

## Conception d'applications interactives

## Méthodologie de conception

### • Les enjeux

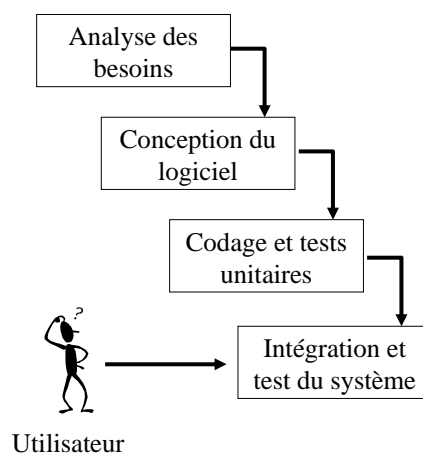
- ↪ 80 % du code des systèmes interactifs est consacré à l'interface utilisateur
- ↪ Les **risques** d'une mauvaise interface :
  - Rejet pur et simple par les utilisateurs (technopathie)
  - Coût d'apprentissage (formation)
  - Perte de productivité
  - Utilisation incomplète (manque à gagner)
  - Coût de maintenance
  - Perte de crédibilité

## Erreurs de conception

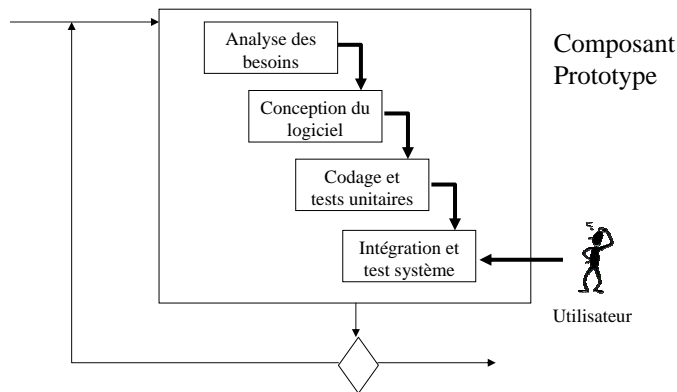
- **Les causes :**

- Théories et modèles inadaptés
  - Absence de solutions opérationnelles
  - Incapacité à généraliser les nouvelles techniques d'interaction
- Formation des concepteurs, développeurs et décideurs
  - Informatique
  - Sciences humaines et sociales

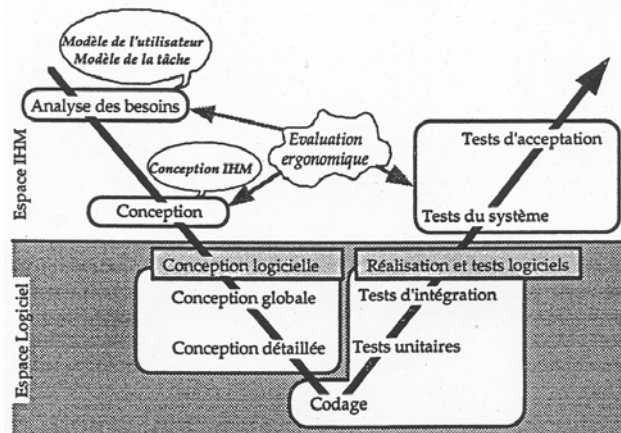
## Méthode classique en cascade



## Approche incrémentale



## Le modèle en V



## Une seule règle essentielle

Il faut  
faire participer  
l'utilisateur  
dans toutes les phases amont :

*Participatory design*

## Modèles conceptuels

- Représentation mentale qui dépend de la connaissance acquise (évolue avec expérience)
- 3 modèles conceptuels :
  - de l'utilisateur à propos du système
  - du concepteur à propos du système et de l'utilisateur
  - du système à propos de l'utilisateur
- ↪ Correspondance entre :  
les variables psychologiques de l'utilisateur et  
les variables physiques du système

## Modèles conceptuels

- **Accomplissement d'une tâche**

- Etablir un but (variables psychologiques)
- Formation d'une intention (variables physiques)
- Spécification d'une suite d'actions
- Exécution des actions
- Perception de l'état du système
- Interprétation
- Evaluation

↪ **Objectif** : réduire les distances mentales entre les modèles (Théorie de l'action de Norman)

## Méthodologie de conception

1. Définir le modèle de l'utilisateur
2. Définir le modèle de la tâche
3. Définir le modèle de l'interaction
4. Définir les concepts informatiques

↪ Au moment de la **spécification détaillée**, on doit disposer :

