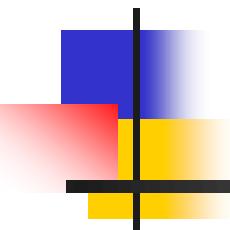
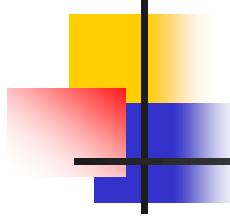


Approches à base de Modèles et Interaction Homme Machine



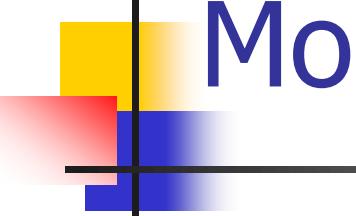
David Navarre
navarre@irit.fr - <http://www.irit.fr/~David.Navarre/>

ICS - Interactive Critical Systems
IRIT



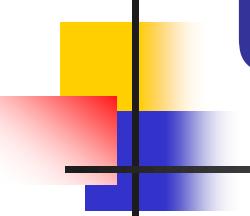
Pourquoi les approches Model-Based ?

- Mettre en avant les informations abstraites
- Mettre en avant les informations importantes
- Aider à gérer la complexité
- Élément important de la plupart des méthodes de conception
- Possibilité de raisonner sur des modèles



Modèles Importants en HCI

- Modèles de tâche
- Modèles de la cognition
- Modèles de l'utilisateur
- Modèles du domaine
- Modèles du contexte
- Modèles de présentation
- Modèles de dialogue

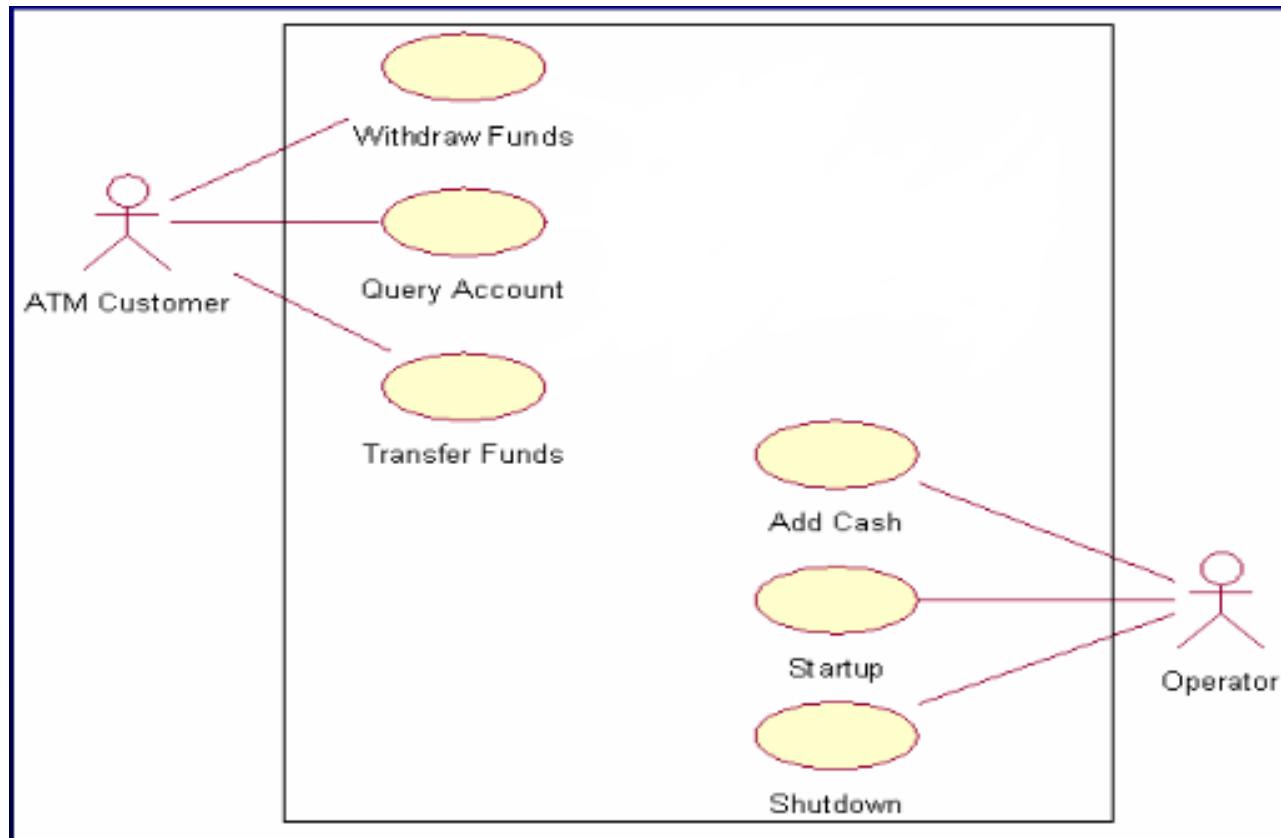


UML et HCI

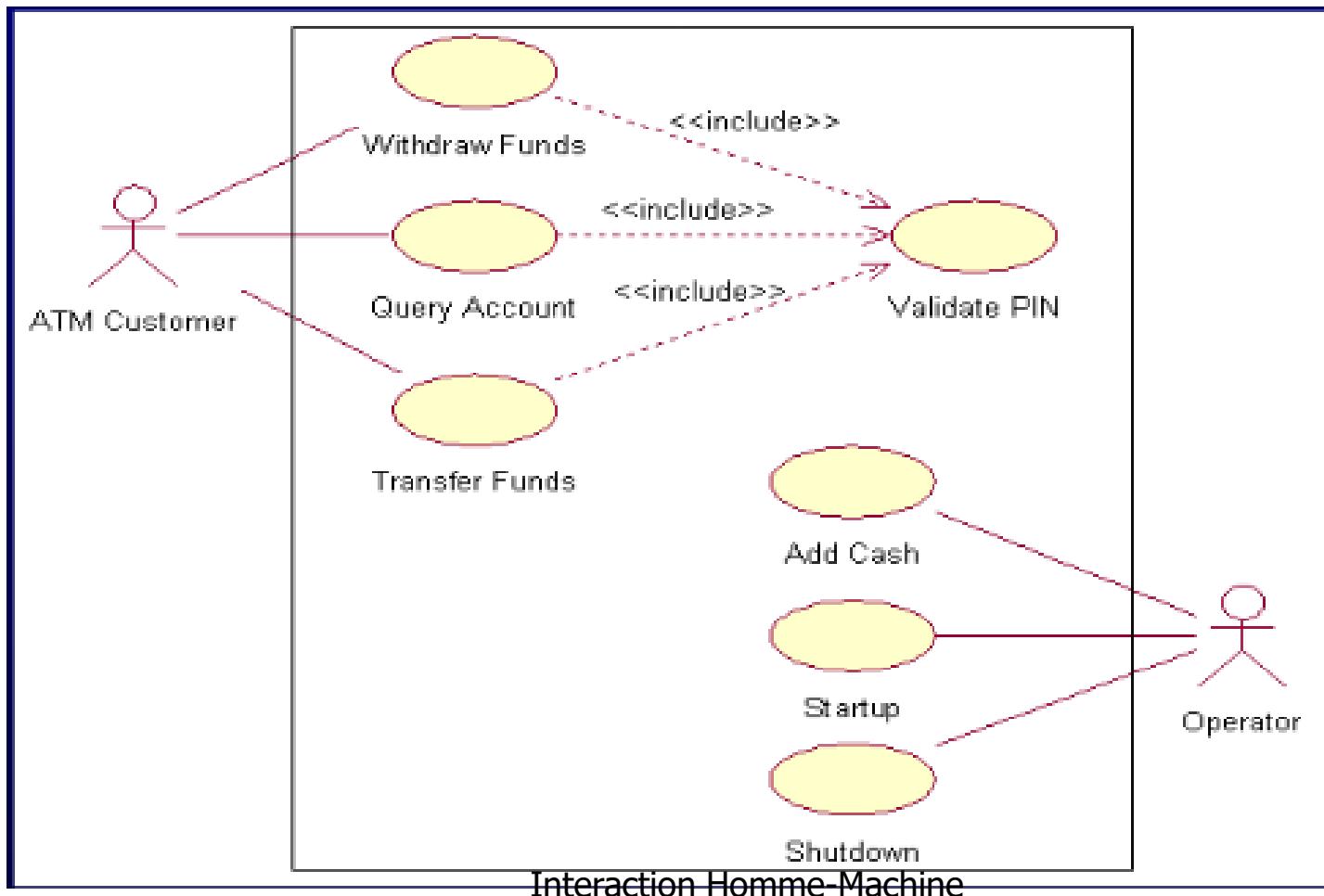
- Approche à base de modèle
- Neuf notations
- Pas de prise en compte de l'interface utilisateur (un peu à partir des Use cases)
- Comment améliorer cela ?

