UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

Desarrollo Basado en Modelos para Interfaces 3D: el Caso de la Cabina de Vuelo

Juan Manuel González Calleros[1], Jaime Muñoz Arteaga[2]
juan.gonzalez@cs.buap.mx, munozar@correo.uaa.mx
[1] Benemérita Universidad Autónoma de Puebla
[2] Universidad Autónoma de Aguascalientes

Outline

1. **Introduction**
2. State of the Art
3. A structured approach to support 3D User Interface Development
4. Conclusion

Introduction

- Web more collaborative
- Increase use of social networks
- Web 2.0

It will be the future landscape of the internet places, not pages?

Introduction

- Current methods:
  - rarely provide the design knowledge
  - are more focusing on the implementation

- Available tools for 3D UIs are: Toolkits, interface builders, rendering engines

Introduction

- Our goal:

  "to demonstrate the feasibility of a MDE-compliant method that is user-centered as opposed to contents-centric for developing 3D UIs"

- We propose:
  - a user-centered approach
  - model-driven architecture
  - separation of concerns

Outline

1. Introduction
2. **State of the Art**
3. A structured method for developing 3DUI
4. Conclusion

State of the art

- Methodological diversity
- Not user centered approaches
- Single entry and single output
- Transformations are hidden

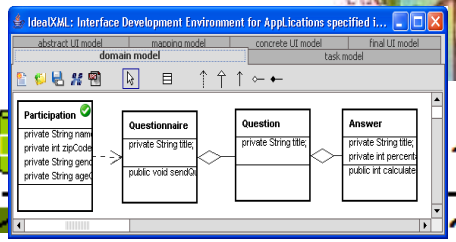
State of the art

- Lack of a methodology for developing 3D UIs
- Lack of user task models
- Lack of models independent to the final implementations
- Lack of a toolkit for such work
- Lack of genuine 3D UIs

Outline

1. Introduction
2. State of the Art
3. **A structured approach to support 3D User Interface Development**
4. Conclusion

Method Outline



Task & Concepts

Task model

Hierarchy level based

[>

Chose relationship / hierarchy*

Confirm

|||

choose Hierarchy level

choose relationship

>>

Toggle hierarchy level

select hierarchy level

Toggle relationship

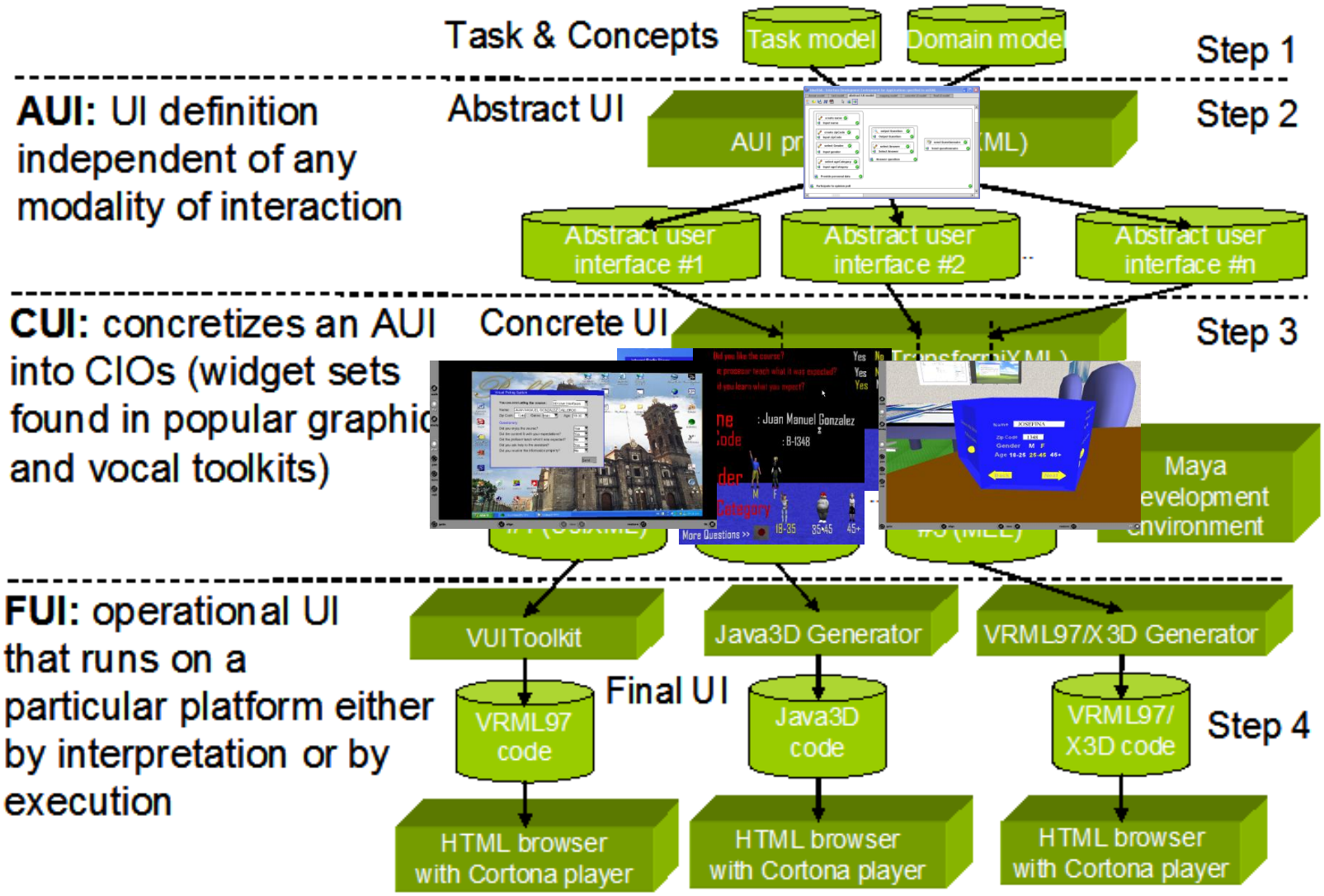
select ralationship

HTML browser with Cortona player

HTML browser with Cortona player

HTML browser with Cortona player

Method Outline



Caso de Estudio

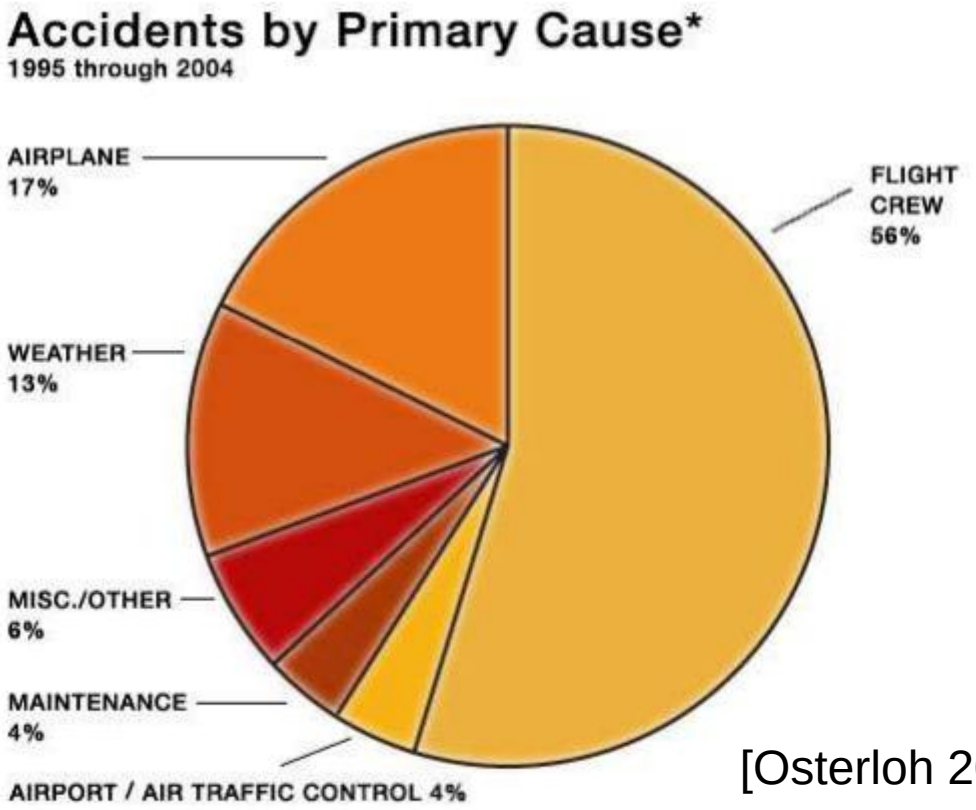
Aplicación del diseño basado en Modelos para cabinas de pilotaje

El problema

- El porcentaje de accidentes aéreo es estable, varia entre 3 y 4 accidentes por millón de salidas
- Para el futuro se anticipa un aumenta en el volumen de vuelos
- En consecuencia se espera tener un accidente serio cada semana si no se logra reducir el procentaje de accidentes
- 55% de los accidentes involucra a la tripulación

El problema

- La principal causa son errores humanos
- Incremento en la automatización del vuelo
  - Piloto se olvida de volar
  - Exceso de confianza sobre las maquinas
  - Sorpresas que los sistemas no consideran
- Tomar en cuanta factores humano en el diseño de sistemas es crucial



[Osterloh 2011]

Photo credit: © Lars Langemeier/A.B./zefa/Corbis
http://www.pbs.org/wgbh/nova/space/pilot-room-for-error.html (Posted 2006/10/17; Boeing Study)

El problema

- Tenemos que tomar en cuenta al piloto (usuario) en la fases de desarrollo de un avión
- Hay que reducir el porcentaje de errores humanos
- Los sistemas de automatización de cabinas de pilotaje han contribuido de forma notable a la reducción de este porcentaje

El problema

- Pilotos
  - Son difíciles de conseguir, tienen una agenda muy apretada
  - Son muy caros, usarlos para hace pruebas sobre prototipos tiene un costo prohibitivo
- Sistemas de Pruebas (Simuladores)
  - Caros
  - Difícil acceso

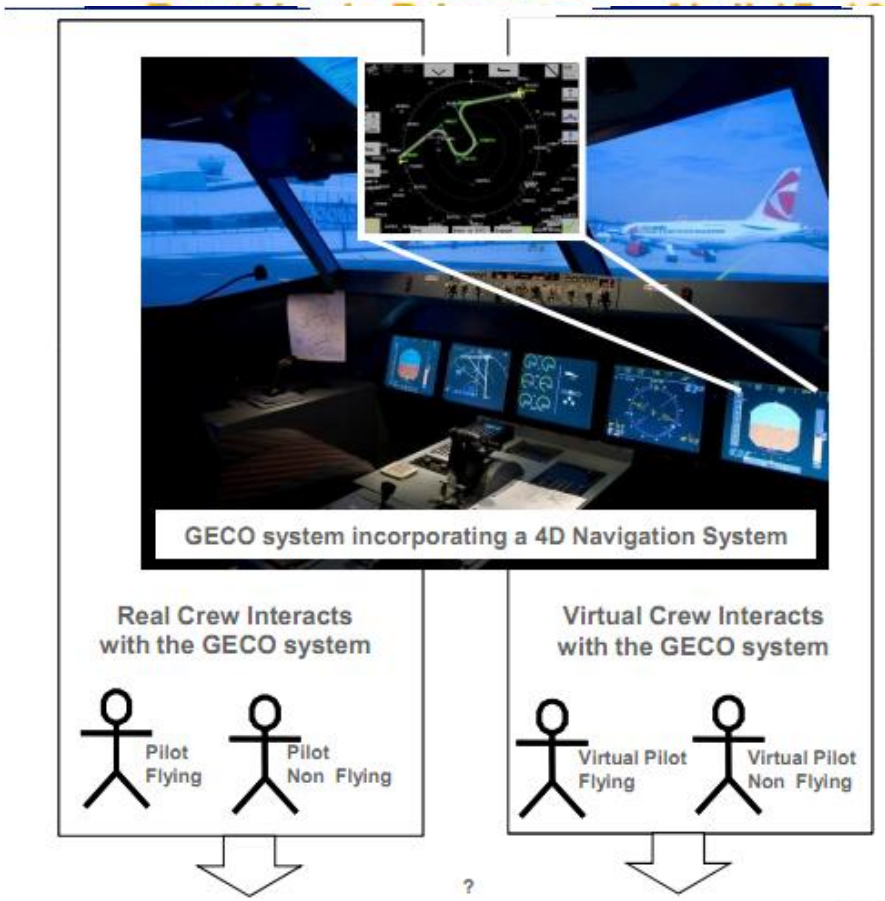
→ Pilotos + Sistemas de Pruebas
  - Exclusivo para centro muy especializados

Propuesta de Solución

- Crear un modelo cognitivo de la tripulación
  - Piloto Volando (PF) y Piloto no Volando (PNF) para reducir necesidad de usar pilotos reales
  - Herramientas de software para dar soporte
  - Simulador de sistema de vuelo
  - **Desarrollo basado en modelos de la ergonomía de las interfaces**
- Con esto se espera predecir ciertos comportamientos del piloto que causen errores y diseñar sistemas que los prevengan

Que tipo de errores comete el piloto

- Se aburre y no sigue el procedimiento
- Sobre carga de información y omite mensajes de alerta importantes que lo hace no seguir los procedimientos



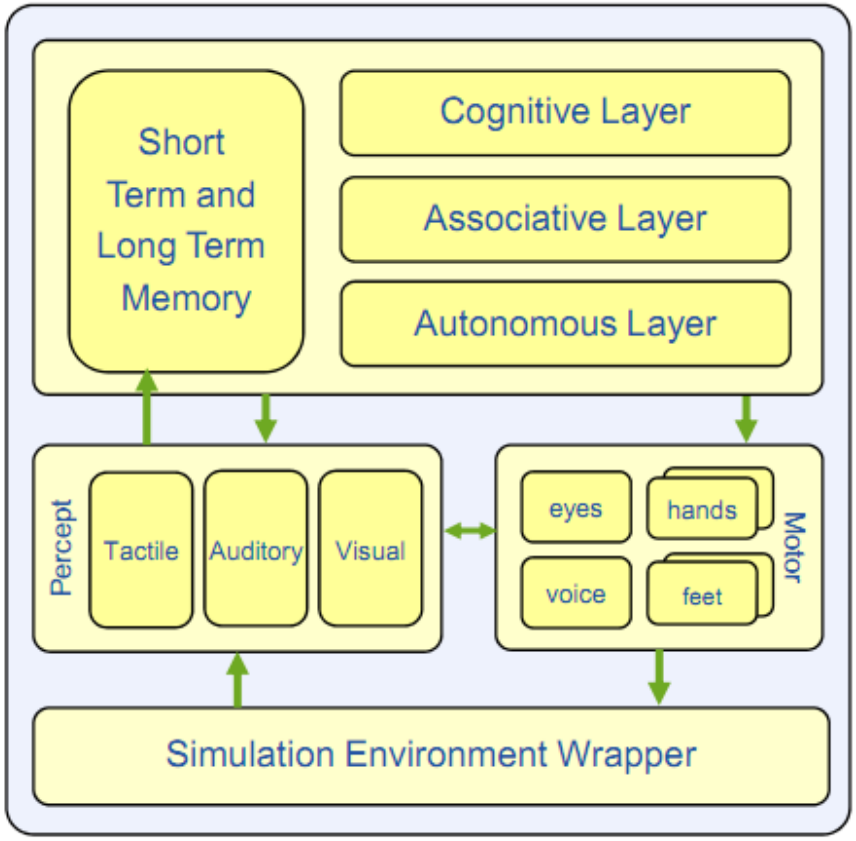
Acciones Reales

Predicción de Acciones

[Osterloh 2011]

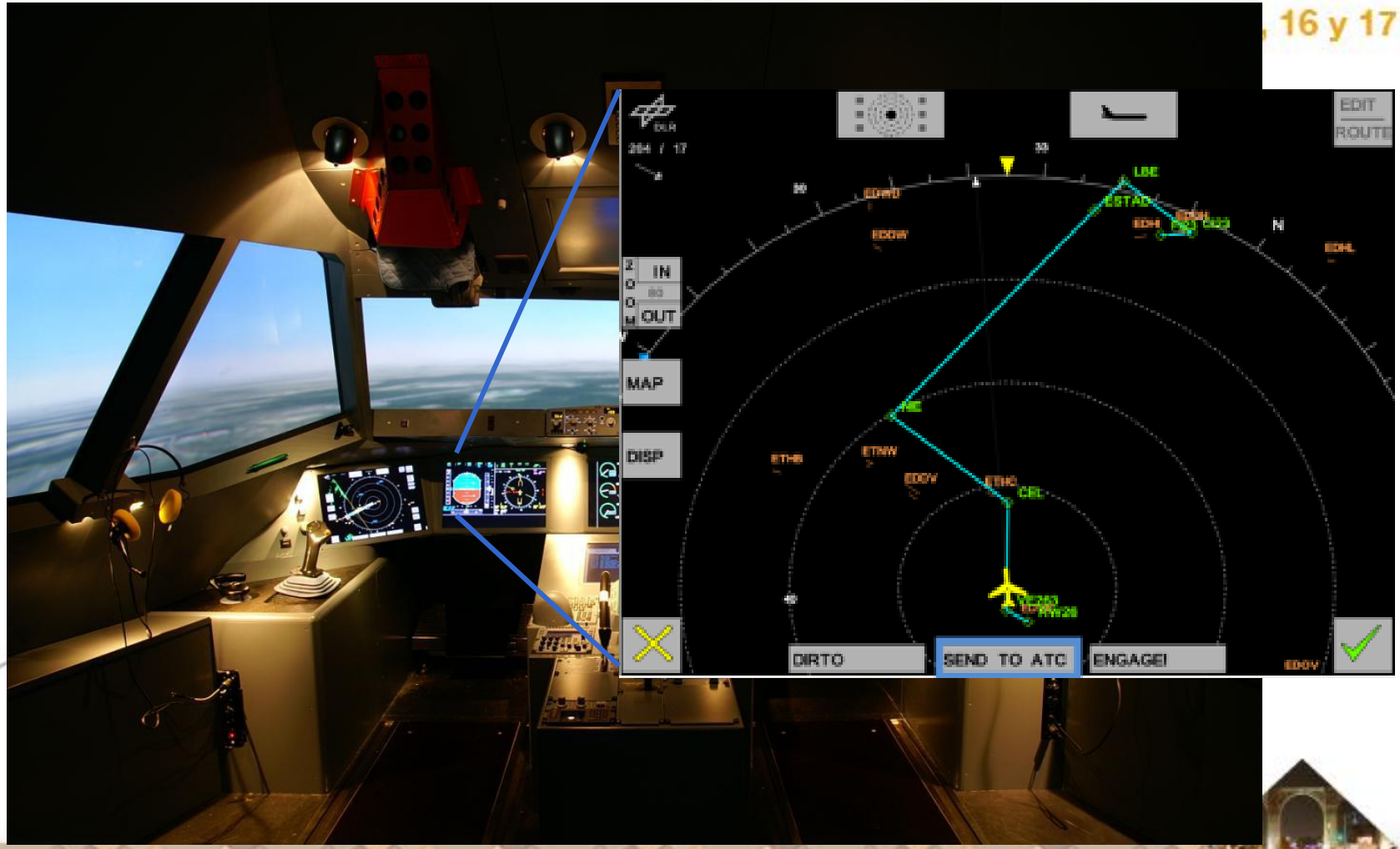
La arquitectura Cognitiva

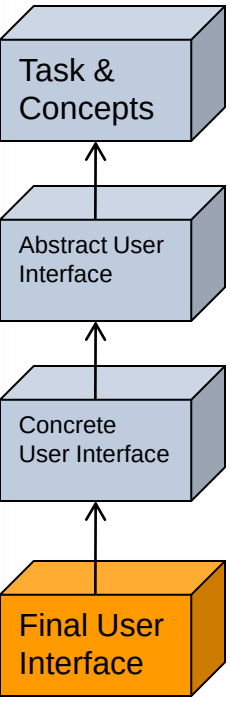
- Modela el comportamiento humano tomando en cuenta factores:
  - Psicológicos
    - búsqueda en la memoria
    - Olvido
    - procesamiento de conocimiento)
  - Fisiológicos
    - movimiento de ojos
    - Manos
    - voz
    - pies



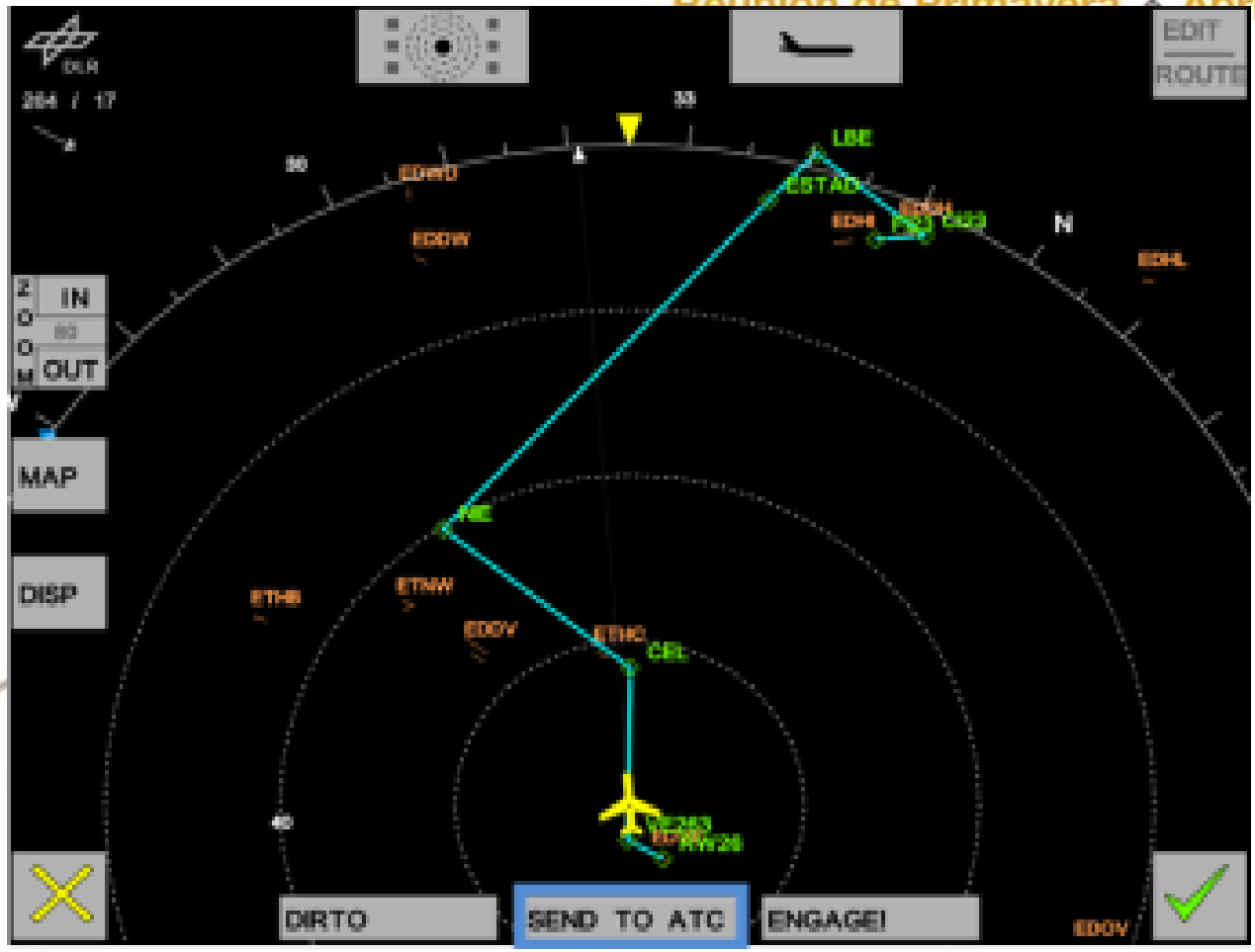
[Osterloh 2011]

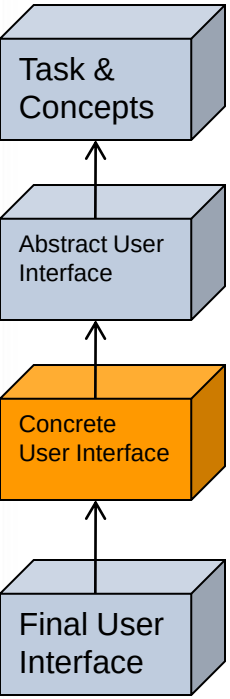
La Interfaz de Usuario requiere un modelo



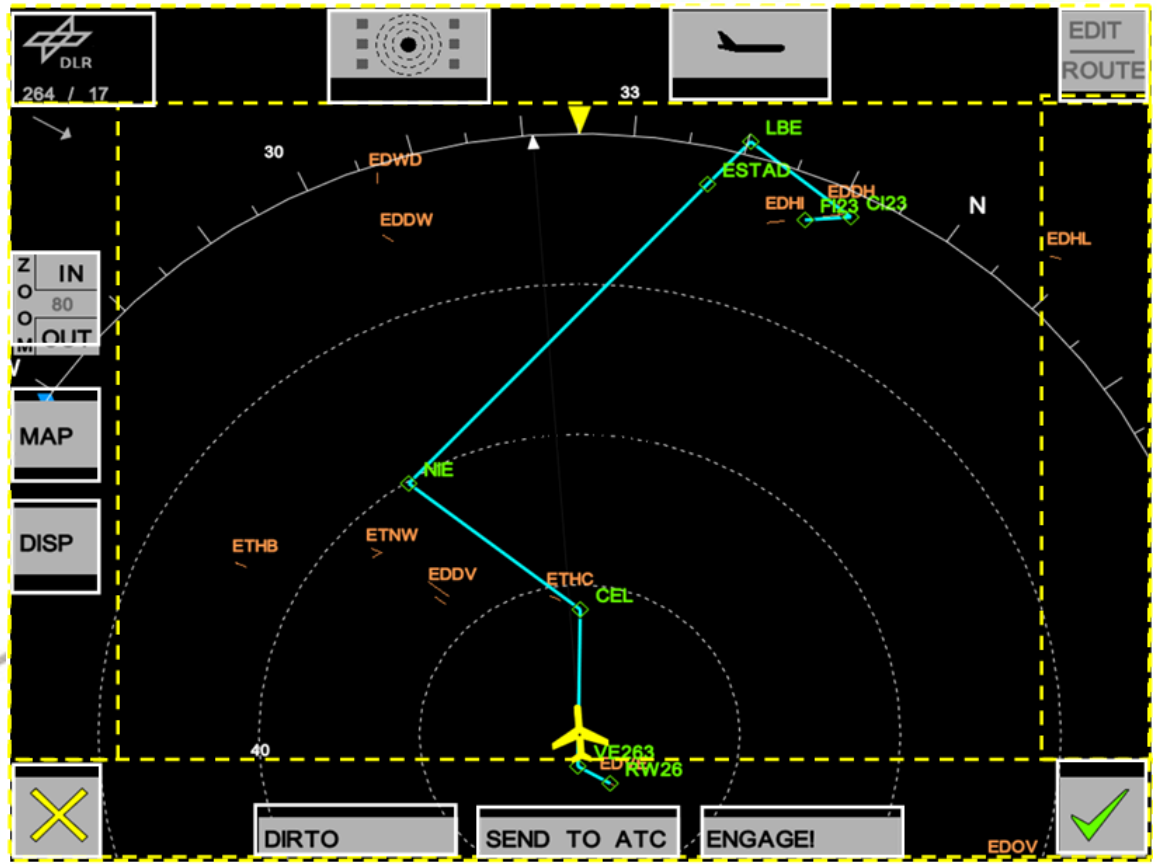


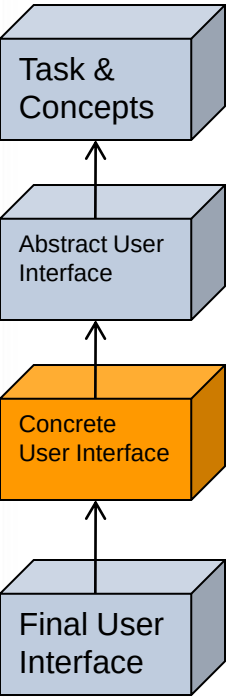
Interfaz para navegación



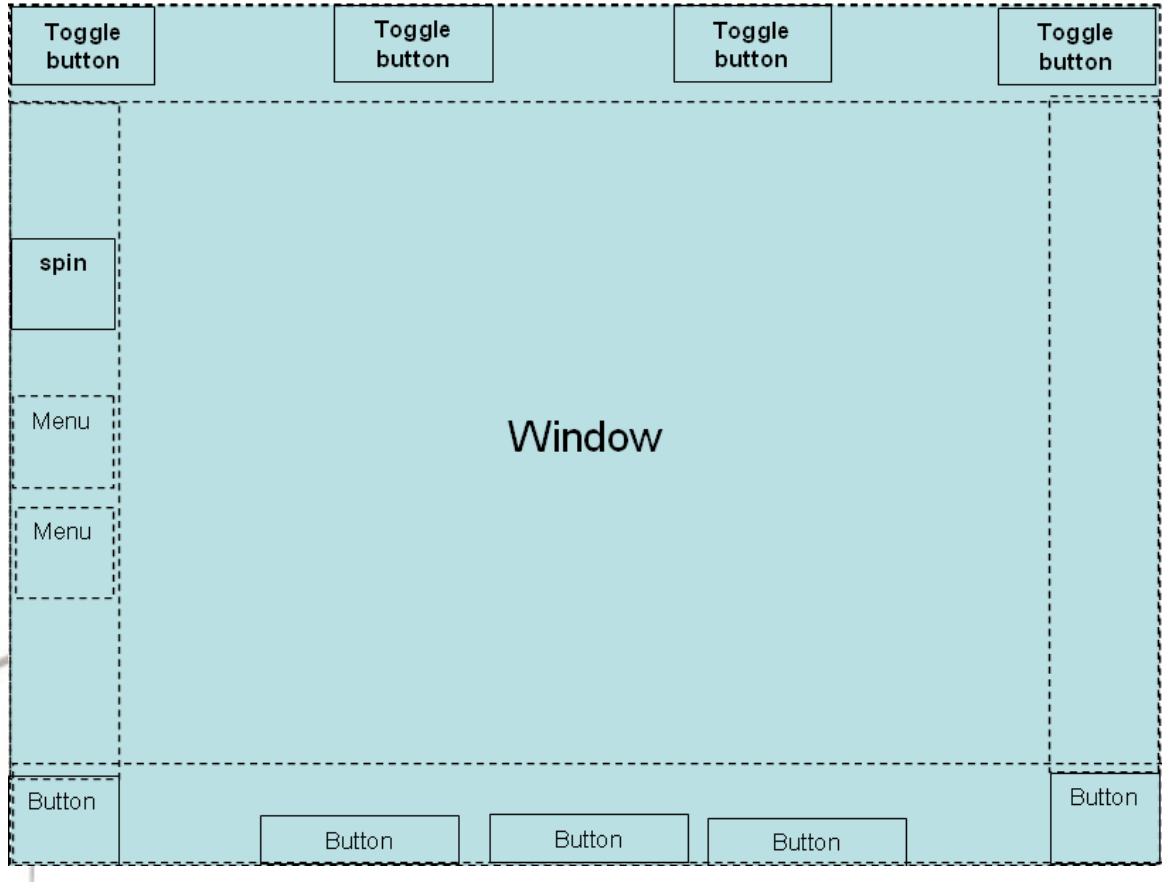


Layout-Interfaz para navegación

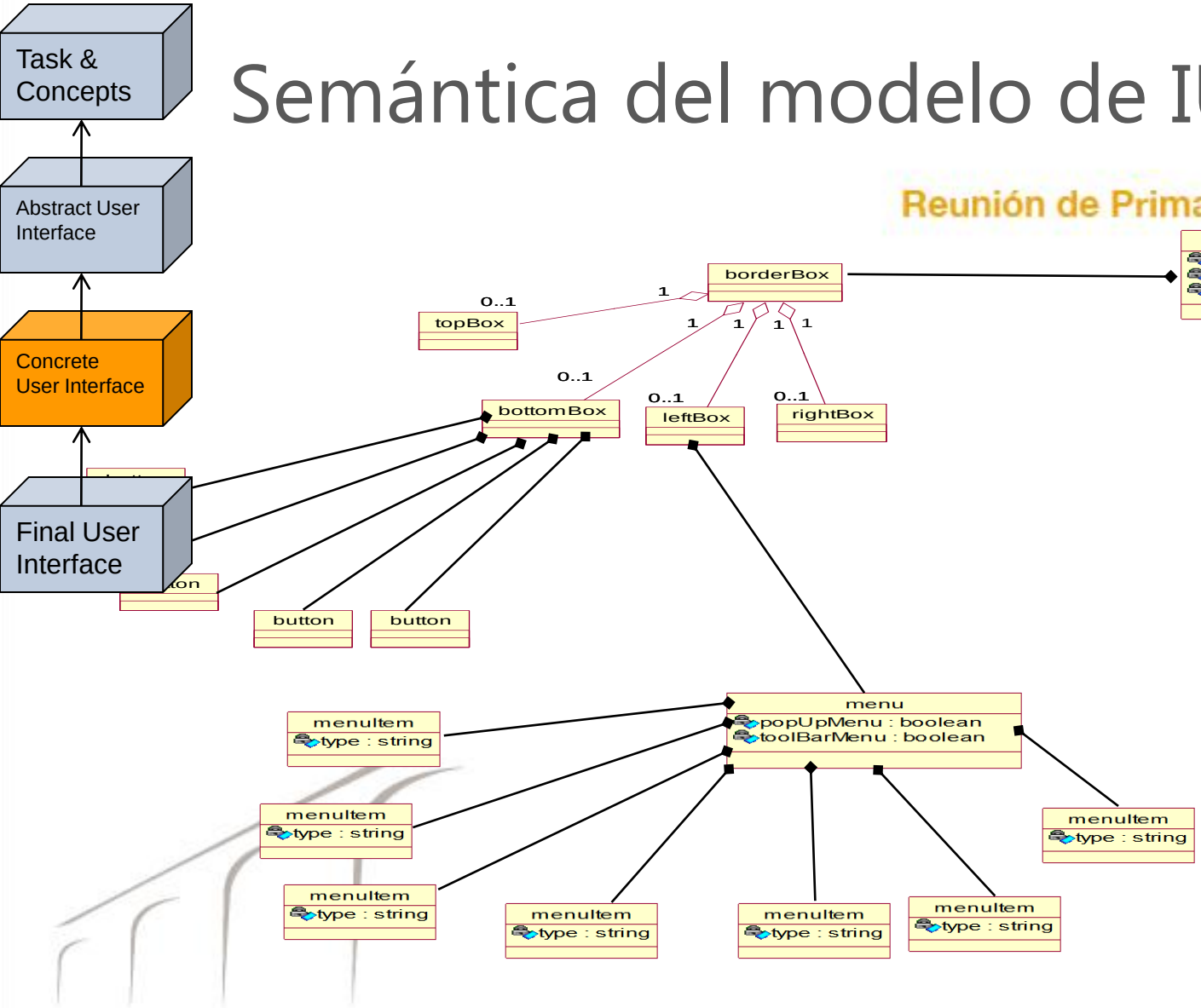




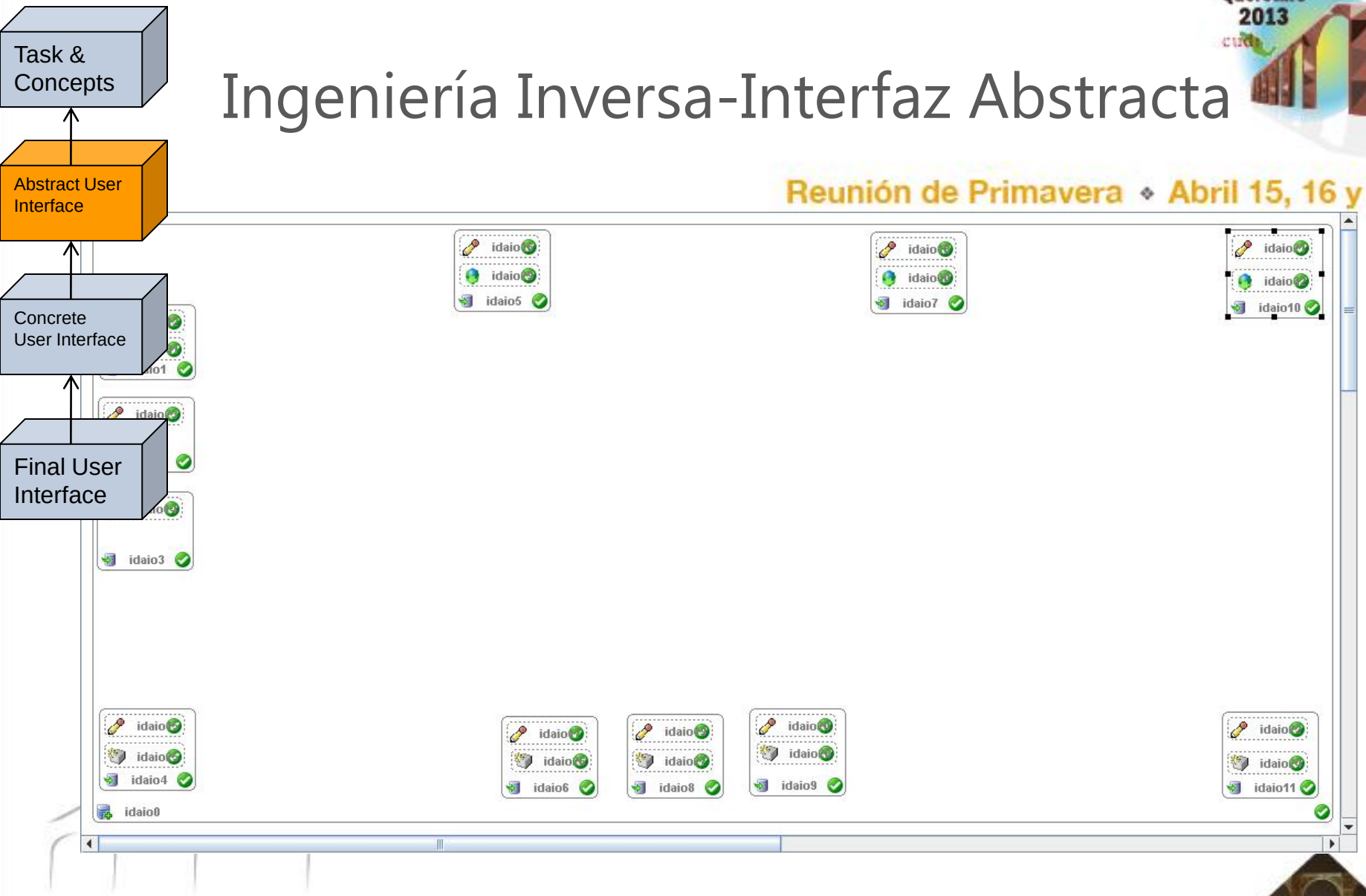
Plano de la IU

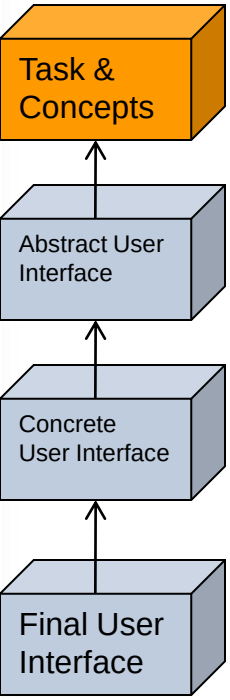


Semántica del modelo de IU



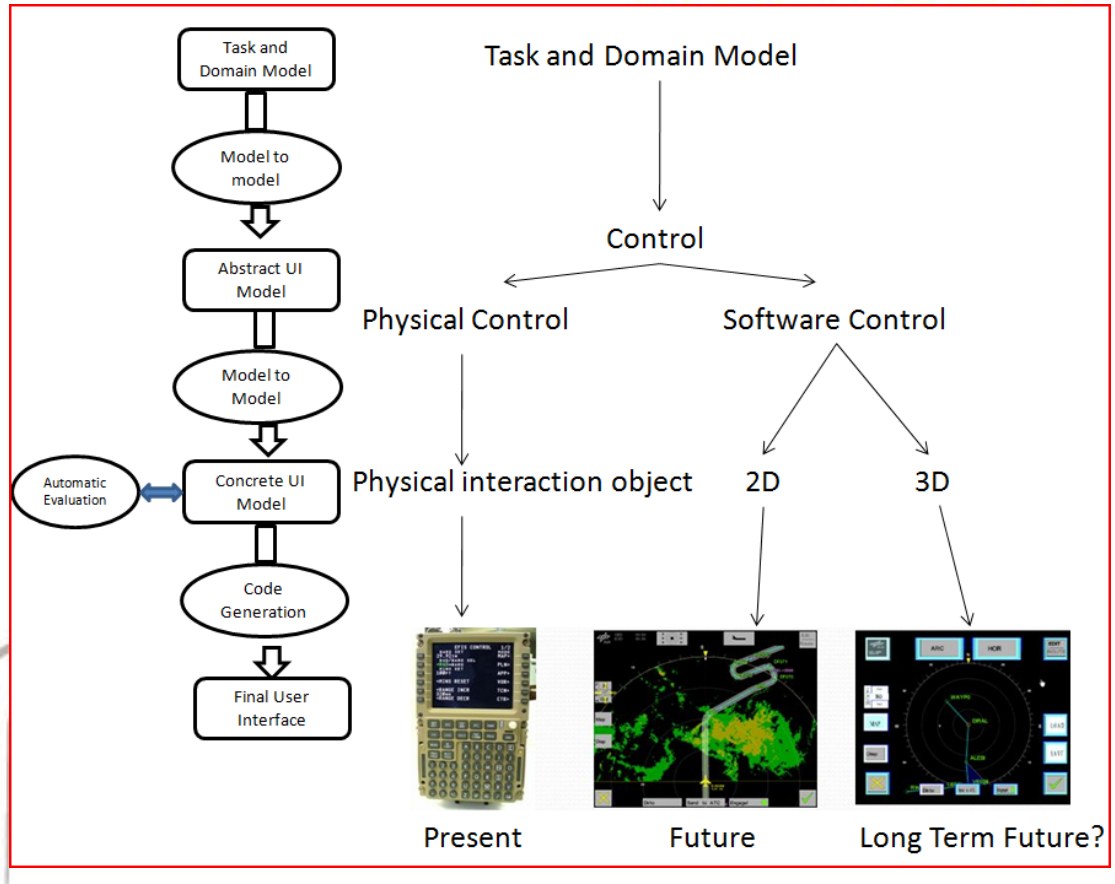
Ingeniería Inversa-Interfaz Abstracta



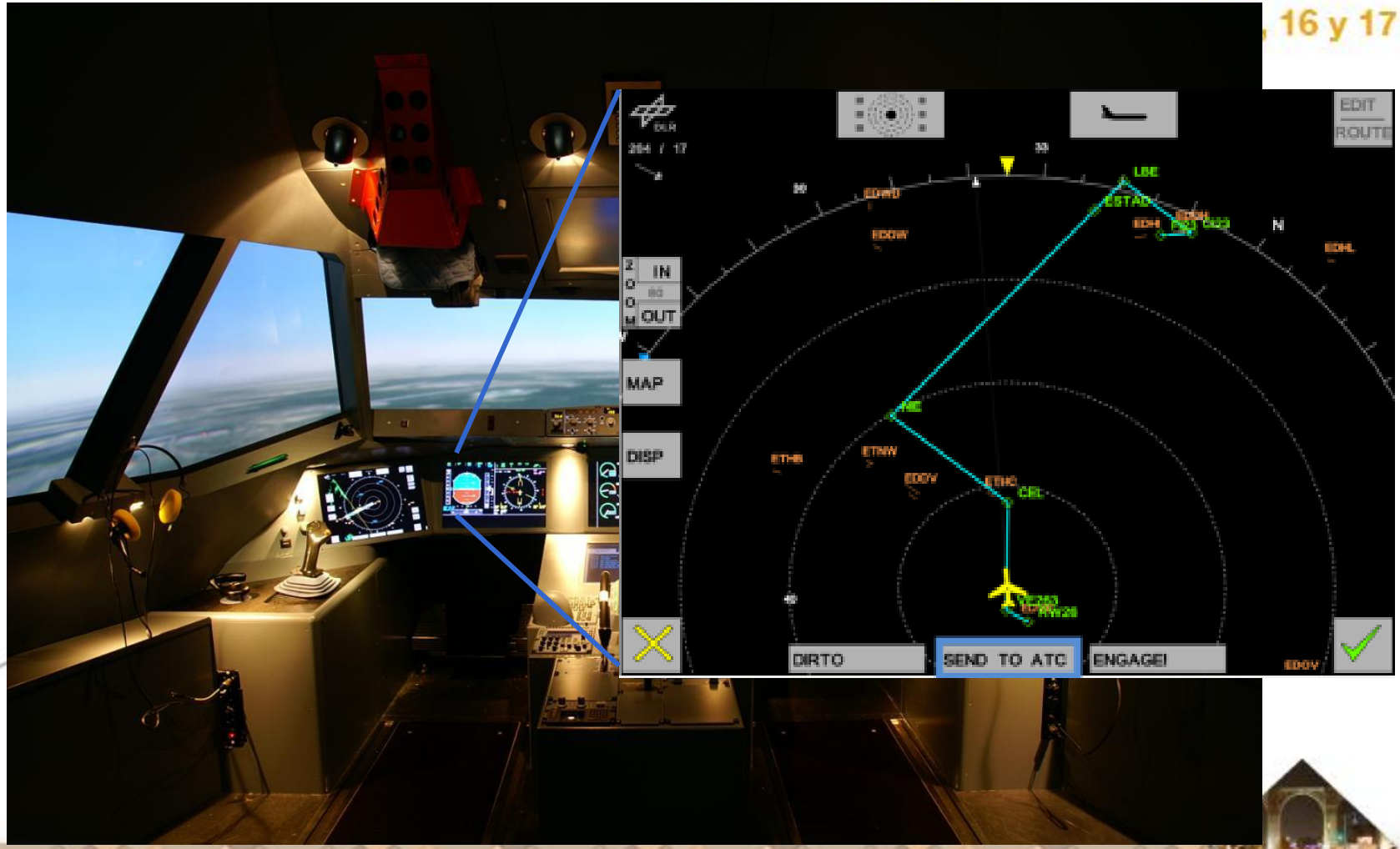


Ingeniería Inversa-Modelo de Tareas

A partir de este modelo se pueden generar muchas soluciones



La Interfaz de Usuario requiere un modelo



Pruebas de usabilidad de IUs

- Maquina 3D

Outline

1. Introduction
2. State of the Art
3. Model-based development of 3D user interfaces
4. **Conclusion**

Conclusion

- We presented a 3DUI Engineering methodology articulated on three axes: models and their specification language, method, and tools that support the method based on the underlying models.

Conclusion

- It has been demonstrated that the global methodology adheres to the principles of MDA and is therefore compliant, except for the standardization process which is ongoing.

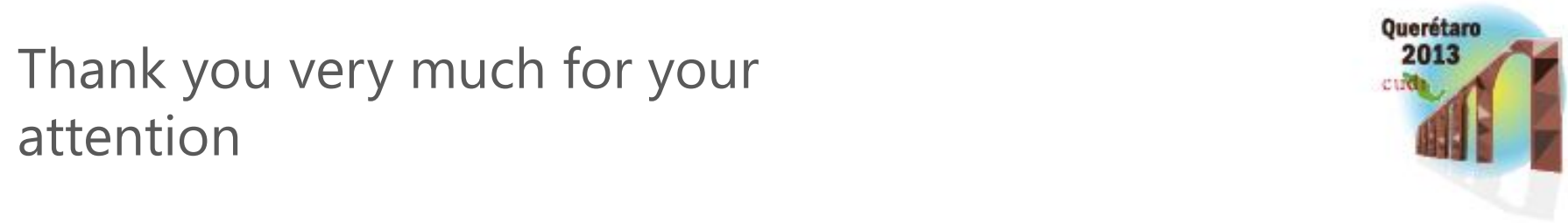
Conclusion

- It has been demonstrated that the global methodology adheres to the principles of MDA and is therefore compliant, except for the standardization process which is ongoing.

Conclusion

- Advantages
  - Modifiability
  - Complexity
  - Rigorous.
  - Reasoning.
  - Processable.

Thank you very much for your attention



Reunión de Primavera ◆ Abril 15, 16 y 17

http://www.usixml.org

User Interface eXtensible Markup Language

For more information and downloading,

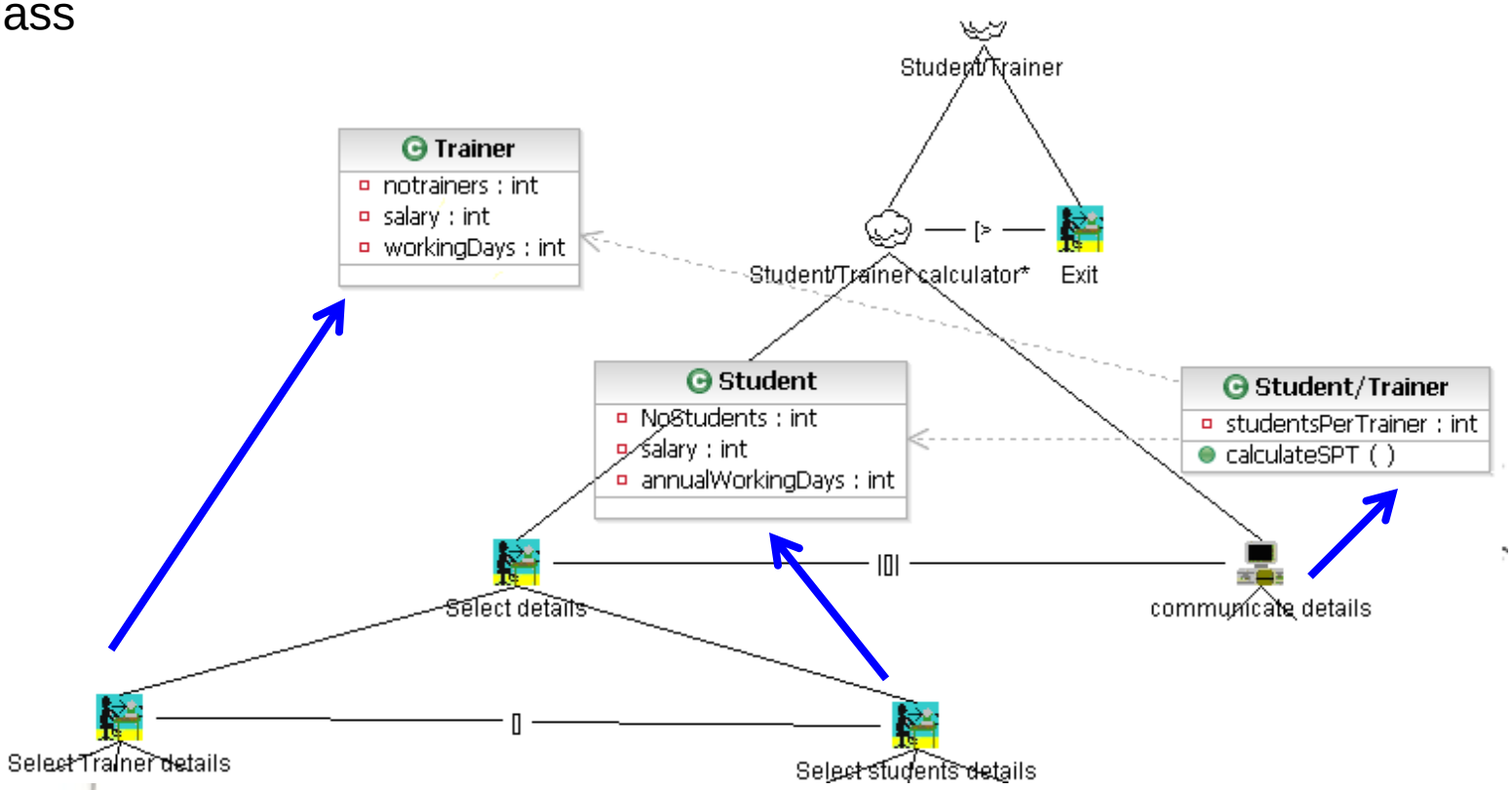
http://www.isys.ucl.ac.be/bchi

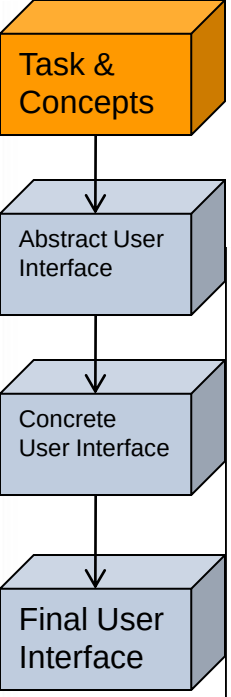
Special thanks to all members of the team!

ADDITIONAL INFORMATION

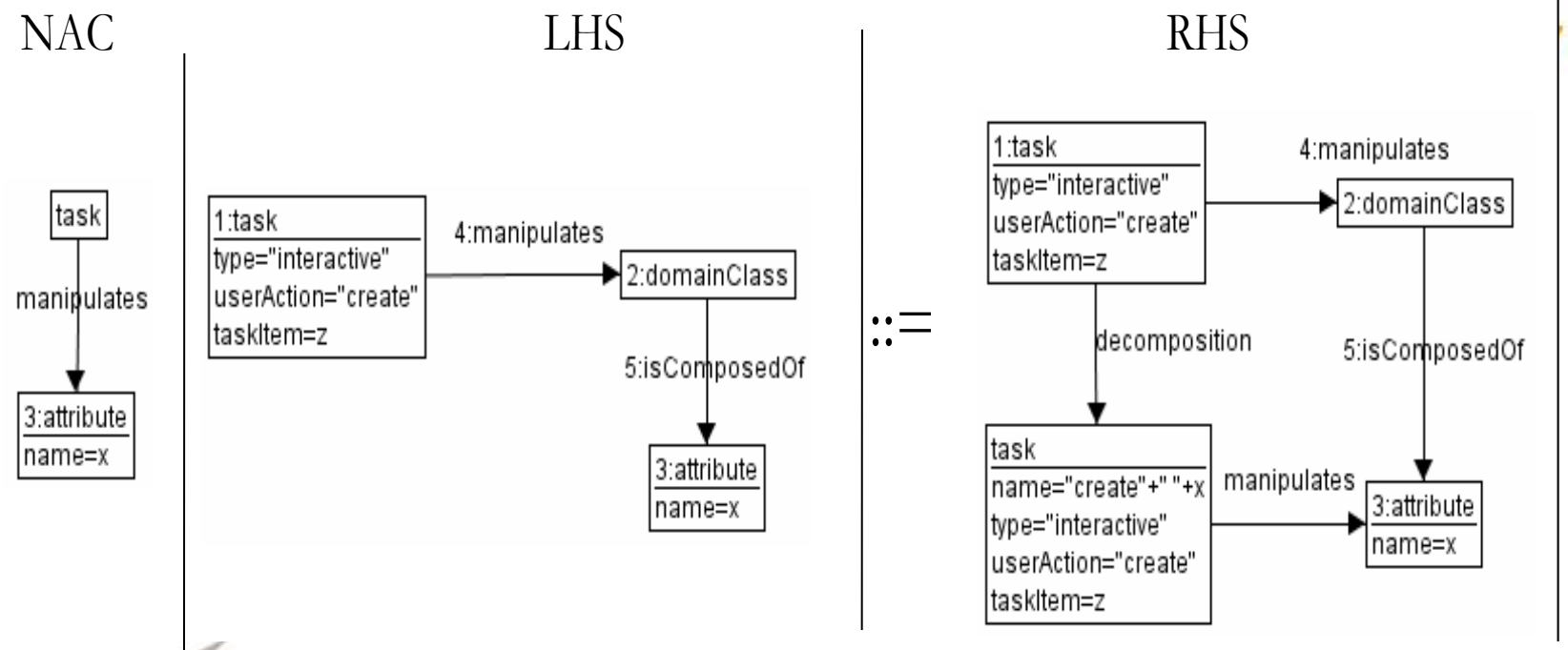
Task and Domain Model

Rule: A task manipulates a domain class





Step1. Consolidation of the task Model



Rule: For each task that manipulates a domain class, a new subtask is created for each attribute.

Each of the new sub-tasks will be mapped on the corresponding attribute of the class

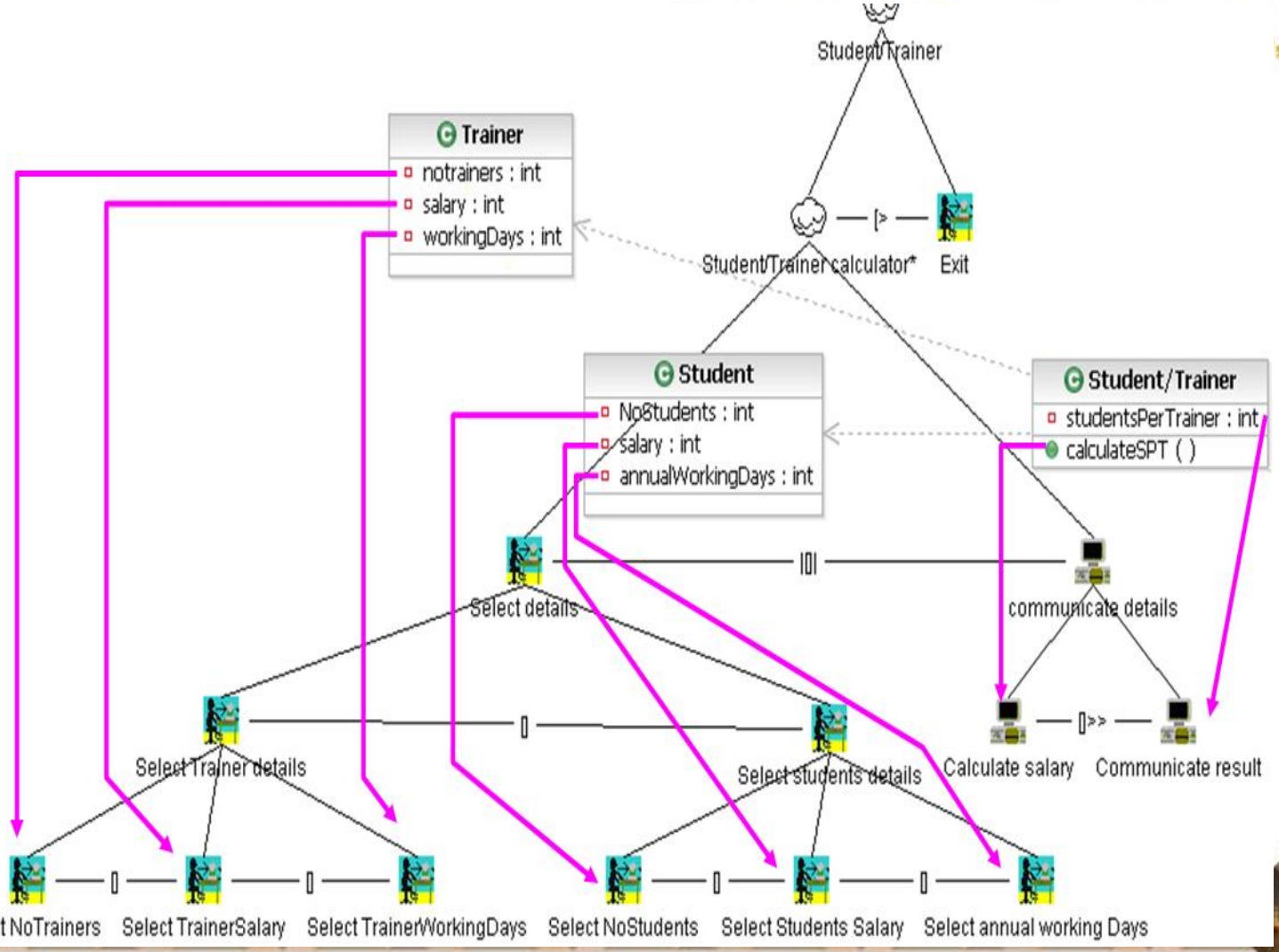
Task & Concepts

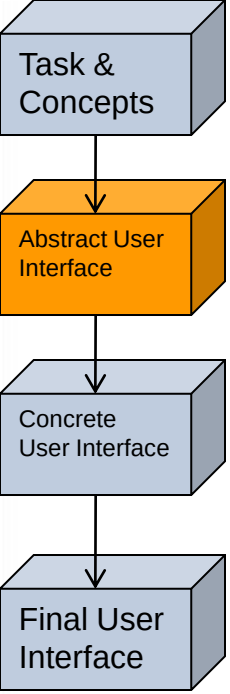
Abstract User Interface

Concrete User Interface

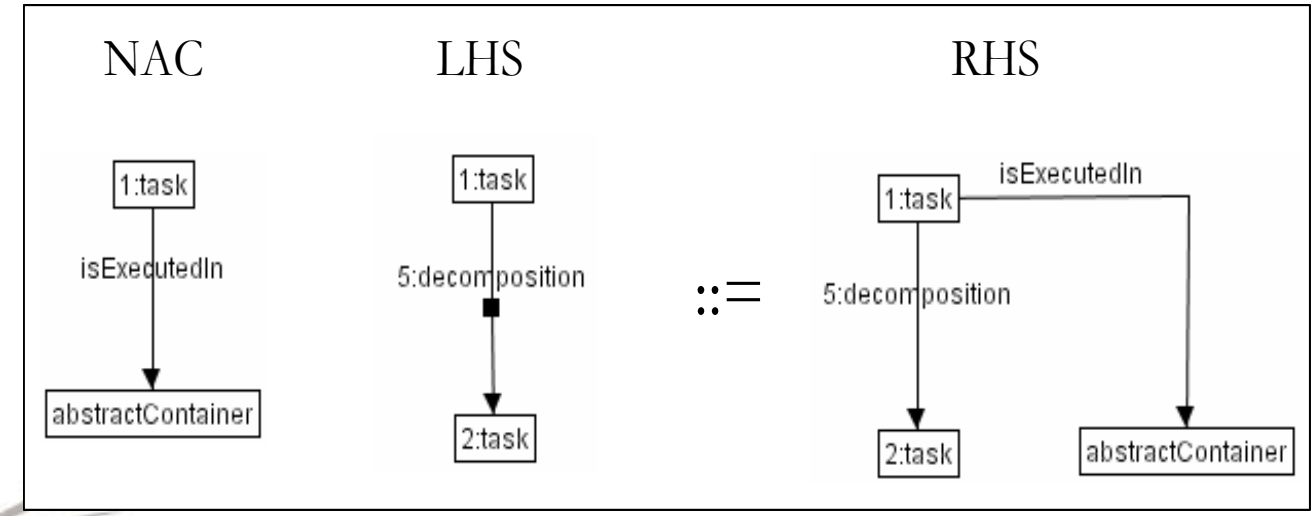
Final User Interface

Step1. Consolidation of the task Model



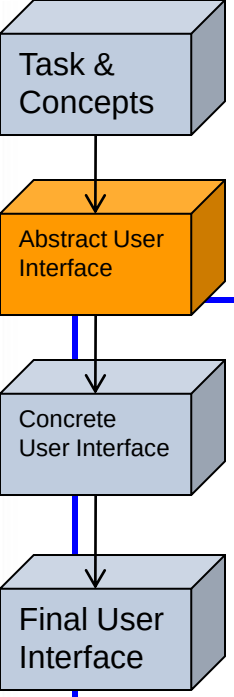


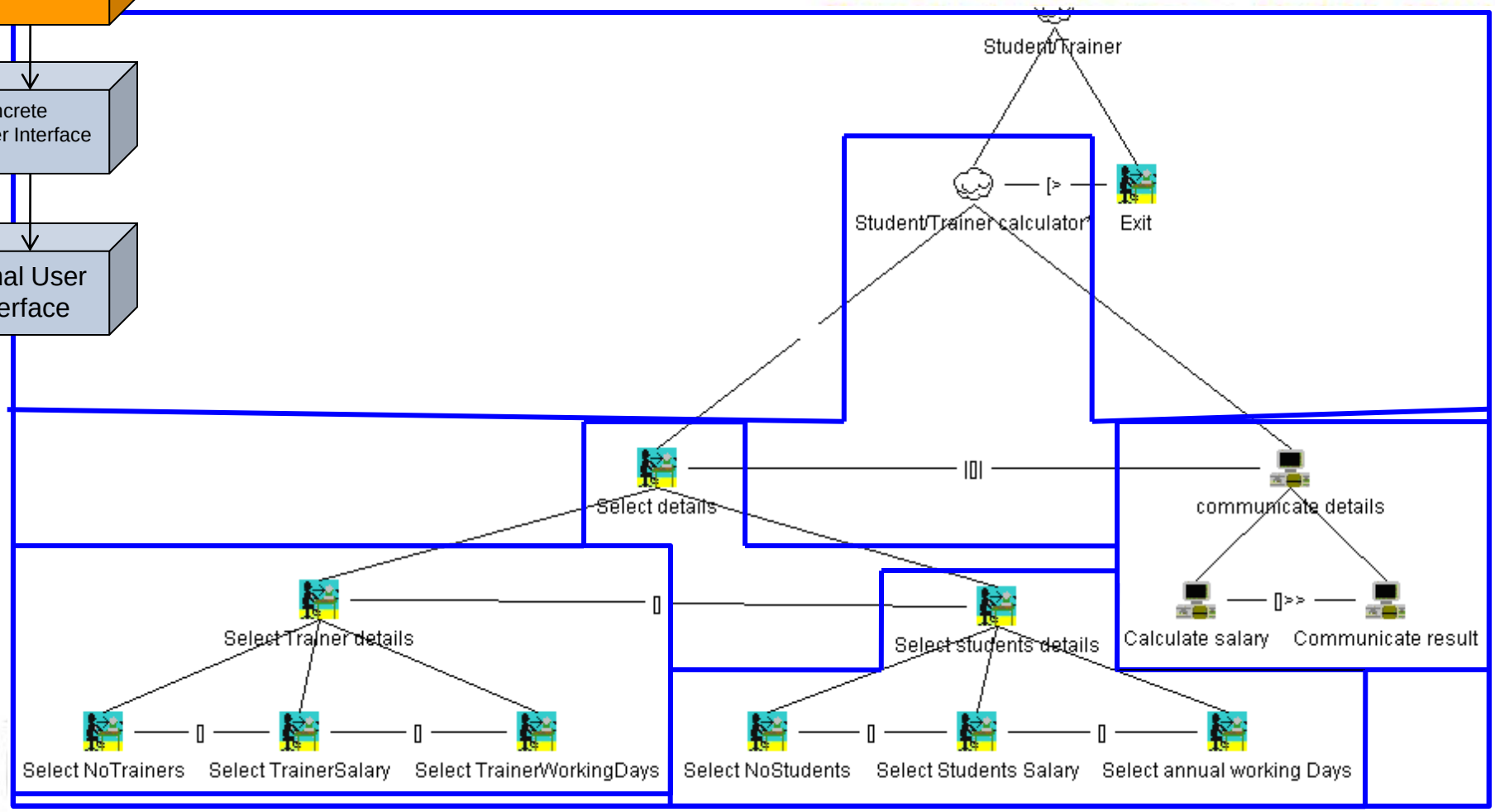
Step 2: From task to Abstract User Interface Model

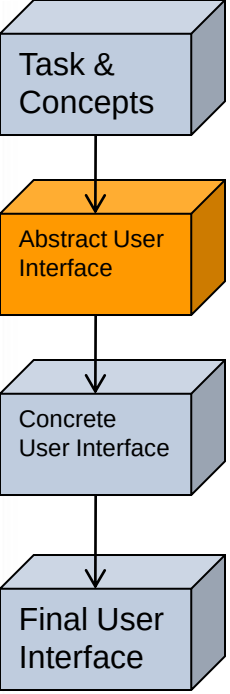


Each task is executed in an abstract container if the task is decomposed into subtasks

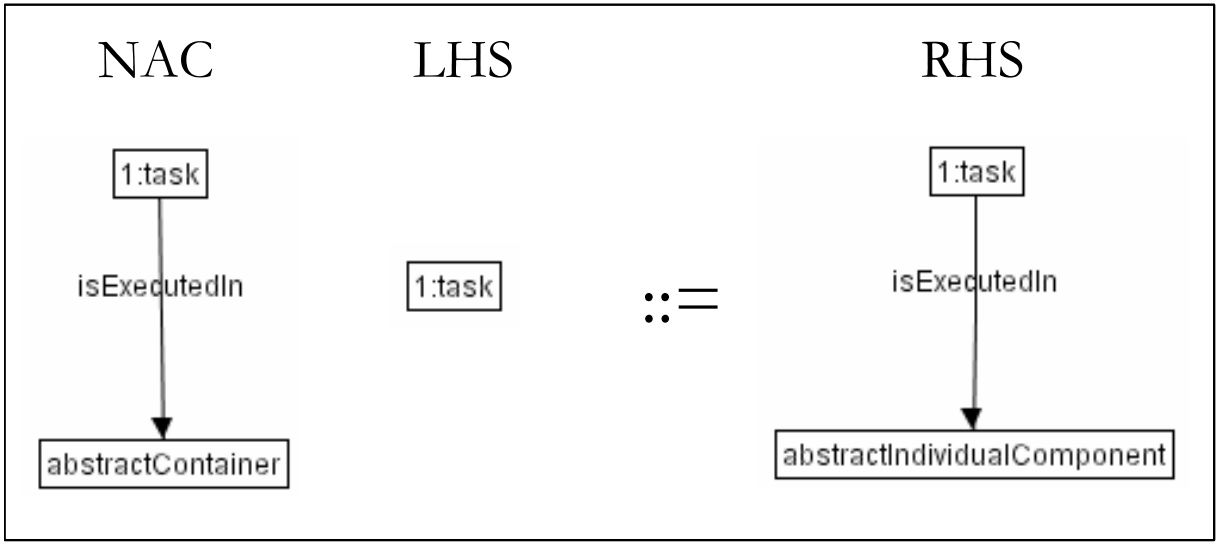
Step 2: From task to Abstract User Interface Model





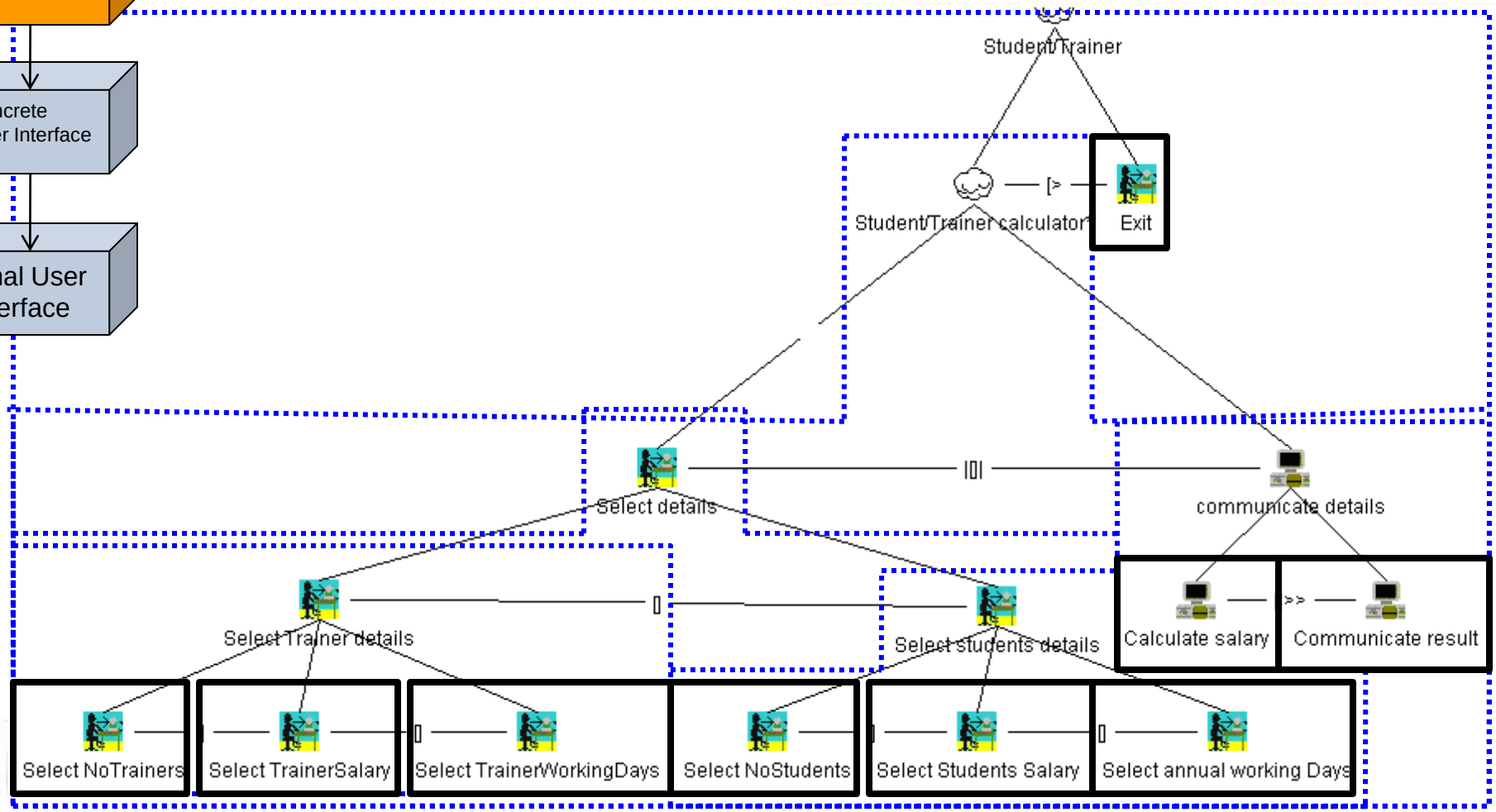


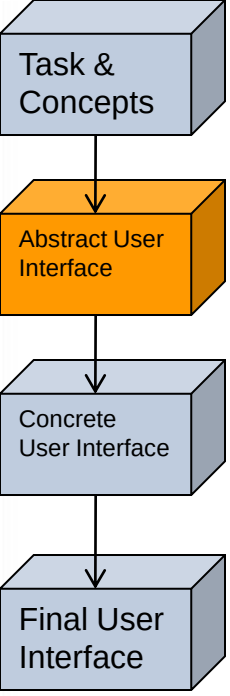
Step 2: From task to Abstract User Interface Model



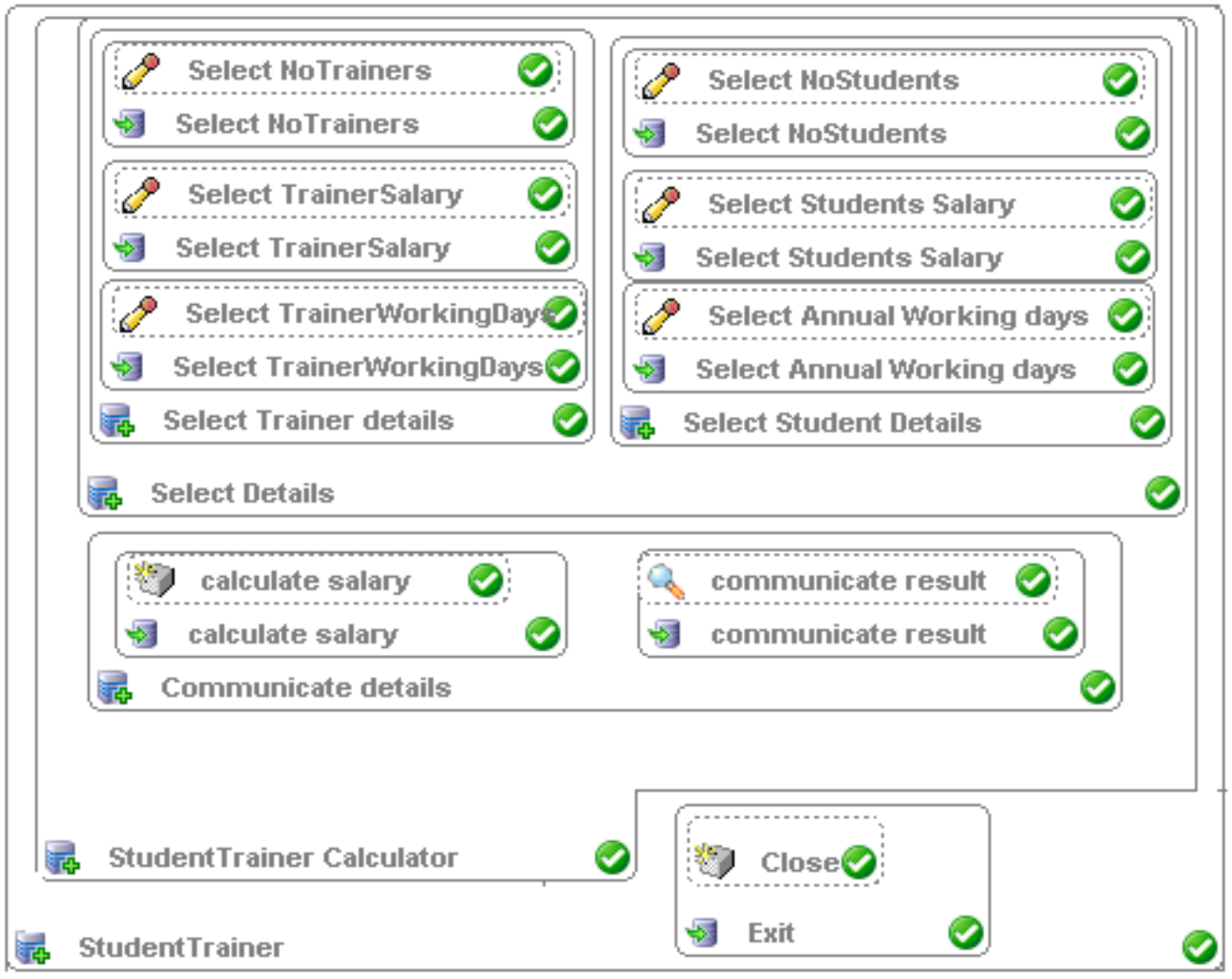
Each leaf task is executed in an abstract individual component.

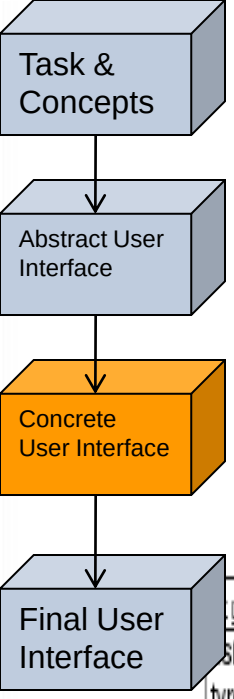
Step 2: From task to Abstract User Interface Model





Step 2: From task to Abstract User Interface Model





Step 3: From Abstract User Interface to Concrete User Interface Model

NAC

:graphicalContainer
si_type=box
type=horizontal

graphicalContainment

graphicalIndividualComponent
xsi_type= slider
defaultContent=x
groupName=y

LHS

abstractContainment

1:abstractContainer

2:abstractIndividualComponent
name=y

9:isReifiedBy

12:isReifiedBy

isComposedOf

4:graphicalContainer

graphicalContainment

3:facet
type=input
actionType=select
actionItem=element
inputCarMin = 0
inputCarMAx=100
inputDataType=int

6:graphicalContainer
xsi_type=box
type=horizontal

RHS

abstractContainment

1:abstractContainer

2:abstractIndividualComponent
name=y

9:isReifiedBy

isComposedOf

4:graphicalContainer

12:isReifiedBy

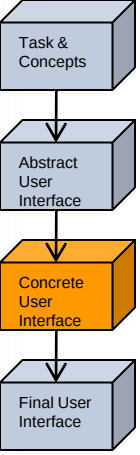
graphicalContainment

3:facet
type=input
actionType=select
actionItem=element
inputCarMin = 0
inputCarMAx=100
inputDataType=int

6:graphicalContainer
xsi_type=box
type=horizontal

graphicalContainment

graphicalIndividualComponent
xsi_type= slider
defaultContent=x
groupName=y



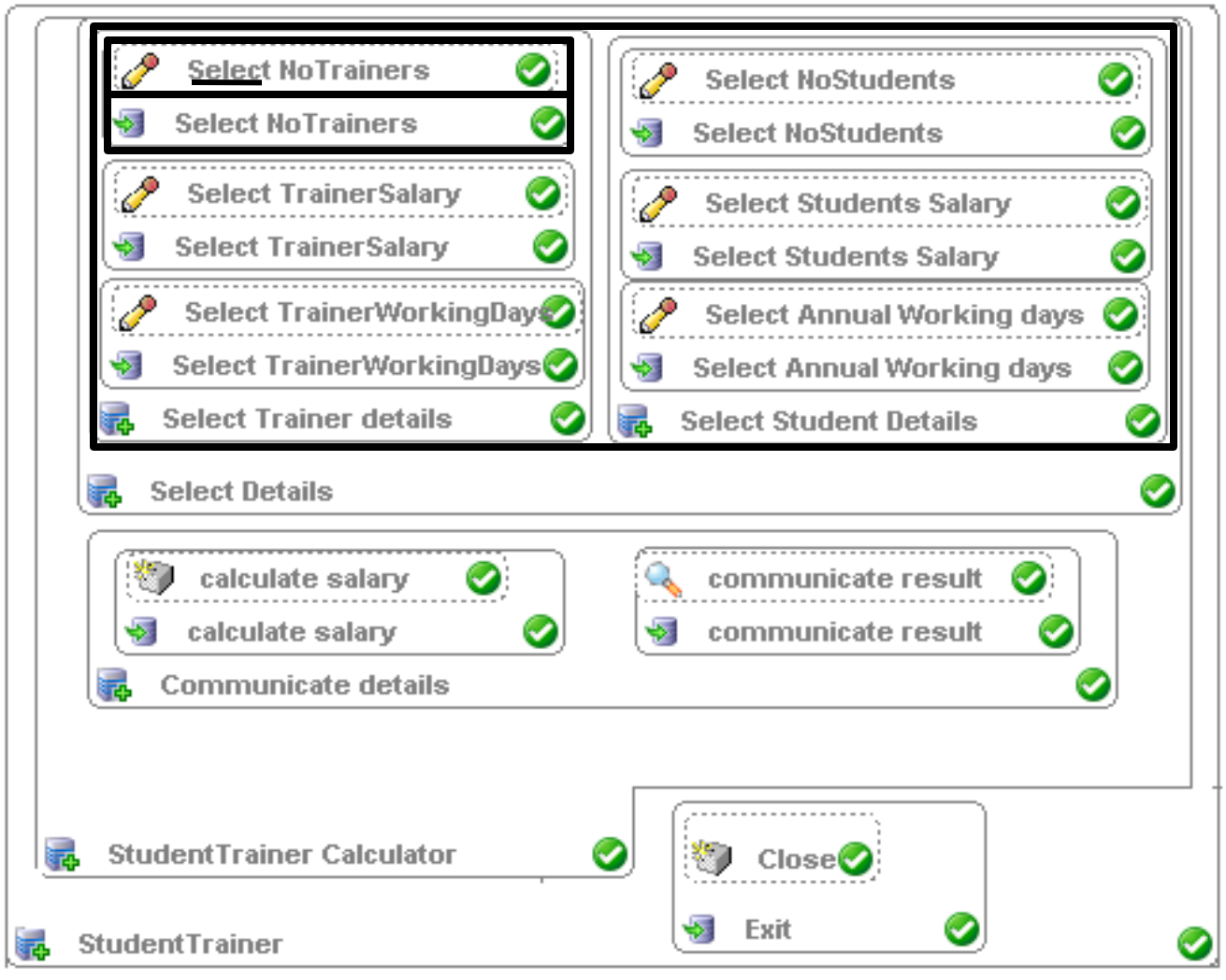
Step 3: From Abstract User Interface to Concrete User Interface Model

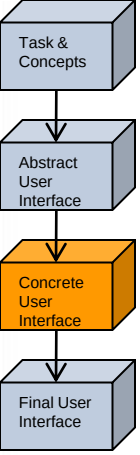
For each Abstract Container

That contains an abstract individual component (AIC)

And the AIC is composed of an input facet

And the facet has an action type select



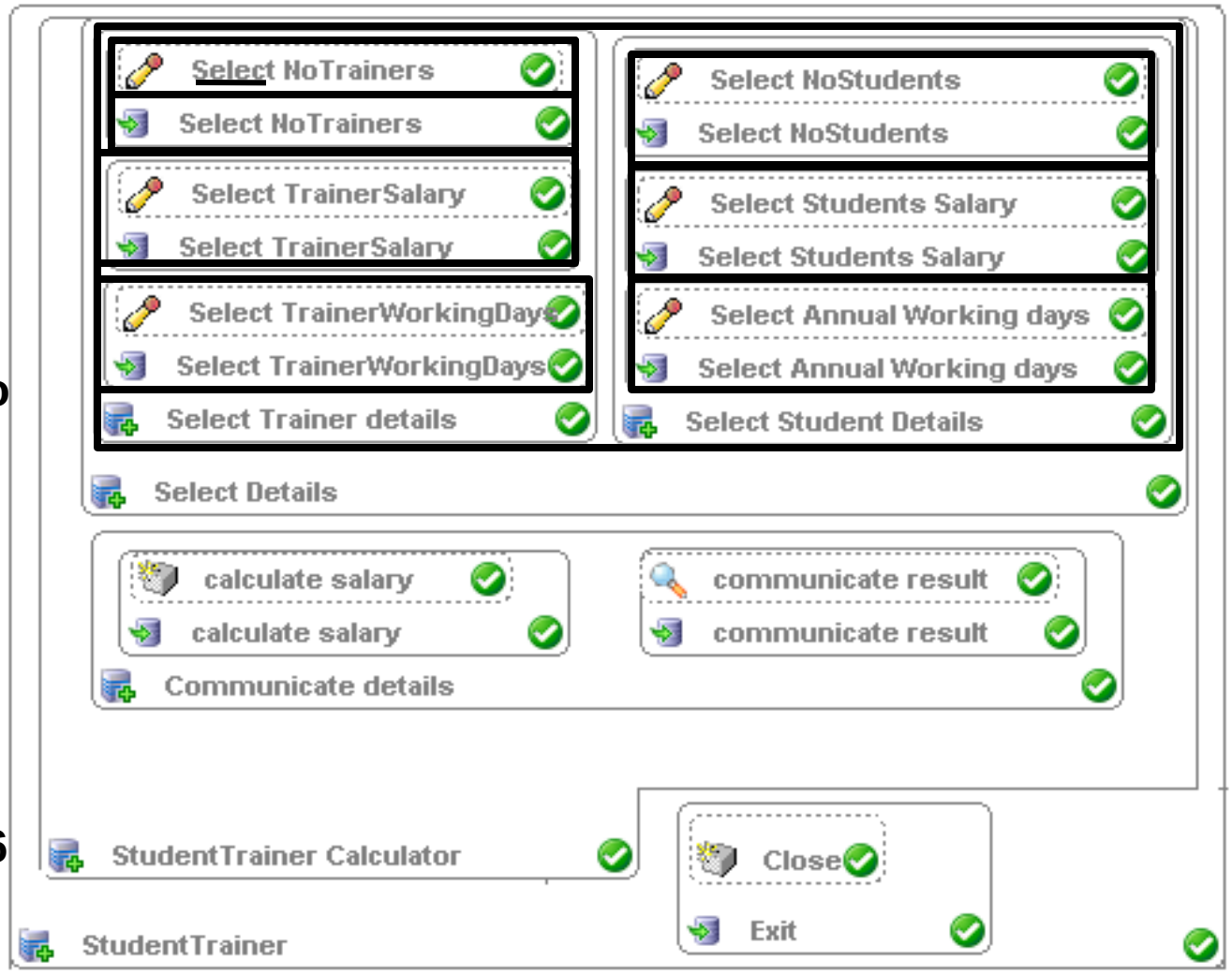


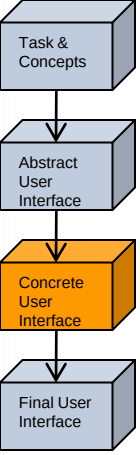
Step 3: From Abstract User Interface to Concrete User Interface Model

And the domain of the data is continuous

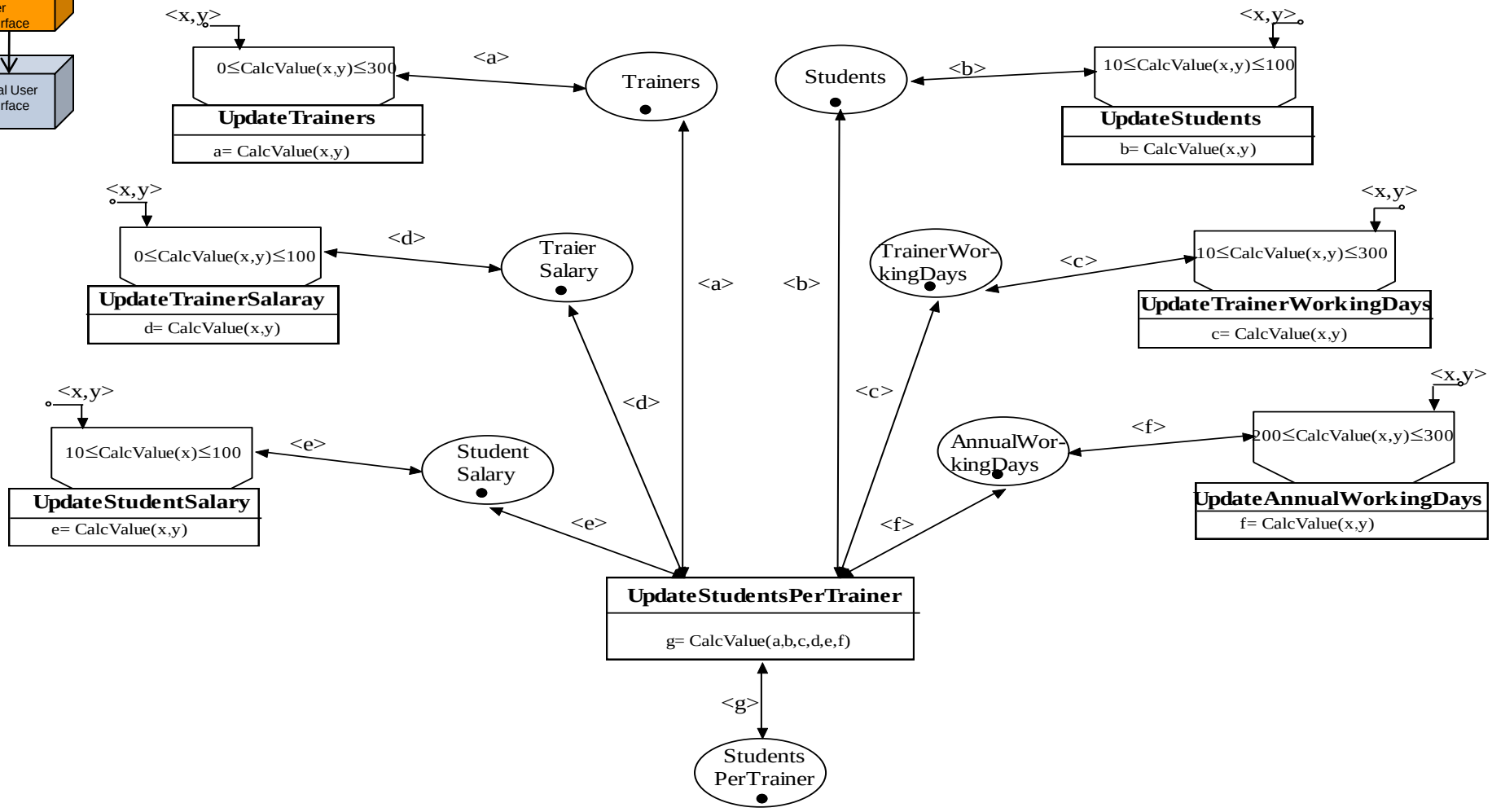
Then concretize the AIC into a Slider contained in a graphical container (Box)

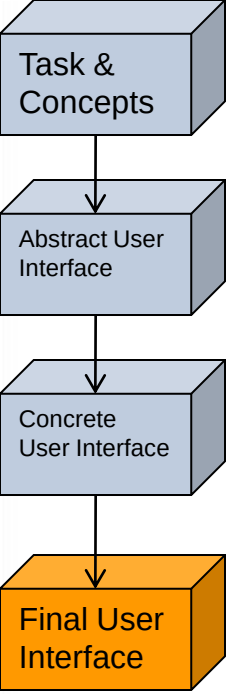
The rule is applied to 6 AIC





Step 4: Adding behaviour





Final Result