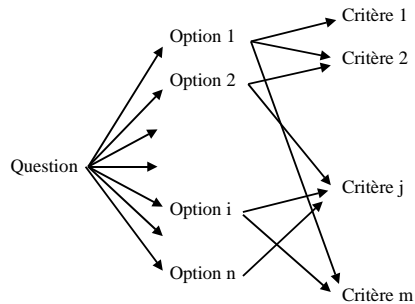
- Des 3 modèles
- De scénarios d'utilisation (*use case*)

## Conseil

- **Il faut documenter les décisions de conception**

↳ Notation QOC (Question Option Critère)

- traçabilité
- aide à la prise de décision (pondérations)



## 1. Définir le modèle de l'utilisateur

- **Objectifs :**

- Identifier les caractéristiques pertinentes de(s) l'utilisateur(s)
- Réduire les distances d'exécution et d'interprétation

- **Pour cela :**

- Parler aux utilisateurs n'est pas un luxe, c'est une nécessité
- Il faut se concentrer sur l'utilisateur le plus tôt possible et de façon continue
- Il faut faire participer l'utilisateur à la conception

## 1. Définir le modèle de l'utilisateur

- **Problèmes :**

- Choix des utilisateurs représentatifs
- Accès aux utilisateurs
- Omission du contexte d'utilisation

- **Représentation :**

- Données générales biométriques et sociales
- Connaissance dans le domaine de l'application et en informatique

## 1. Définir le modèle de l'utilisateur

- **Techniques :**

- Classification générale si utilisateur non disponible
- Interview,
- Questionnaire (6 à 12 utilisateurs représentatifs)
  - ⌘ Degré de confiance, "je ne sais pas"
  - ⌘ Poser plusieurs fois la même question de façon différente pour comparer
  - ⌘ Eviter les réponses prévisibles

## 2. Définir le modèle de la tâche

- **Tâche** : But + Procédure pour l'atteindre
- **Procédure** : ensemble de sous-tâches liées par des relations de composition et des relations temporelles
- **Tâche élémentaire** : tâche décomposable en actions physiques (opération d'E/S)

## 2. Définir le modèle de la tâche

- **Méthodes** :
  - Observation directe (enregistrement avec 2 observateurs) en laboratoire ou sur le terrain
  - Entretien (ouvert, dirigé, incident critique)
  - Groupe de travail : rassembler utilisateurs et concepteurs (brainstorming, scénario)

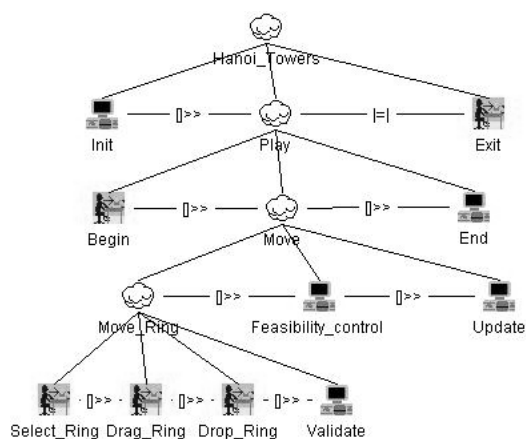
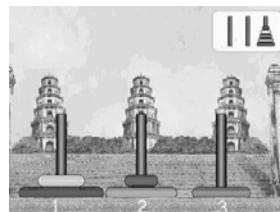
## 2. Définir le modèle de la tâche

- Résultats attendus :

- Identifier les variables psychologiques
- Construire la hiérarchie de tâches jusqu'aux tâches élémentaires
- Décorer chaque tâche : préconditions, postconditions, fréquence, complexité, criticité, acteur responsable (utilisateur ou système) ...
- Penser aux exceptions
- Evaluation avec utilisateur

## 2. Définir le modèle de la tâche

- Exemple d'outil CTTE



### 3. Définir le modèle d'interaction

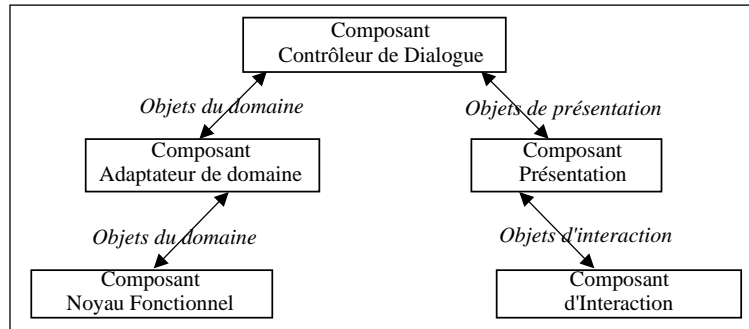
- Choix d'un environnement technique et d'un style d'interaction associé
- Représentation des interfaces (storyboard)
- Définition pour chaque tâche élémentaire : action + feedback

### 4. Définir des concepts informatiques

- **Objectif :**
  - Etablir correspondance entre les concepts du domaine et les concepts informatiques
  - Distinguer parmi les objets informatiques ceux qui dépendent du domaine et ceux qui sont de l'IHM
  - Construction d'une architecture logicielle

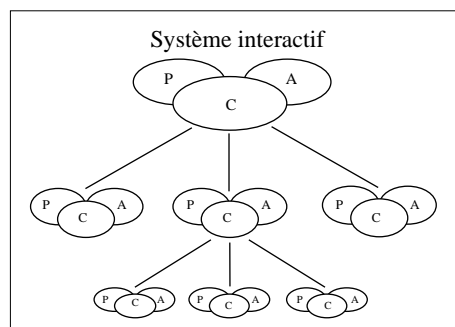
## Modèles en couches

- Ex: *Arch*



## Modèles multi-agents

- Ex : *PAC* ou *MVC*



## Problème de mise en œuvre

- Adéquation parfois difficile entre un modèle théorique et les outils de développement
- Ex 1 : les outils de présentation type CD-ROM interactif sont très performants sur le plan de la présentation et du contrôle mais sont souvent assez pauvres sur le plan de l'abstraction (Director et Toolbook)
  - ↪ Dissémination de l'abstraction dans la présentation

## Problème de mise en œuvre

- Ex 2 : les outils de programmation d'applications interactives complexes sont très performants sur le plan de l'abstraction, mais le contrôle est laissé à la discrétion du programmeur et il faut au moins 6 mois d'expérience pour maîtriser les boîtes à outils de présentation (type Visual C++, Borland C++, Java)
  - ↪ Apprentissage difficile
  - ↪ Vitesse de développement limitée

## Problème de mise en œuvre

- Ex 3 : les outils de programmation d'applications interactives simples constituent généralement un bon compromis (Visual Basic, Delphi)
  - ↪ Il faut s'imposer des règles de programmation pour ne pas tomber dans la facilité qui nuit à la maintenance et à l'efficacité
  - ↪ Utiliser programmation orientée objet chaque fois que c'est possible
  - ↪ Découpage en modules clairement identifiés

## IHM multi-cibles

## Problématique

- IHM multicibles pour applications interactives car :
  - ↪ Besoin pour les utilisateurs d'interfaces adaptées à leurs situations et à leurs personnalités,
  - ↪ Multiplication des contextes de travail (bureau, nomade, connecté, non connecté, ...),
  - ↪ Apparition de nouveaux supports, avec des caractéristiques spécifiques.

## L'adaptation multi-supports

- Les langages de programmation (C++, JAVA, langage spécifique, ...),
- Les systèmes d'exploitation (Windows, Unix, palm-OS, Symbian, openTV, ... ),
- Caractéristiques des supports ( taille de l'écran, couleurs, moyens d'interaction, ...).



## Objectif

- Limiter les coûts de développement et de maintenance en proposant un **processus de génération d'IHM multicibles pour applications interactives** le plus automatique possible.

## La Plasticité

- Définition

Capacité d'une interface à s'adapter aux contraintes matérielles et environnementales dans le respect de son utilisabilité.  
(Thévenin, 1999)

## Solutions actuelles

Plusieurs Approches:

- ↳ Migration ou traduction des interfaces,
- ↳ Méta-langage (XIML, UIML,...),
- ↳ Approches à base de modèles.  
(Modèle des tâches, modèle de navigation,..)

## Grille d'analyse

CIBLE			CONCEPTION																					
U	P	E	Spécification de modèles													Type		Transformation						
			Niv		Initiaux					Trans. et fin.			C	A	T	Niv abstr.			Props					
M	M	C	T	U	P	E	E	Tr	C	I	I	IF				C	A	T	C	I	I	IF	F	R
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	○	•	•	•	•	•	○	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	○	•	•	•	•	•	•	•	•	•	○	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	○	•	•

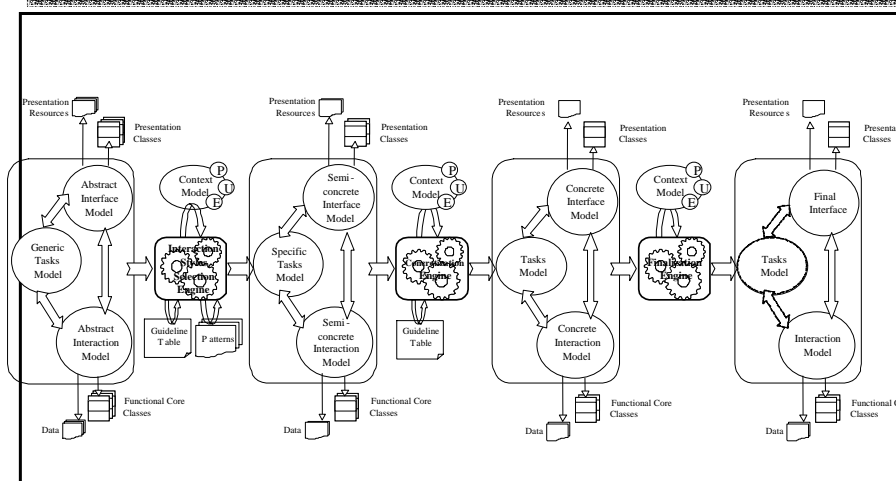
OUTILS	R	EXECUTION										
		Calcul			Mise en œuvre							
		R	R	M	Portée				Reprise			
	r	s		A	E	T	C	P	S	T	A	
ArtStudio												
AMF-Based approach			○	○		○	•					
EDPHA		○		○								
CBME		○	•	-	?	○	?	?	?	?	?	?

• Couvert  
 ○ Partiellement couvert  
 - Non couvert  
 ? Sans avis.

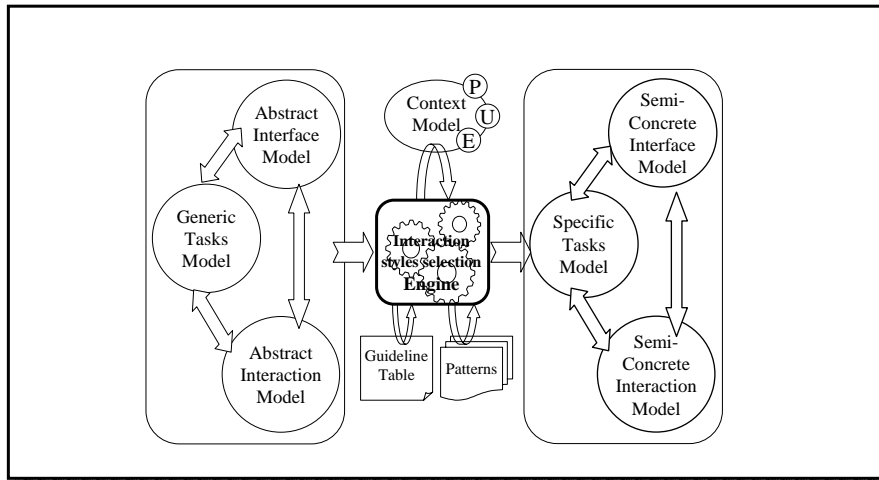
## • Légende

<b>Outils</b>	Nom de l'approche. Exemple : Vaquita, Teresa, ArtStudio
<b>Cible</b>	La cible de l'adaptation : U (Utilisateur) P (Plate-forme) E (Environnement)
<b>Conception</b>	Concerner les outils de production d'IHM. Ils sont à caractériser selon deux volets : la spécification de modèles et la transformation de ces modèles.
<b>Spécification</b>	- Préciser le niveau d'abstraction traité : modèles (M) / méta-modèles (MM) - Préciser la portée de la spécification en termes des modèles considérés : Modèles initiaux : C (Concepts), T (Tâches), U (Utilisateur), P (Plate-forme), E (Environnement), Ev (Evolution), Tr (Transition) Transitoires et finaux : CT (Concepts-Tâches), IA (Interface abstraite), IC (Interface concrète), IF (interface finale)
<b>Transformation</b>	- Préciser le type de la transformation : concrétisation (C), abstraction (A), traduction (T) - Préciser le niveau d'abstraction auquel s'applique la transformation : CT, IA, IC, IF - Préciser les propriétés couvertes : factorisation de modèles (F), traçabilité des choix de conception (R pour Rationale), conservation de l'information (Co), capitalisation des modèles (Ca)
<b>Exécution</b>	Caractériser la contribution de l'outil à l'exécution. Trois volets sont considérés : la reconnaissance du contexte d'usage, le calcul et la mise en œuvre de la réaction en cas de changement de contexte d'usage.
<b>Reconnaissance (R)</b>	La reconnaissance couvre la capture et l'identification du contexte d'usage
<b>Calcul (Calc)</b>	Préciser le type de réaction couvert : - remodelage par reconfiguration de composant (Rr) ou substitution de composants (Rs) - redistribution ou migration (M)
<b>Mise en œuvre</b>	- Préciser si l'outil suppose des IHM précalculées (P) - Préciser la portée de l'adaptation : le système interactif complet (A pour Application), un espace de travail (ET), un concept (C) ou relève-t-elle du niveau pixels (P) ? - Préciser le grain de reprise de l'interaction : session (S), tâche composée ou élémentaire (T), action physique (A)

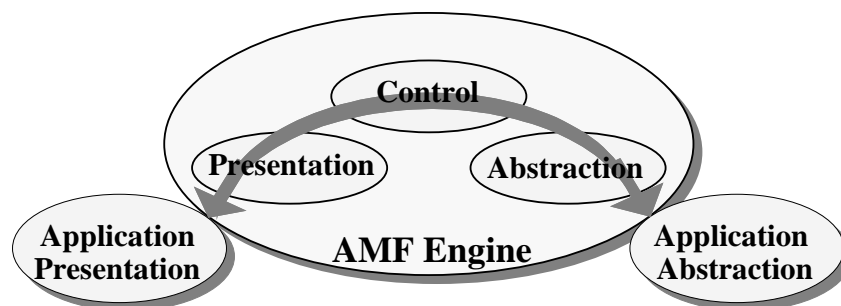
## A generation process in 4 steps



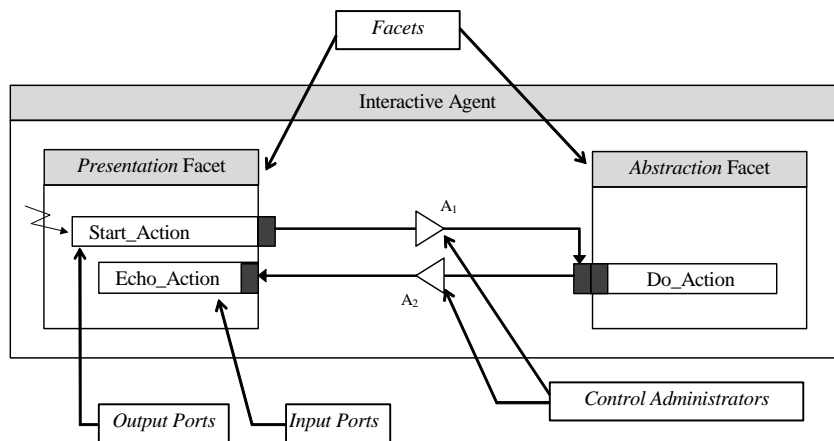
## Focus on the 2 first steps



## The interaction model : AMF



# AMF Basics



# AMF in XML

```

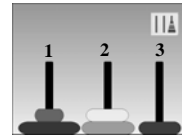
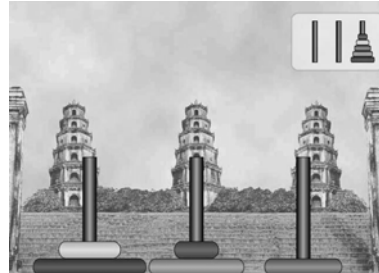
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Agent (Agent*, Administrator+, Facet+)>
<!-- ATTLIST Agent
  Name CDATA #REQUIRED
  Sub-agent CDATA #REQUIRED
  Type CDATA #REQUIRED
-->
<!-- ELEMENT Facet (Port+)
  -->
<!-- ATTLIST Facet
  Name CDATA #REQUIRED
  Type CDATA #REQUIRED
-->
<!-- ELEMENT Port EMPTY
  -->
<!-- ATTLIST Port
  Name CDATA #IMPLIED
  Type CDATA #REQUIRED
  TypeIO (2 | i | o) #REQUIRED
  TypeAC (Abstract | Concrete) #REQUIRED
  DaemonName CDATA #REQUIRED
  FacetName CDATA #REQUIRED
-->
<!-- ELEMENT Administrator (Sources+, Targets+)
  -->
<!-- ATTLIST Administrator
  Name CDATA #REQUIRED
  Type (Simple | Return | Filter | ReturnFilter | Sequence) #REQUIRED
  TypeAC (Abstract | Concrete) #REQUIRED
-->
<!-- ELEMENT Targets EMPTY
  -->
<!-- ATTLIST Targets
  Name CDATA #REQUIRED
-->
<!-- ELEMENT Sources EMPTY
  -->
<!-- ATTLIST Sources
  Name CDATA #REQUIRED
-->

```

# Sample

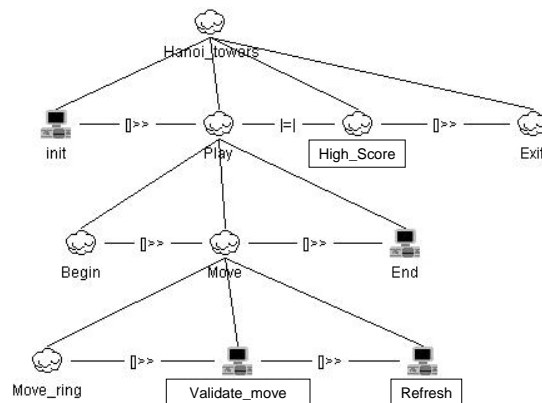
## Towers of Hanoi

- Move ring on stems.
- Rule : no ring on a smaller one.
- Move interaction :
  - ↳ Direct manipulation
  - ↳ Keyboard selection
  - ↳ Remote controller
  - ↳ ...



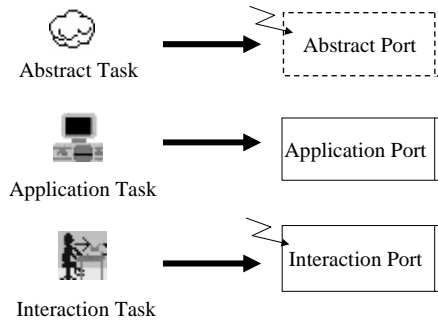
# Abstract models

## CTT Task model



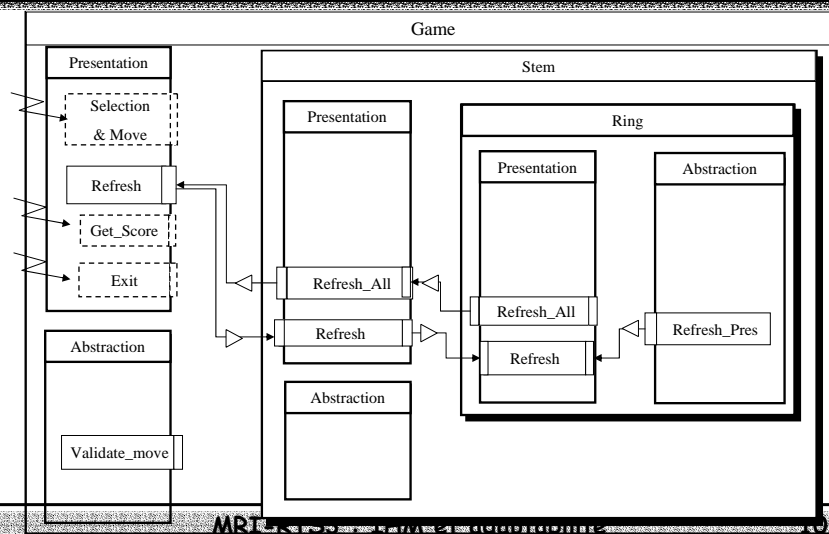
# Abstract models

Relationship between the task and the interaction models.



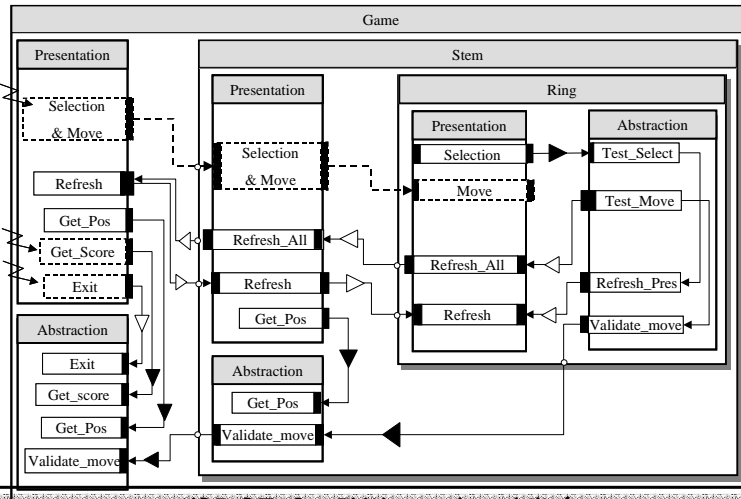
# Abstract Models

Interaction Model



# Abstract Models

## Abstract Interaction Model



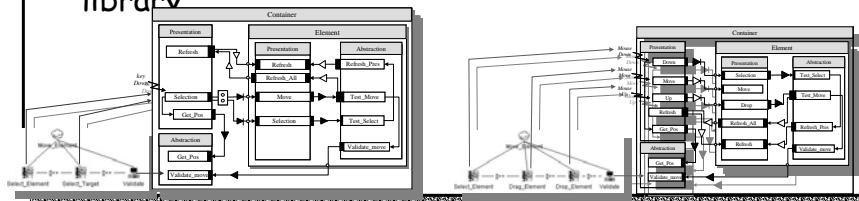
# Abstract Models

## Interface Model

- Static representation of possible resources
- Currently, our own basic representation
- Want to re-use (extend?) one of the XML-based language (UIML, XIIML, USIXML...)

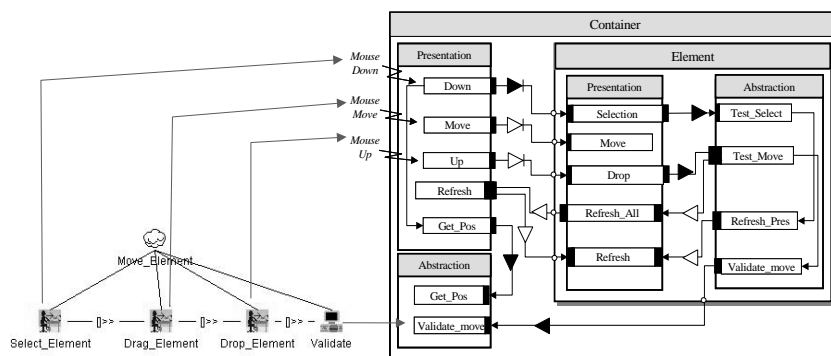
# Concretization

- Use Context <P, U, E>
  - ↳ Large screen, Mouse,
  - ↳ Adult user,
  - ↳ No network.
- First step : Select patterns of interaction in a library



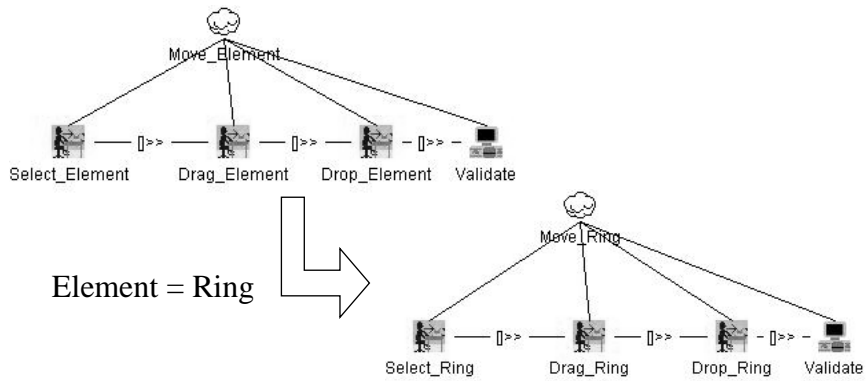
# Concretization process

Use of task patterns coupled to interaction patterns



The select and move pattern using a pointing device

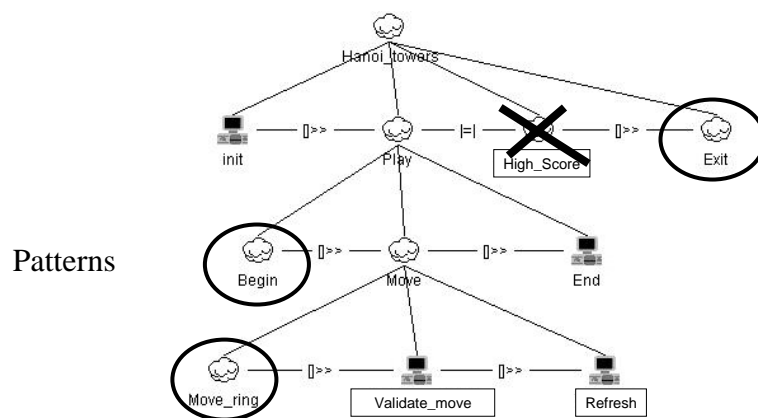
## Tasks Concretization



MRI-RTS3 : IHM et adaptabilité

110

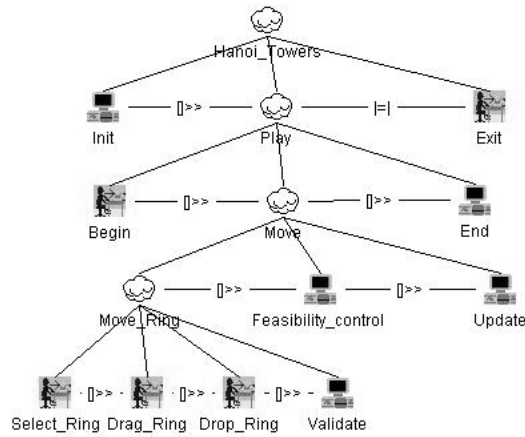
## Tasks Concretization



MRI-RTS3 : IHM et adaptabilité

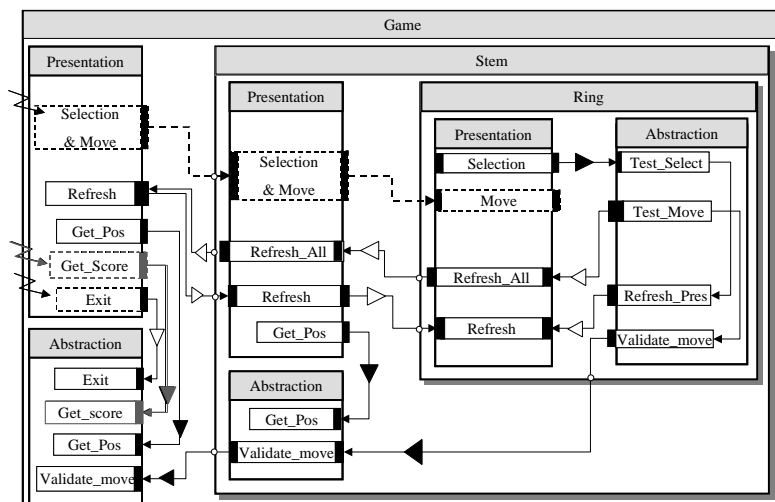
111

# Tasks Concretization



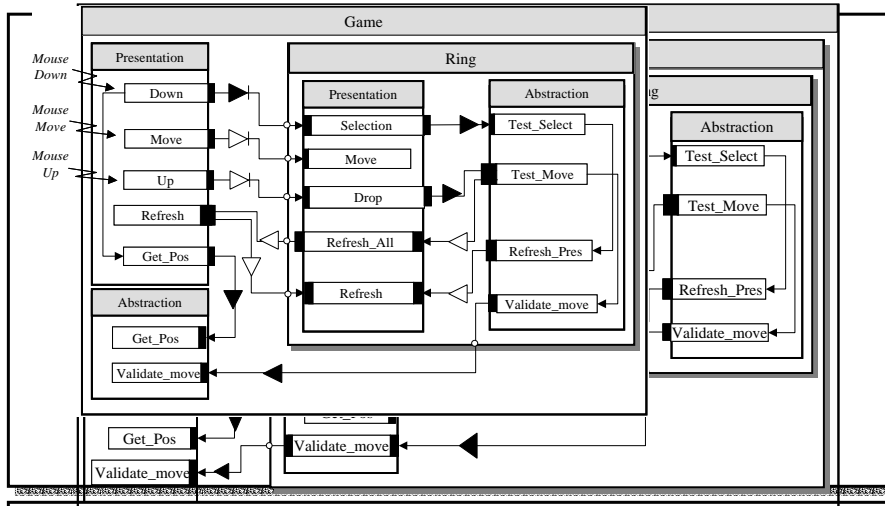
# Concretization

Filtering and enrichment of the interaction model



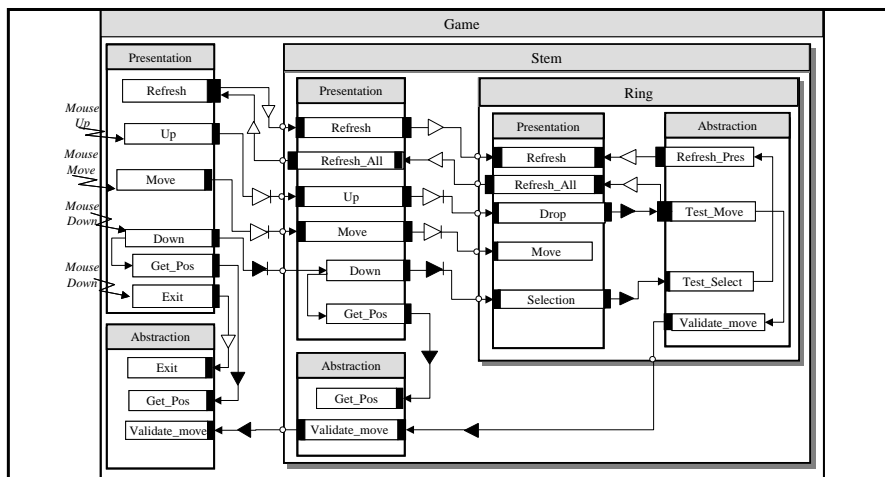
# Concretization

## Patterns insertion



# Concretization

## Patterns insertion



# Run-time