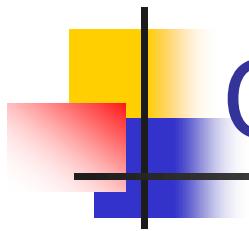
- Raffiner les Use Cases en modèles de tâches ?

Use Cases et tâches (1/2)



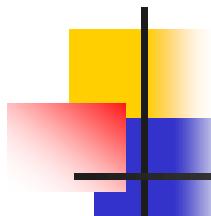
Use Cases et tâches (2/2)





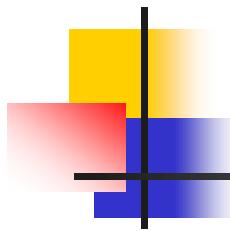
Qu'est-ce que l'analyse de tâches?

- Des méthodes pour déterminer
 - Ce que font les utilisateurs
 - Les "outils" avec lesquels ils travaillent
 - Les "informations" qu'ils savent ou doivent savoir sur leur travail
 - Les erreurs qu'ils risquent de commettre
- De façon complète et non-ambiguë



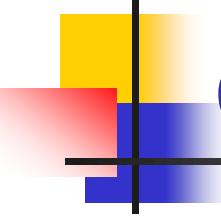
Un exemple

- Pour nettoyer la maison
 - Sortir l'aspirateur
 - Brancher l'aspirateur
 - Nettoyer les pièces
 - Quand le sac est plein, le vider
 - Ranger l'aspirateur
- Il faut savoir
 - Où est l'aspirateur
 - Où sont les prises
 - Comment vider le sac et où sont les sacs vides
 - Évaluer si une pièce est propre ou sale
 - ...



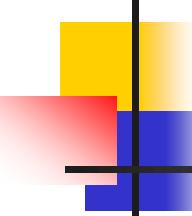
Recueil d'information (analyse)

- Travail des opérateurs/utilisateurs
 - Procédure prévue (règlementation)
 - Procédure effective (observée – variabilité)
 - Procédure minimale (optimale)
- Analyse des éléments du poste de travail
 - Post-it, pense-bête, ...
 - Annotations
 - Autres outils non reliés,...



Attention aux niveaux d'abstraction

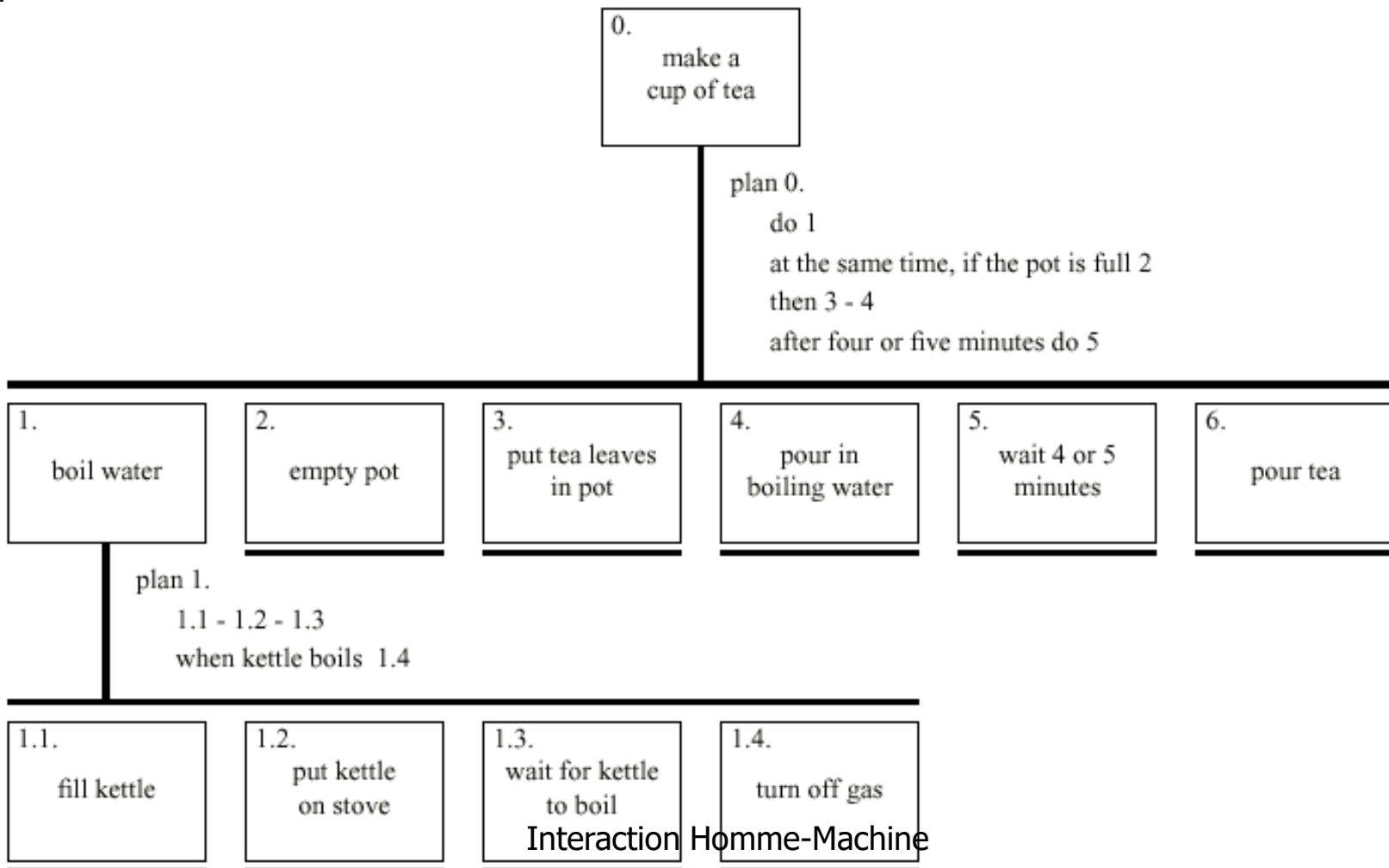
- Si vous posez la question suivante:
Qu'êtes-vous en train de faire ?
- Vous pouvez récupérer:
 - Je tape ctrl-B
 - Je met "bonjour" en gras
 - Je met un mot en gras
 - Je met en évidence un mot
 - Je modifie un document
 - Je tape une lettre
 - Je garde le contact avec ma famille
 - J'attends un coup de fil ??

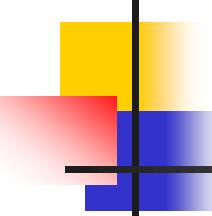


Construction de l'arbre des tâches

- 1 Faire une liste des tâches
- 2 Regrouper les tâches de haut niveau
- 3 Décomposer les tâches de bas niveau
- Où s'arrêter ?
 - Est-ce que "vider le sac" est assez décomposé?
 - Objectifs: ne décomposer que les tâches pertinentes (objectif optimisation/automatisation du travail)
 - Actions utilisateur: cognitives/motrices souvent peu intéressantes à développer (évaluation de performance)
 - Actions système dans d'autres modèles
- En haut = reproduction de l'espèce
- En bas = vie biologique (respirer, boire un café, ...)

Exemple faire une tasse de thé

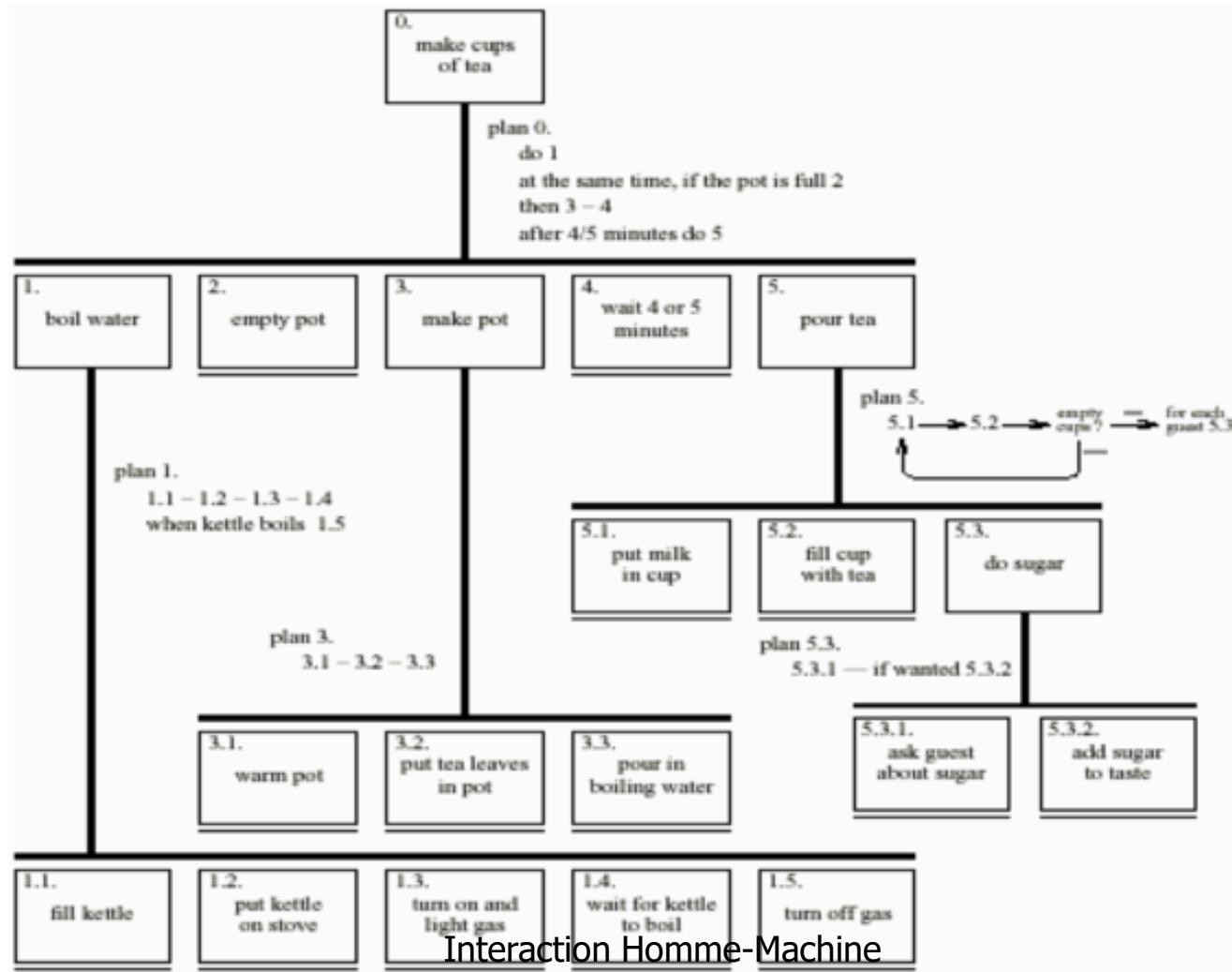


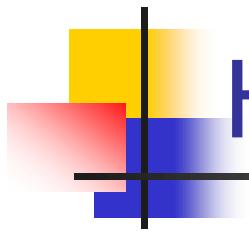


Raffinement des modèles

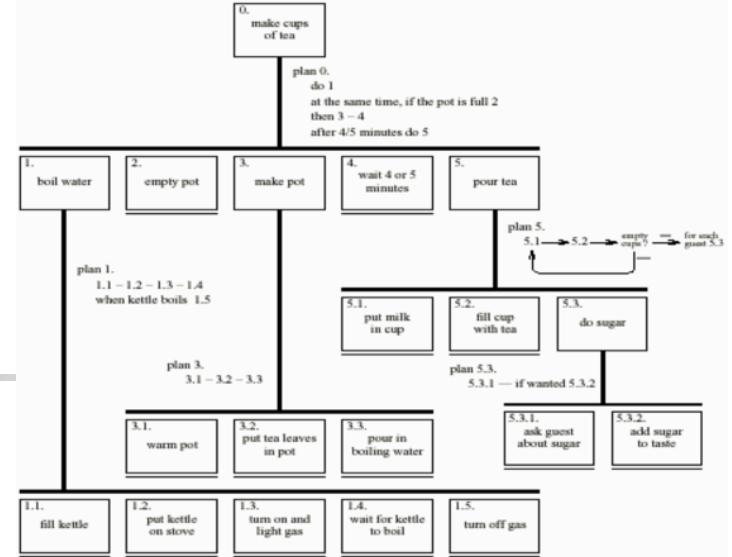
- Une fois un modèle réalisé (peu importe la notation)
 - Comment s'assurer qu'il est correct ?
 - Comment l'améliorer ?
- Quelques heuristiques
 - Actions couplées: e.g., où est "turn gas on"?
 - Restructurer : e.g. rendre une tâche générique `make pot'
 - Balance : e.g. est-ce que `pour tea' est plus simple que "make pot"?
 - Généraliser: e.g. faire une tasse ... en faire plusieurs

Exemple: faire une tasse de thé

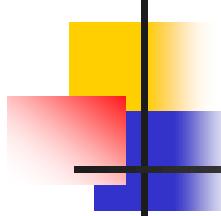




HTA: Limitations



- Pas de description des informations
- Comportement décrit de façon textuelle (référence à l'arbre difficile)
- Comportement entre sous arbres ne peut être représenté
- Comportement décrit en langue naturelle (when, if, while, repeat, ...)

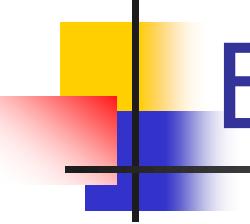


Définitions

- Buts
- Tâches
- Tâches élémentaires
- Analyse de tâches (task analysis)
- Modélisation des tâches (Task Model)
- Activité (analyse, modélisation, ...)

Example de l'analyse de la tâche

- Touriste
 - Réserver un billet de train
 - Chercher un monument dans une ville
 - Jeu des 15
- Client d'un distributeur de billets

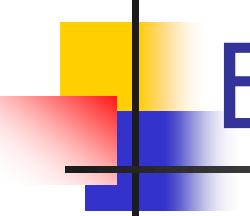


Etude de Cas: Le jeu de 15

- Le jeu se joue à 2
- Vous disposez des jetons 1, 2, 3, 4, 5, 6, 7, 8, 9
- On joue chacun son tour, chaque joueur prend un seul jeton à la fois, un jeton ne peut être pris qu'une fois
- Le premier qui possède 3 jetons (parmi ceux qu'il possède i.e. 3, 4 ou 5) dont la somme est exactement 15 a gagné

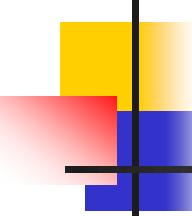
Devenez des experts

Jouez à ce jeu sans aucun outil (papier, crayon, ...)



Etude de cas jeu des 15

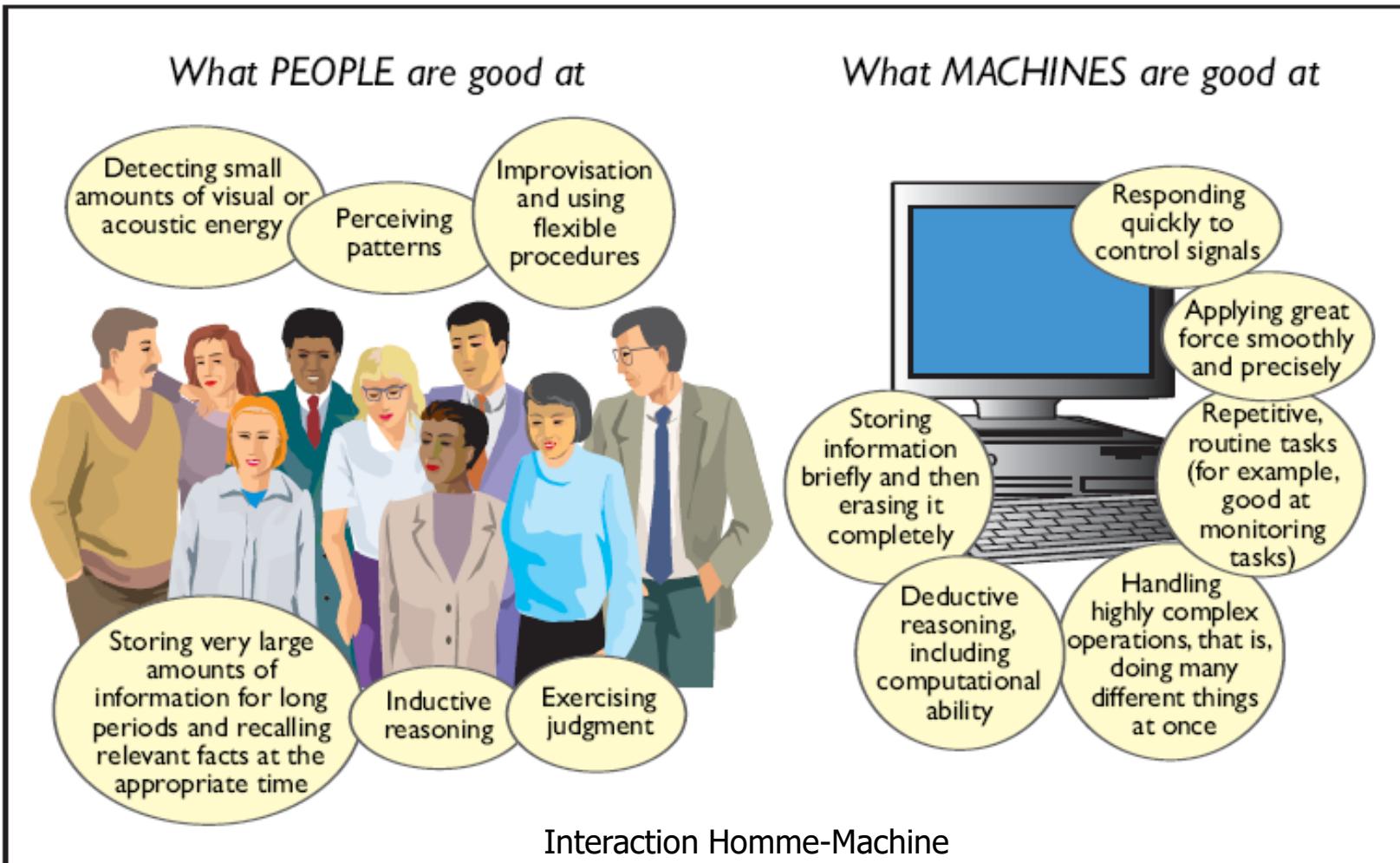
- Analyser la tâche d'un joueur
 - Proposer une décomposition en sous-tâches
- Proposer une interface (papier + crayon ou informatique) pour ce jeu
 - L'interface doit (bien entendu) assister au maximum la tâche du joueur



Attention !

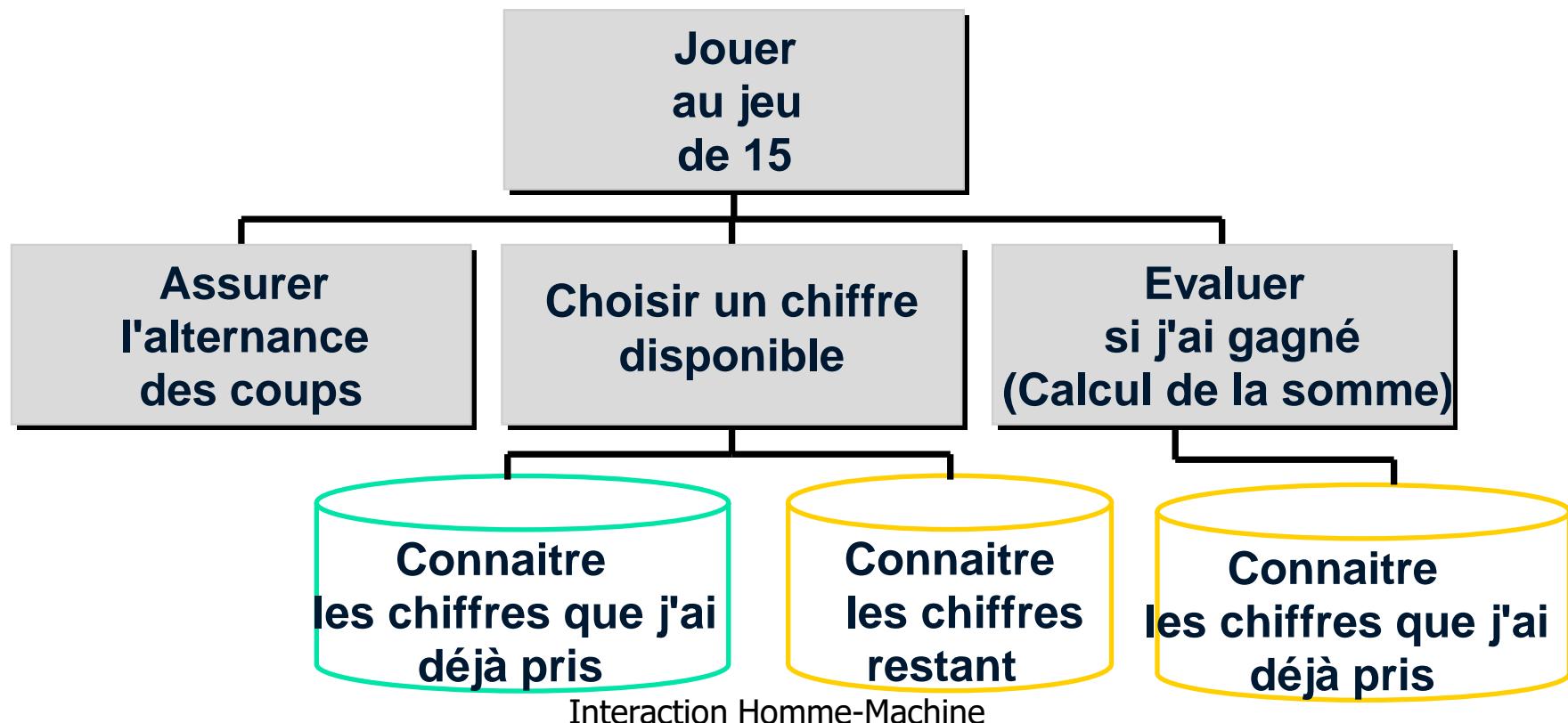
- Approche cognitiviste
- Pas d'inclusion du système
- Correction de chacun des modèles
- Pas de description algorithmique (calcul de si j'ai gagné)
- Pas de migration des tâches (le système calcule automatiquement)
- Qui commence

Carver Turoff March 2007/Vol. 50, No. 3 comm. of the acm (from Fitts 51)

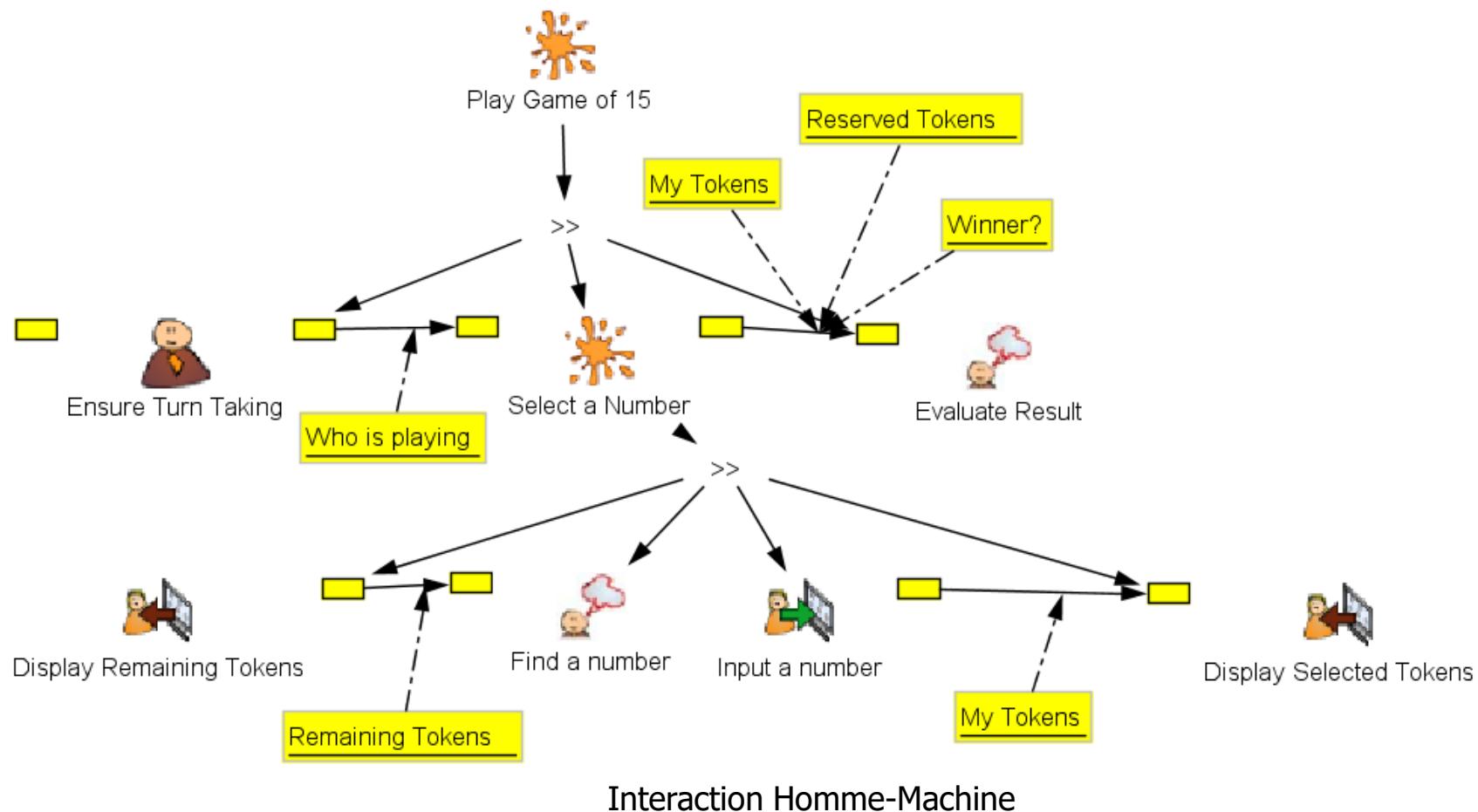


Analyse de la tâche

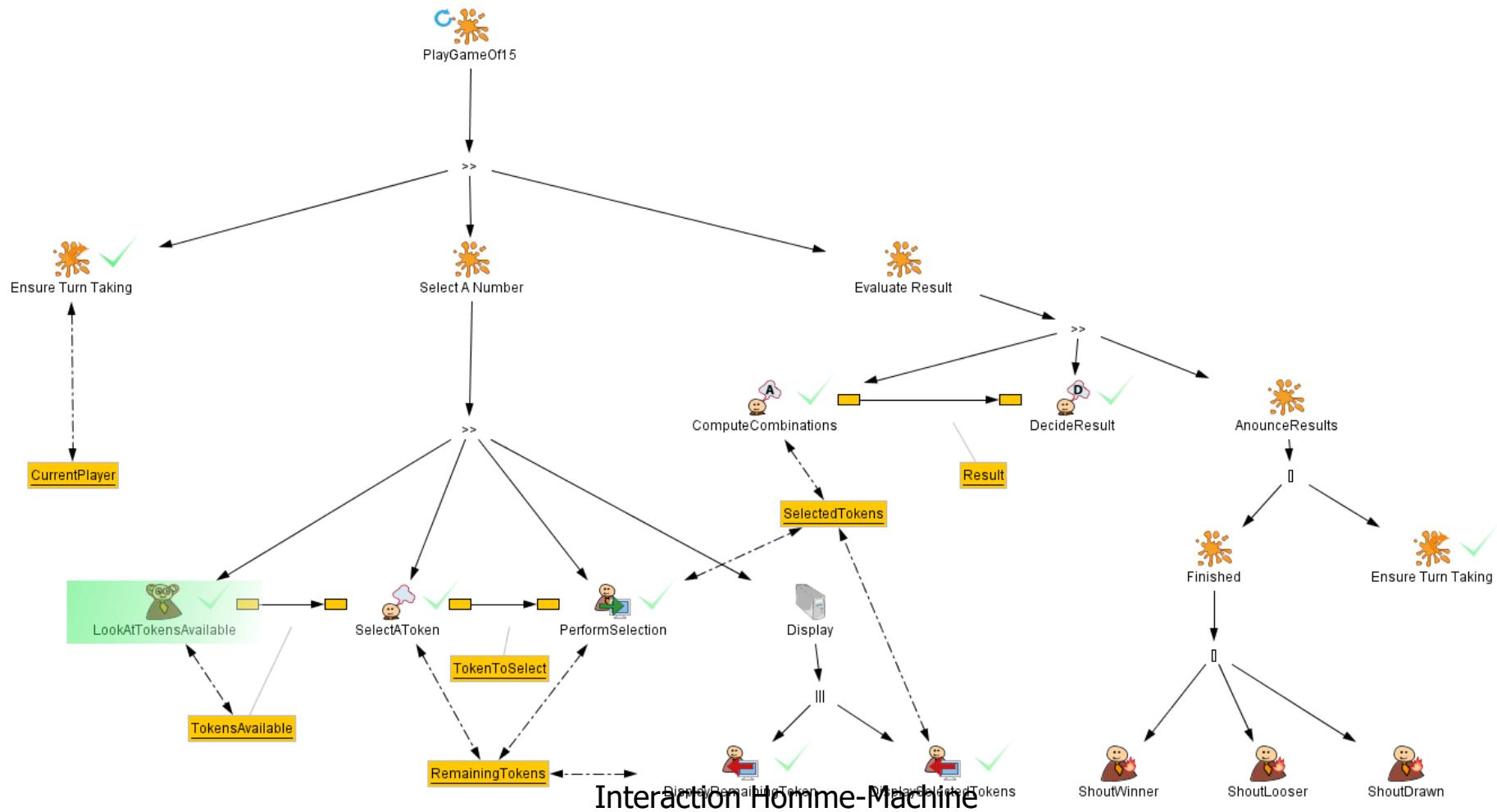
Hiérarchie des tâches et sous-tâches



Modélisation de tâche HAMSTERS



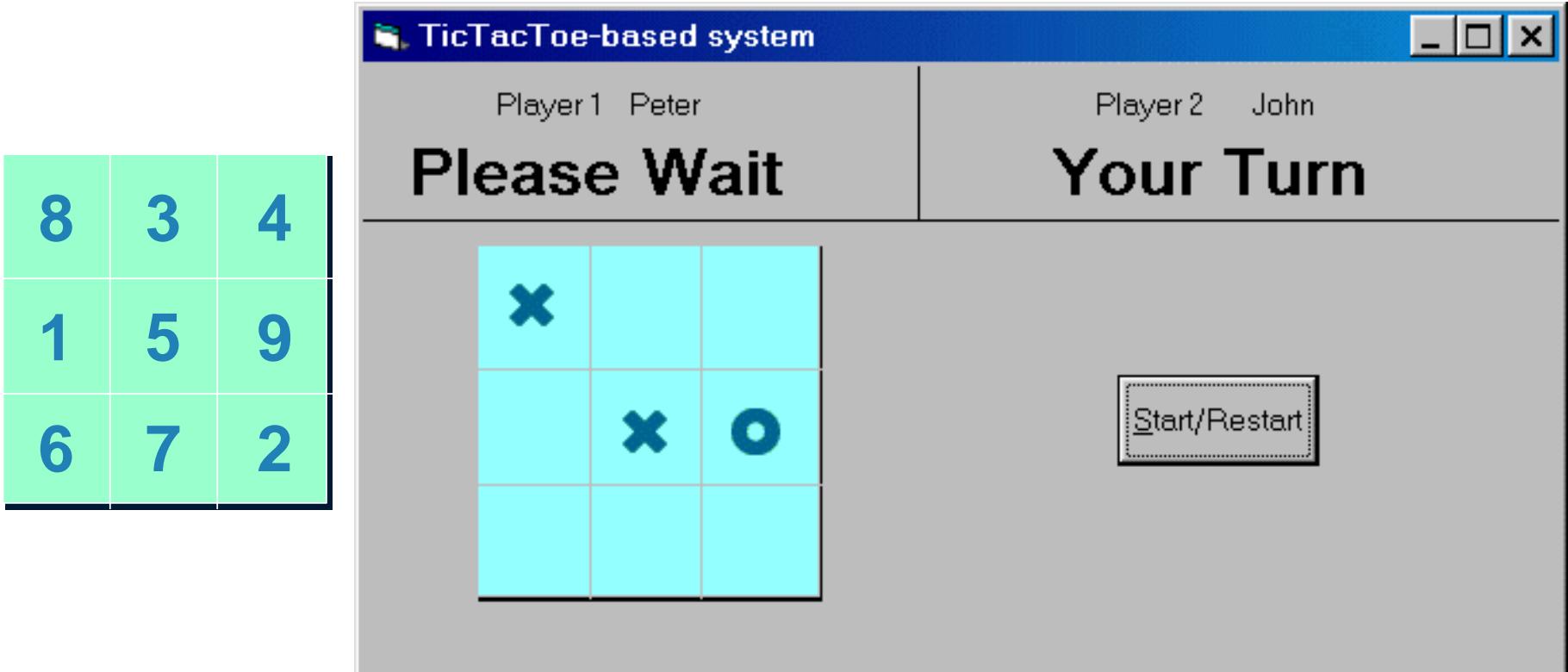
Modélisation de tâche HAMSTERS



Votre solution



Une autre solution

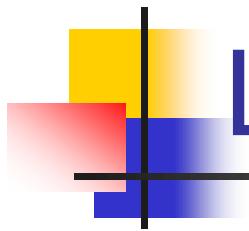


Logique de fonctionnement / d'utilisation

- Exemple d'un système informatique
 - On dispose d 'une base de données conçue suivant le schéma suivant :



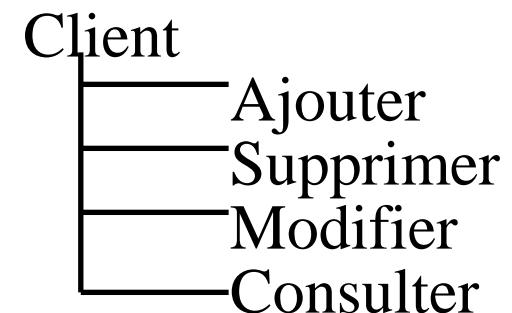
- Proposer une arborescence de menu permettant de manipuler les informations de cette base



Logique de fonctionnement

Vision de l'informaticien

L'arborescence de menus reflète les fonctions internes de la base de données

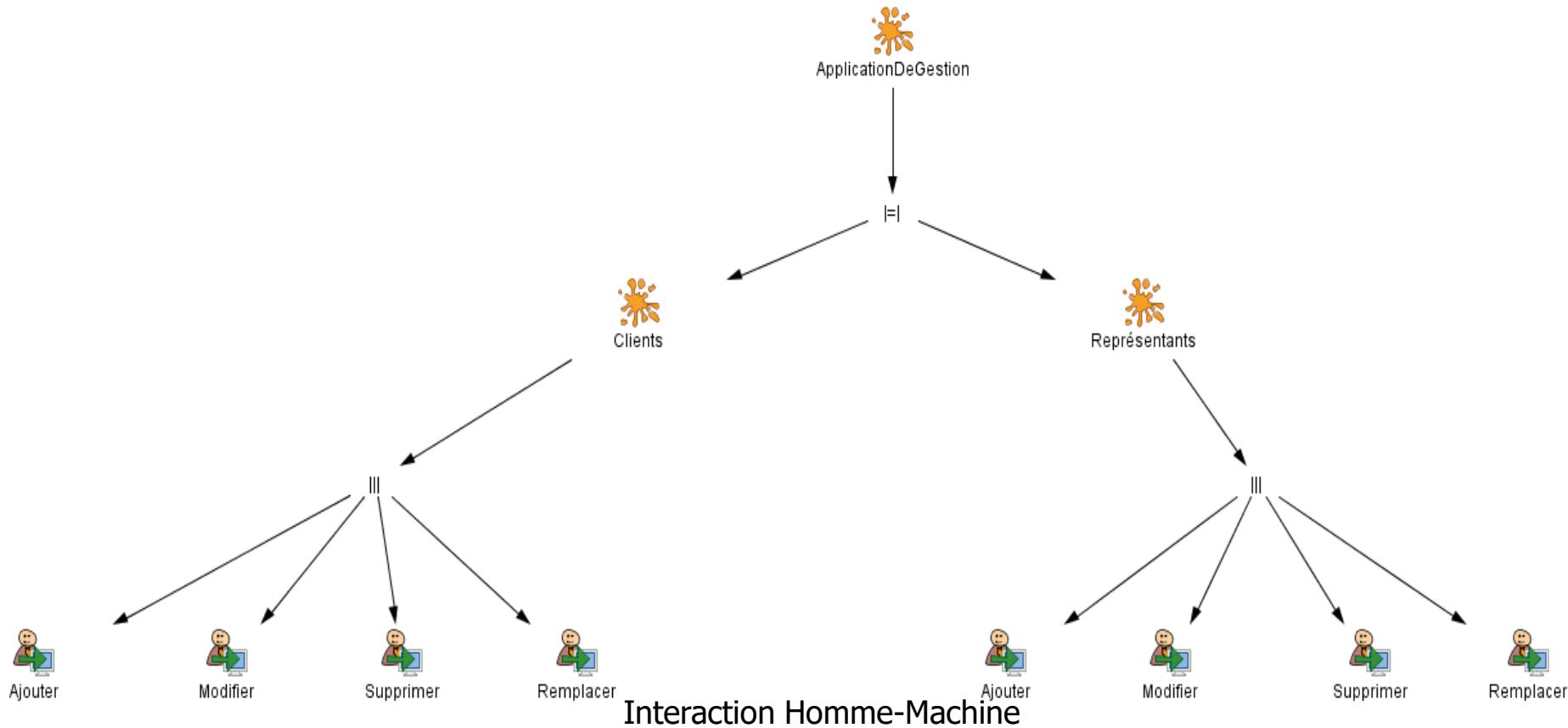


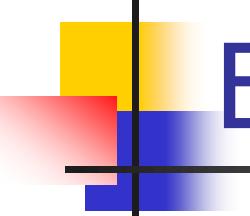
Représentant



Logique de fonctionnement

Vision fonctionnelle



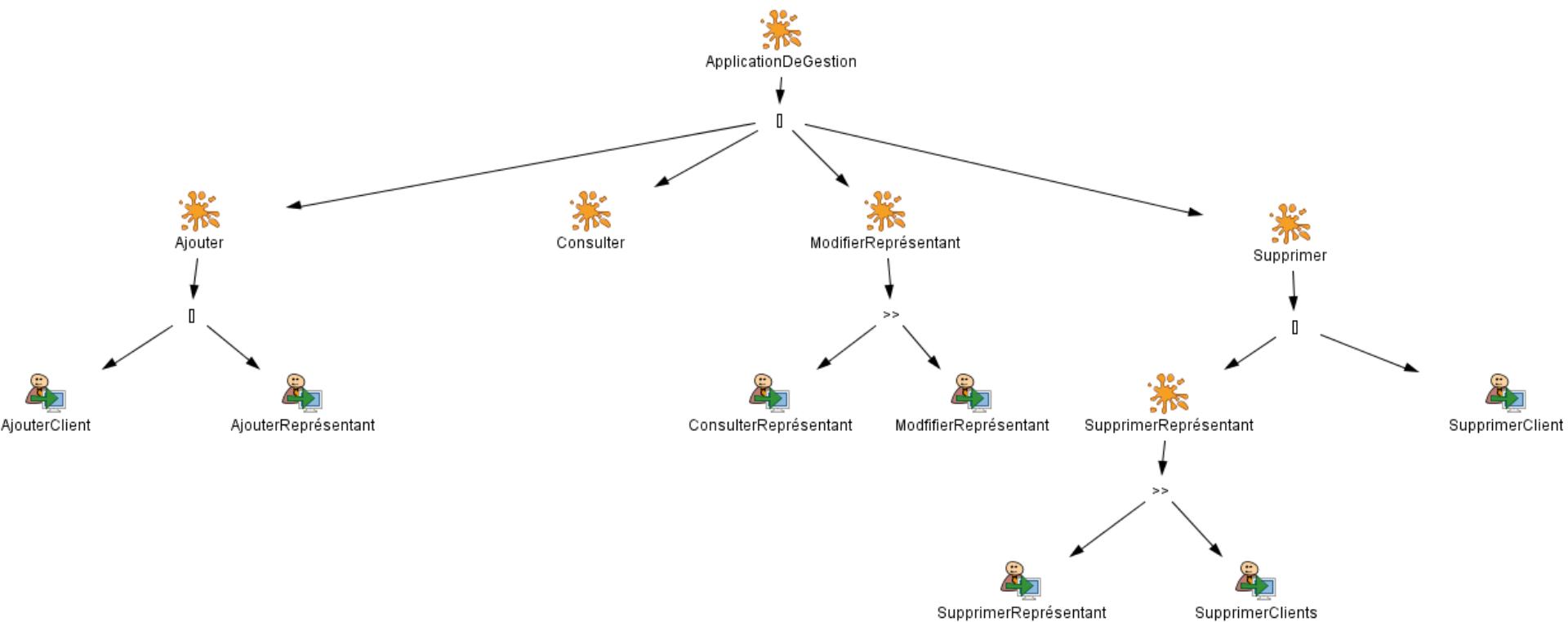


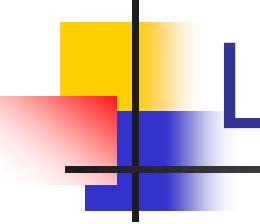
Exemple de tâche

- « Je passe mon temps à saisir des nouveaux clients dans la base de données, et je dois leur affecter le représentant le mieux adapté (en fonction de différents critères) ... Parfois je reçois un bon de commande envoyé par un représentant nouvellement recruté, je dois alors saisir à la fois le nouveau client et le nouveau représentant ... Il arrive que des représentants quittent la société, en général ils partent avec leurs clients, il faut faire du ménage dans la base de données ... »
 - ➔ Proposez une arborescence de menus adaptée à cette tâche.

Logique d'utilisation (1/3)

- Organiser le système d'après l'analyse de la tâche

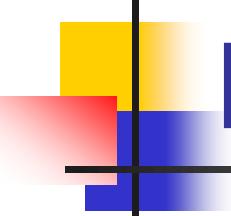




Logique d'utilisation (2/3)

- Il n'y a pas
 - De menu suppression représentant (sans les clients)
 - De modification de client
 - ...
- Il y a W. Buxton, BCS HCI 96
 - L'accès aux clients d'un représentant à partir de la suppression du représentant
 - Une consultation indépendante de client et de représentant
 - ...

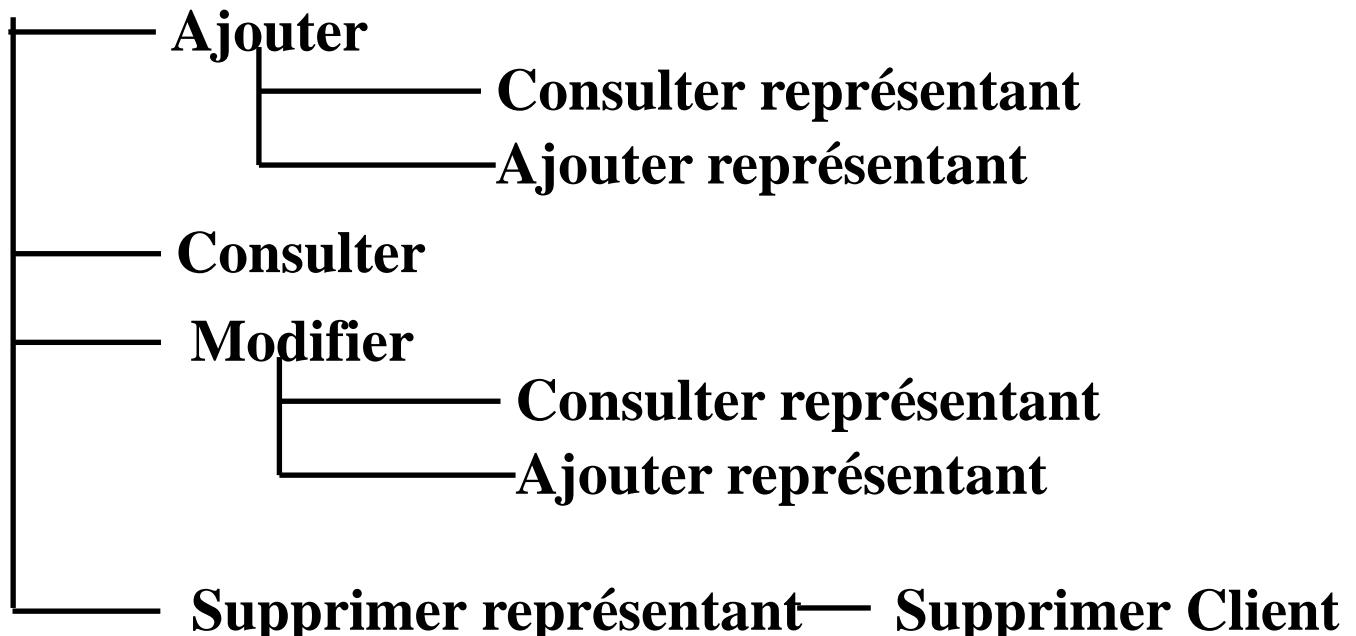
$$U=1/F$$



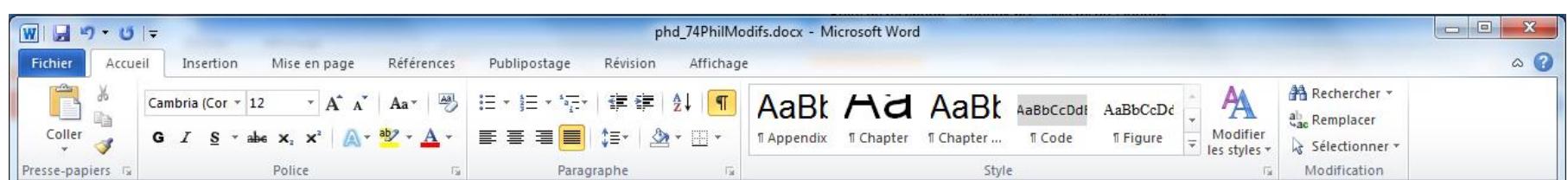
Logique d'utilisation (1/3)

- Organiser le système d'après l'analyse de la tâche

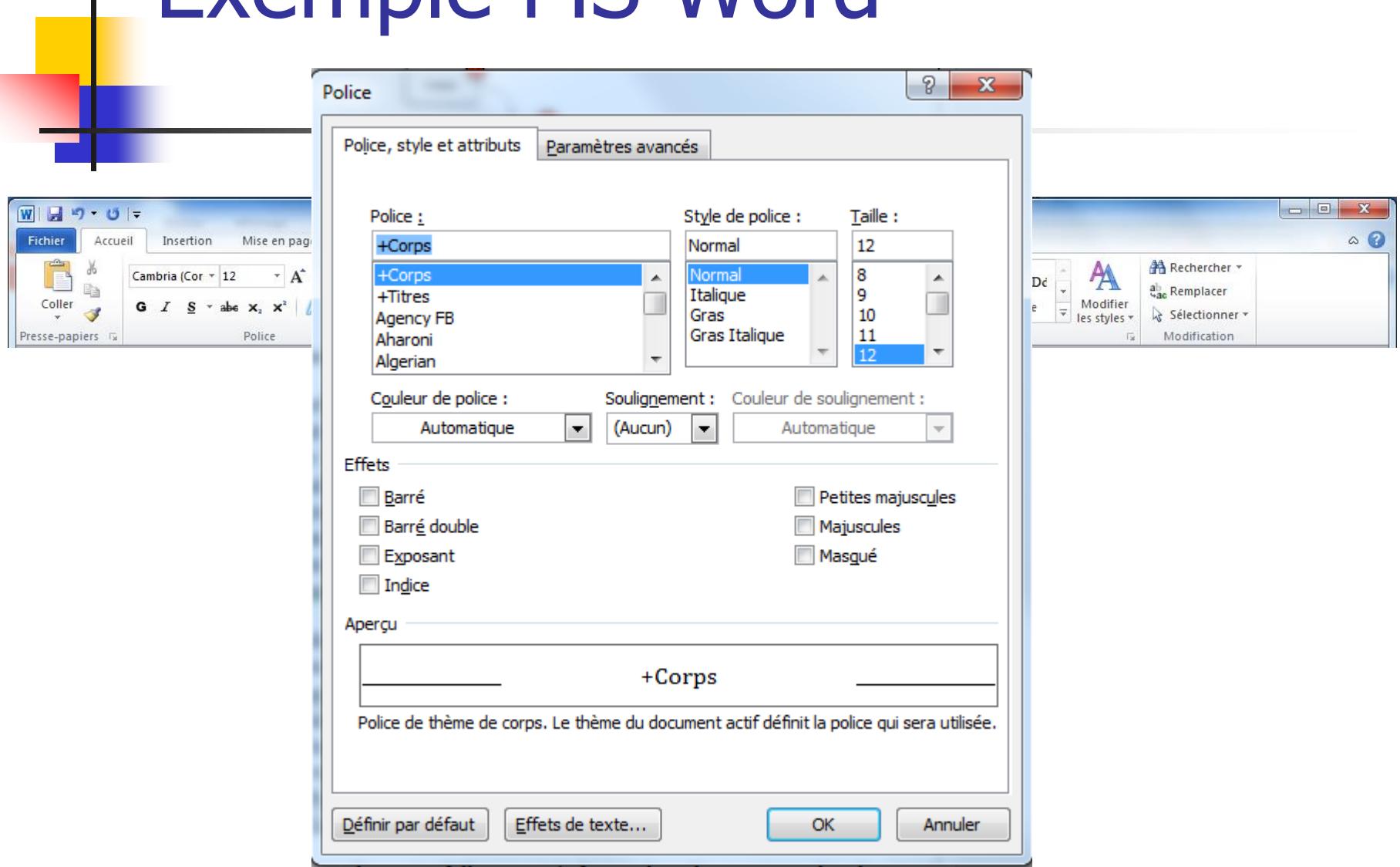
Gestion commerciale



Exemple MS Word



Exemple MS Word



Exemple MS Word

Créer un style à partir de la mise en forme

Propriétés

Nom : Style1

Type de style : Paragraphe

Style basé sur : Normal

Style du paragraphe suivant : Style1

Mise en forme

Cambria (Corps) 12 G I S Automatique

Paragraphe précédent Paragraphe précédent

Texte exemple Texte exemple

Paragraphe suivant Paragraphe suivant

Police : 12 pt, Français (France), Espace
Après : 0 pt, Style : Style rapide
Sur base de : Normal

Ajouter à la liste des styles rapides Mettre à jour automatiquement
 Uniquement dans ce document Nouveaux documents basés sur ce modèle

OK Annuler

Styles

Effacer tout

(Latin) Arial, 20 pt
(Latin) Arial, 36 pt
(Latin) Arial, Centré
(Latin) Cambria
(Latin) Verdana, 10 pt
(Latin) Verdana, 10 pt, Gra:
(Latin) Verdana, 10 pt, Gra:
(Latin) Verdana, 10 pt, Itali
(Latin) Verdana, 10 pt, Itali
(Latin) Verdana, 10 pt, Itali
(Latin) Verdana, 16 pt, Gra:
(Latin) Verdana, 42 pt, Cou
(Latin) Verdana, 42 pt, Gra:
(Latin) Verdana, Gauche :
(Latin) Verdana, Gras
(Latin) Verdana, Gras, Cent
(Latin) Verdana, Italique
10 pt, Après : 0 pt

Afficher l'aperçu
 Désactiver les styles liés

Options...   

Interaction Homme-Machine

GénieSystèmesInteractifsM1 CISI.pptx - PowerPoint

FICHIER ACCUEIL INSERTION CRÉATION ANIMATIONS DIAPORAMA RÉVISION AFFICHAGE

Philippe Palanque

Coller Nouvelle diapositive Diapositives Presse-papiers

Disposition Rétablir Section

Police Paragraphe Dessin

Remplissage Styles rapides Contour Effets Sélectionner Modification

1

Génie des Systèmes Interactifs

Philippe Palanque
palanque@irit.fr <http://www.irit.fr/~Philippe.Palanque>

Interactive Critical Systems
Université Paul Sabatier (Toulouse 3)

UNIVERSITÉ TOULOUSE III
PAUL SABATIER

IRIT
Institut de Recherche en Informatique de Toulouse

2

Plan du cours

- Introduction et définitions
- Processus de développement des SI
- Les techniques de spécification formelle pour les SI
- Les techniques d'ingénierie des SI
- Les méthodes de conception de SI
- Perspectives de recherche

3

Introduction à l'IHM

Gardez votre esprit critique !

LEUR CONTEXTE

Final application needs Human make the adaptation

BORIA Application area

Business CHAPTER

Complex products Application area

Enriched design Design and development

Design Design and development

Development process

4

Cliquez pour ajouter des commentaires

COMMENTAIRES

FRANÇAIS (FRANCE)

DIAPOSITIVE 1 DE 297

Interaction Homme-Machine

GénieSystèmesInteractifsM1 CISI.pptx - PowerPoint

FICHIER F ACCUEIL L INSERTION S CRÉATION E TRANSITIONS K ANIMATIONS A DIAPORAMA P RÉVISION R AFFICHAGE N

Nouvelle diapositive Rétablir Section

Coller Presse-papiers Diapositives

Police Paragraphe Dessin Modification

Rechercher Remplissage Styles rapides Remplacer Contour Effets Sélectionner

Bienvenue. Reprendre là où vous vous êtes arrêté : Diapositive 39 16/10/2014 - 13:03

Génie des Systèmes Interactifs
Philippe Palanque
philanque@irit.fr - <http://www.irit.fr/~Philippe.Palanque>
Interactive Critical Systems
Université Paul Sabatier (Toulouse 3)
UNIVERSITÉ TOULOUSE III PAUL SABATIER

Génie des Systèmes Interactifs

Philippe Palanque
philanque@irit.fr - <http://www.irit.fr/~Philippe.Palanque>

Interactive Critical Systems
Université Paul Sabatier (Toulouse 3)

UNIVERSITÉ TOULOUSE III PAUL SABATIER Université de Toulouse

IRIT Institut de Recherche en Informatique de Toulouse

1 Plan du cours

- Introduction et définitions
- Processus de développement des SI
- Les techniques de spécification formelle pour les SI
- Les techniques d'ingénierie des SI
- Les méthodes de conception de SI
- Perspectives de recherche

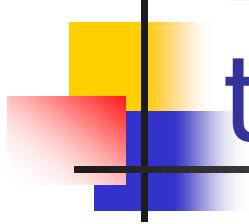
2 Introduction à l'IHM

Gardez votre esprit critique !

3 Cliquez pour ajouter des commentaires

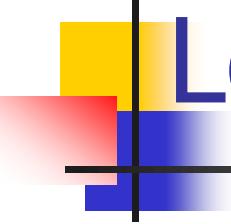
4

DIapositive 1 DE 297 FRANÇAIS (FRANCE) COMMENTAIRES COMMENTAIRES



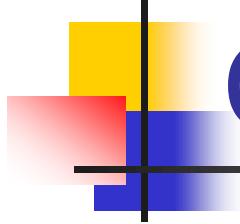
Ingénierie des modèles de tâche

- Notation modifiable et concise
- Approche systématique capable d'indiquer comment réutiliser les informations des modèles de tâche
- Outils automatiques pour utiliser efficacement ces informations

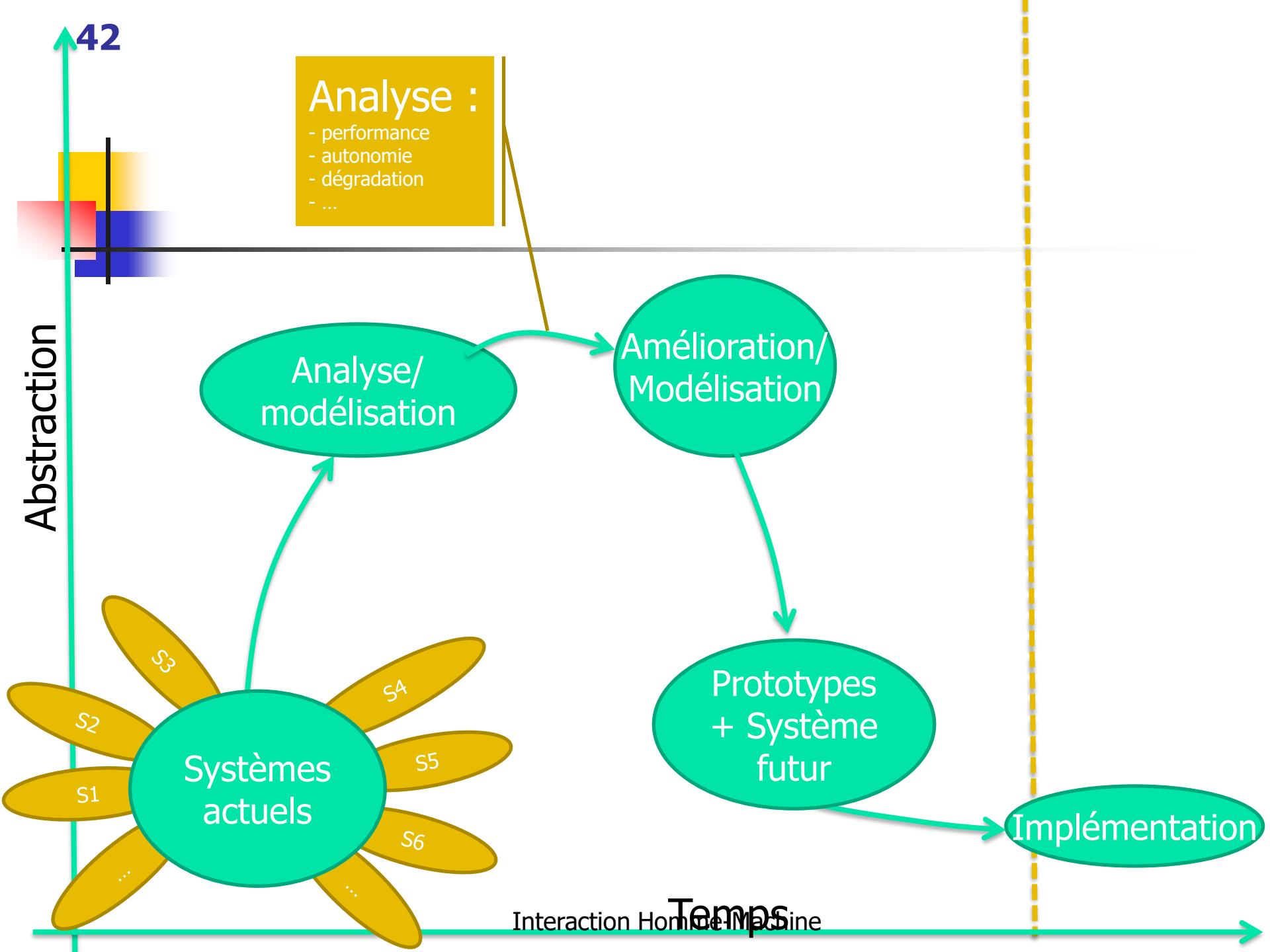


Les différents modèles de tâche

- Système existant
 - Décrire les tâches des utilisateurs sur les systèmes actuellement à leur disposition
- Système futur
 - Envisionner les tâches de l'utilisateur sur un système futur (en cours de conception)
- Activité de l'utilisateur
 - Recenser les activités menées par les utilisateurs sur les systèmes actuels ou futur (un processus d'abstraction et de modélisation est nécessaire pour représenter les informations recueillies)

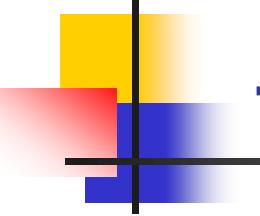


Courbe du soleil (Merise)



Modèles de tâches versus Scénarios

- Les scénarios contiennent des descriptions précises d'utilisation dans un contexte précis et en fonction d'un système donné
- Les modèles de tâche décrivent les activités possibles théoriques et leurs relations
- Les scénarios ont une couverture moindre et sont plus précis
- Les scénarios peuvent aider à la modélisation des tâches
- Les modèles de tâches peuvent aider à l'identification de scénarios intéressants



Utilisation des modèles de tâche

- Améliorer la compréhension de l'application (et en particulier de son utilisation)
- Enregistrer les résultats de discussions multidisciplinaires
- Aider à la conception
- Aider à l'évaluation de l'utilisabilité
- Aider à l'évaluation de l'efficacité
- Aider l'utilisateur durant son travail (aide contextuelle)
- Documentation (contenu + structure)

Représentation des modèles de tâche

- