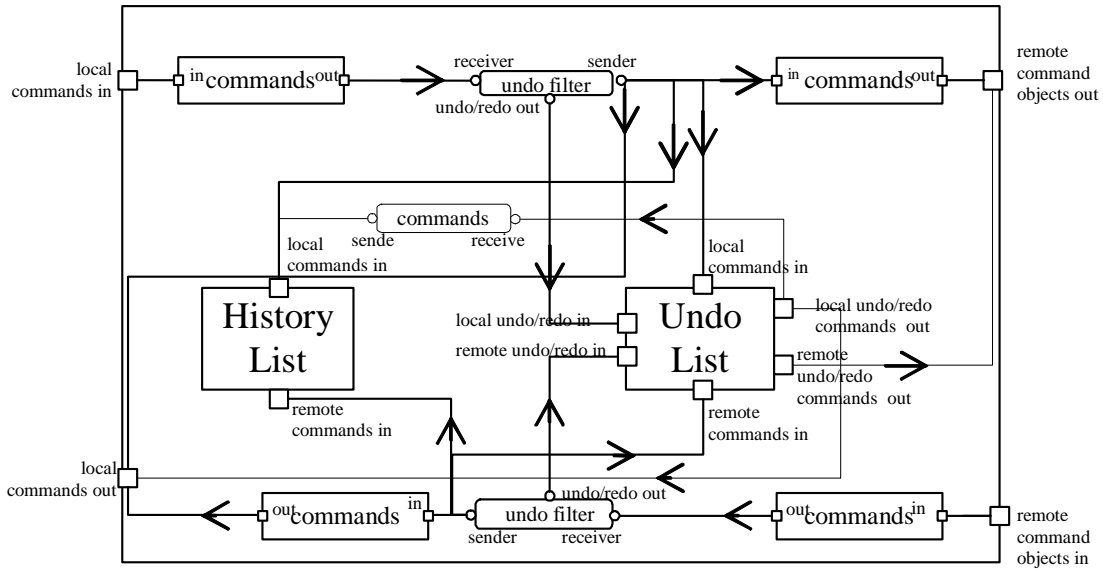
<provide a prose overview and be sure to explain all the components and connectors referencing any detailed definitions that you have included>

...



1.2.2 Supervisor Component

...



1.3 Global Evaluation

...

1.4 Resource Budgeting

...

2 Module View

...

2.1 Global Analysis

...

2.1.1 Factor Table (additional and changed factors)

...

Factor	Flexibility and Changeability	Impact

2.1.2 Issue Cards (additional and changed issue cards)

...

2.1.2.1 <title>
... Influencing Factors: ...
Solution: ...
Strategy: ...
Related strategies: ...

...

2.2 Central Design Tasks

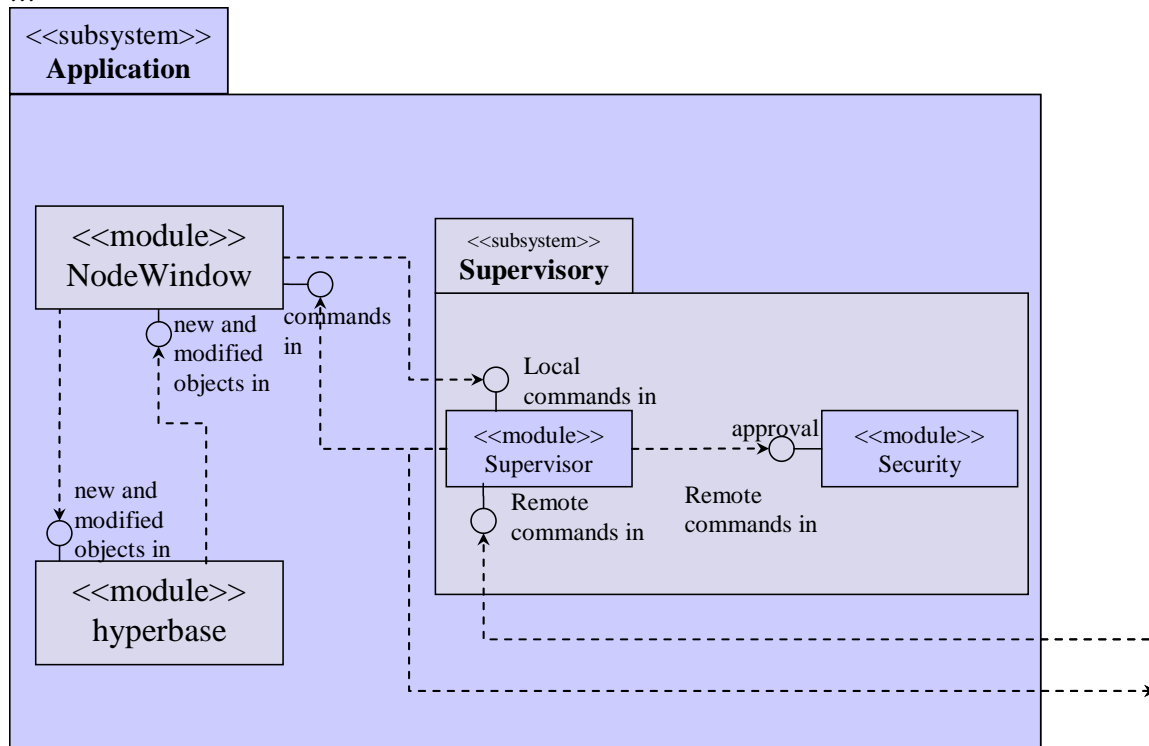
...

2.2.1 Mapping Conceptual Elements to Modules

...

2.2.2 Use Dependency Diagrams between Modules

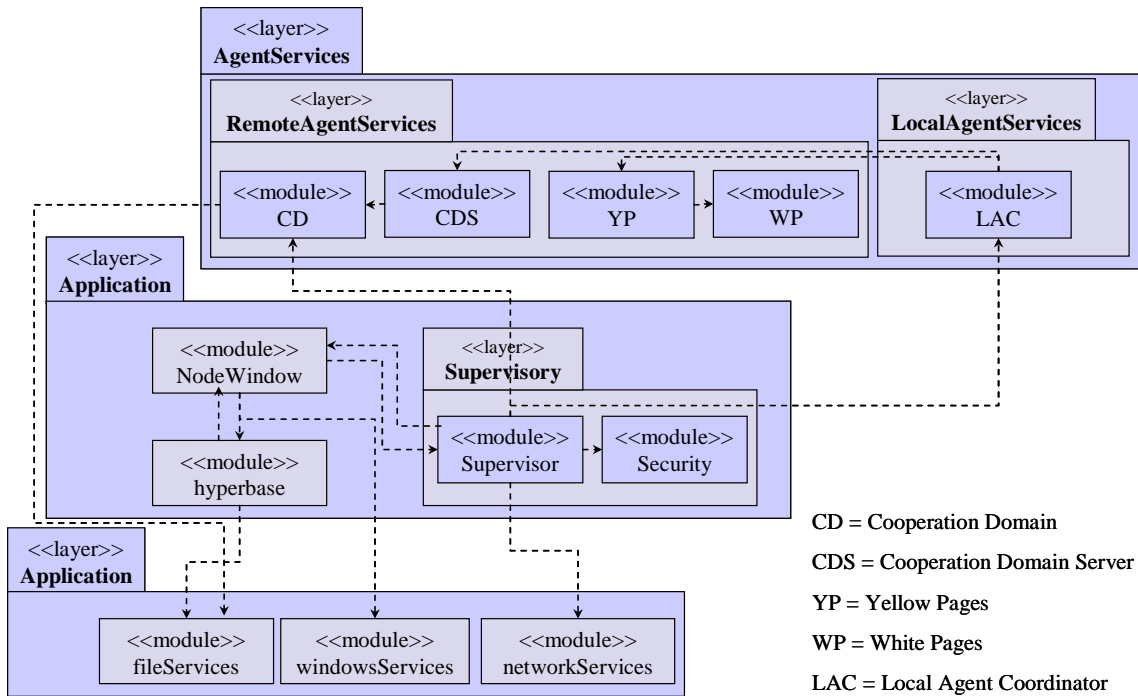
...



...

2.2.3 Diagrams Assigning Modules to Layers

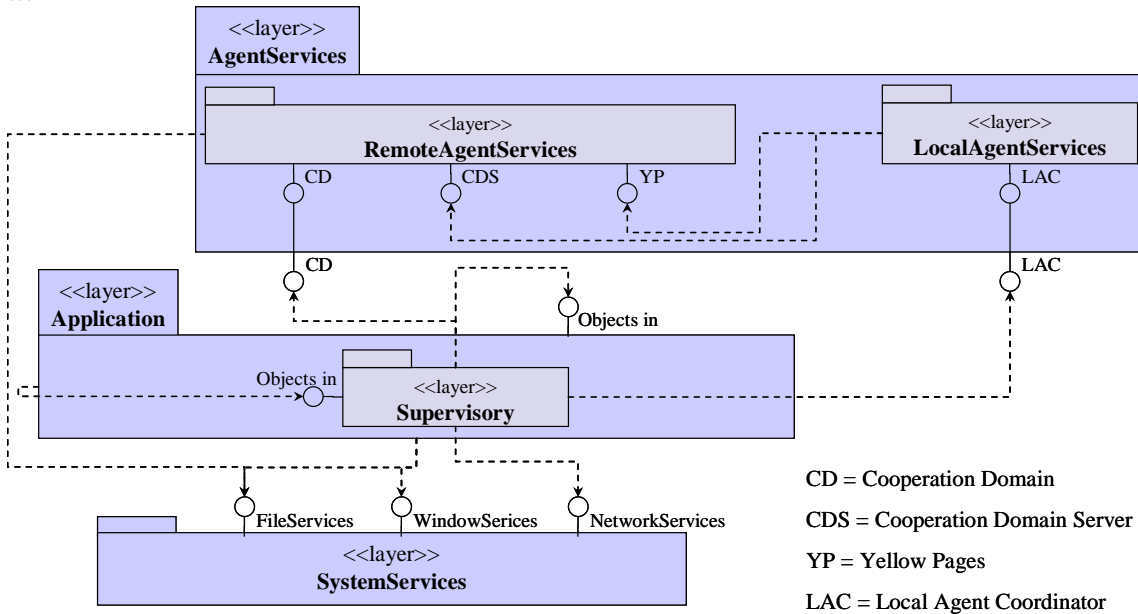
...



...

2.2.4 Use Dependency Diagram between Layers

...



...

2.2.5 Global Evaluation

...

2.3 Final Design Task: Interface Design

...

2.3.1 Interface Definition: Security/Approval

RegisterObject(id, pin, object, securitySpec) → status
 Registers securitySpec for object provided that id and pin allow it. Succeeds only if this is the first registration of object. id will become the owner of object.

ChangeObject(id, pin, object, securitySpec) → status
 Updates security with securitySpec for object provided that id and pin allow it.

DeleteObject(id, pin, object) → status
 Deletes object's security registration provided id and pin allow it

ApproveOp(id, pin, command) → status
 Returns OK iff id and pin are allowed to execute command.

status: {OK, denied, unknownObject, invalidSecuritySpec, invalidPin, wrongPin, unknownID}

...

3 Execution View

...

3.1 Global Analysis

...

3.1.1 Factor Table (additional and changed factors)

...

Factor	Flexibility and Changeability	Impact

3.1.2 Issue Cards (additional and changed issue cards)

...

3.1.2.1 <title>
... Influencing Factors: ...
Solution: ...
Strategy: ...

Related strategies:
...

...

3.2 Central Design Tasks

...

3.2.1 Platform Elements

...

Platform Element Type	Description
<<process>>	...
<<DLL>>	...
...	

...

3.2.2 Communication Paths

...

Communication Mechanism	Description
Socket	...
SC (System Call)	...
...	

...

3.2.3 First Approximation

...

Conceptual Component	Platform Element Type
Application	<<process>>
...	

...

3.2.4 Refinement

...

Module	Runtime Entity	Associated Attributes
Application	<<process>>	e.g., host type, replication, resource allocation (cf. Hofmeister et al., p 154)
...		

...

[Note: Host type = combination of CPU, vendor and OS that uniquely identifies a system (examples: i686-redhat-linux, i486-suse-linux, i686-pc-linux-gnu)]

3.2.5 Complete Execution Configuration

Figure 1 shows the modules contained in the runtime entities and the specific communication paths. ...

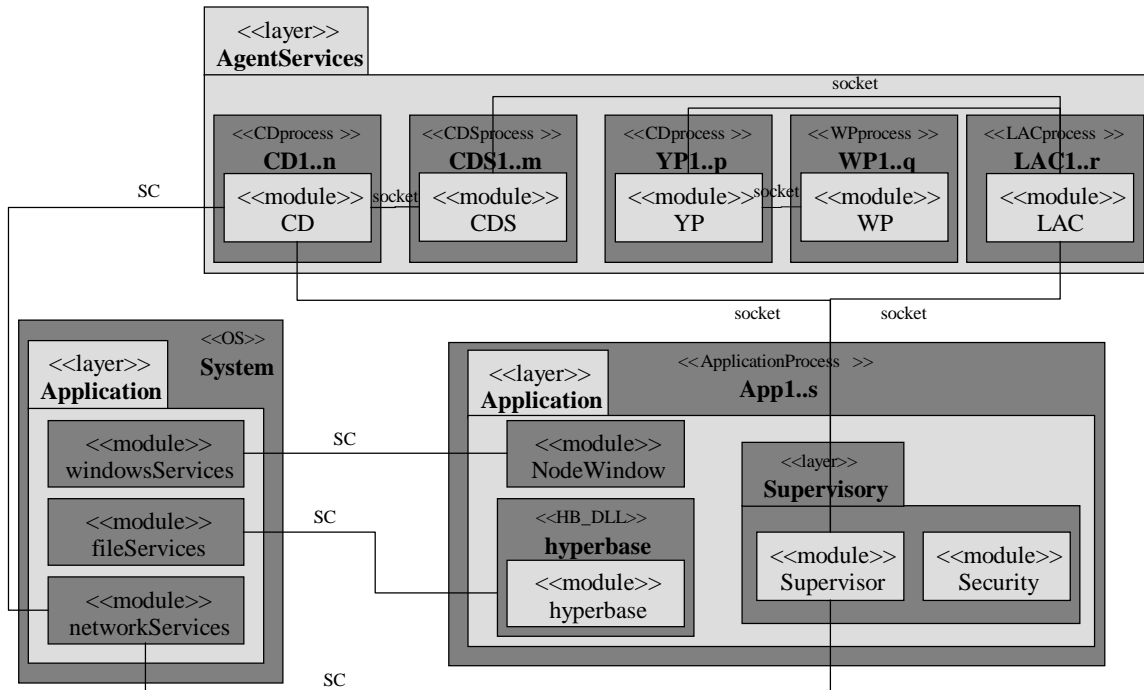


Figure 1: Detailed Execution Configuration Diagram

... Figure 2 shows an interaction diagram of the communication protocol that takes place over the socket communication between the LAC, CDS, and CD when a cooperation domain needs to be created. ...

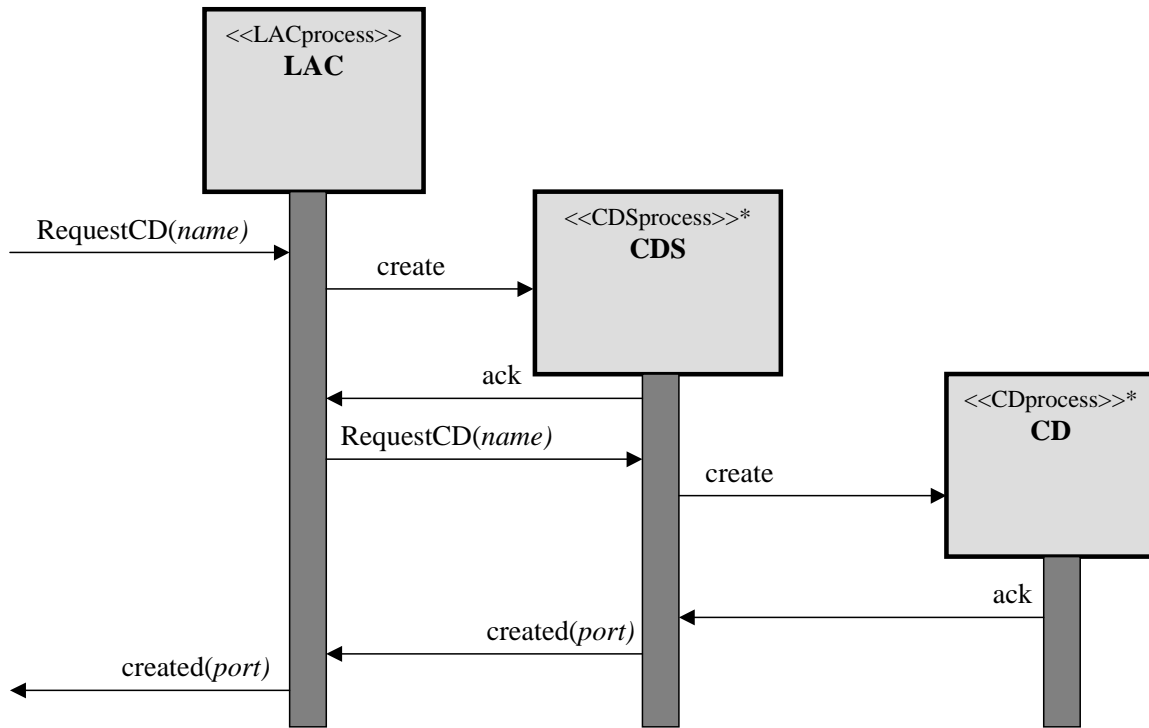


Figure 2: UML Sequence Chart (Interaction Diagram)

3.2.6 Global Evaluation

...

3.3 Final Design Task

...

3.3.1 Resource Allocation

...

(cf. Hofmeister et al., p. 153, Table 6.2 for examples)

3.3.2 Architecture Overview

....

Figure 3 describes the overall execution configuration for a single CPU that acts both as a "server" and as a "client" -- an "all-inclusive", "general purpose" machine. Many CPUs will not run all of these processes, but all will run an OS (of course). All machines that run ordinary agents (<<ApplicationProcess>>es) will run LACs, and all machines that run CDs will also run CDSs. There is exactly one LAC per CPU if there is a LAC, but there may be zero or more instances of all other processes.

The bold lines in Figure 3 indicate a process may communicate with the other process on another CPU. Unbolded communication lines indicate the communication is restricted to the same CPU.

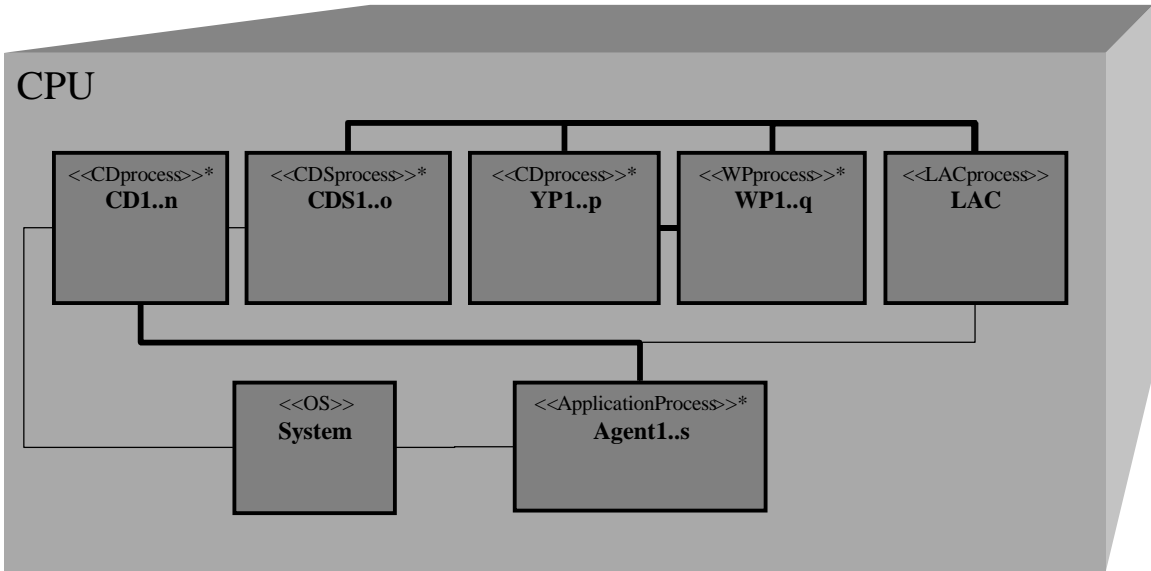


Figure 3: Overall Execution Configuration (Architecture Overview)

4 Code View

...

4.1 Global Analysis

...

4.1.1 Factor Table (additional and changed factors)

...

Factor	Flexibility and Changeability	Impact

4.1.2 Issue Cards (additional and changed issue cards)

...

4.1.2.1 <title>
<p>...</p> <p>Influencing Factors:</p> <p>...</p>

Solution: ... Strategy: ... Related strategies: ...

4.2 Central Design Tasks

4.2.1 Source Components

Source Component	Type	Sources (relation)
application.h	<<header>>	
application.cpp	<<source>>	application.h (import)
...		

4.2.2 Intermediate Components

Conceptual Component	Platform Element Type	Sources (relation)
superisor.obj	<<binary>>	supervisor.cpp (compile)
supervisor.lib	<<library>>	supervisor.obj (link) security.obj (link) ...
application.obj	<<binary>>	application.cpp (link) ...
...		

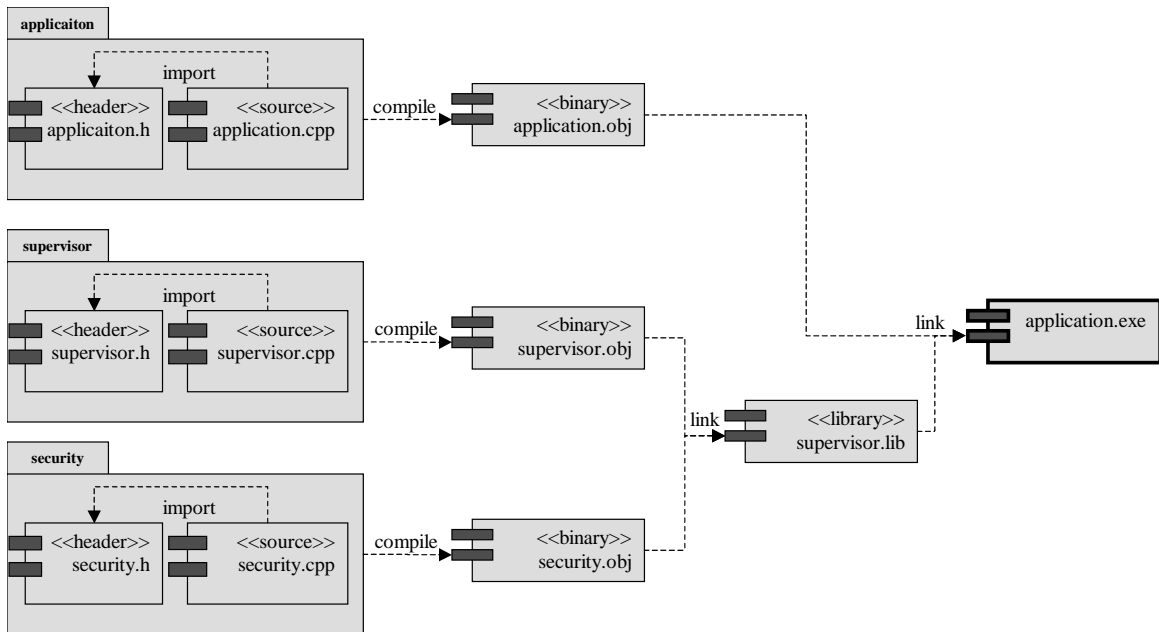
4.2.3 Deployment Components

Conceptual Component	Platform Element Type	Sources (relation)
application.exe	<<executable>>	application.obj (link) supervisor.lib (link) ...
...		

4.2.4 Relations

...

<Note: The diagram below is just an illustration to show you what kind of diagram is expected from you in this section. Please delete this diagram and include your own diagrams.>



4.2.5 Global Evaluation

...

4.3 Final Design Tasks

4.3.1 Build Procedure

...

<Note: The makefile-like build procedure below is just an illustration to show you what is expected from you in this section. Please delete this build procedure and include your own.>

```

application.exe: application.obj, supervisor.lib, ...
application.obj: application.cpp, applicaiton.h, ...
supervisor.lib: supervisor.obj, security.obj, ...
supervisory.obj: supervisor.cpp, supervisor.h, ...
security.obj: security.cpp, security.h, ...
...
    
```

4.3.2 Configuration Management

...

<Note: Address the four elements of the Configuration management definition:

- Identification and definition of configuration items
- Planning and controlling of releases, versioning and the change process of configuration items
- Recording and reporting of the status of configuration items and change requests
- Verification of the completeness and correctness of configuration items

(cf. lecture notes)>

A1: Glossary

...

A2: Summary of design decisions

...

A3: <Any other relevant appendices...>

...

References

(Gamma, Helm, Johnson & Vlissides, 1995) E. Gamma, R. Helm, R. Johnson, J. Vlissides (1995). Design Patterns. Addison Wesley. ISBN 0.201-63361-2.

...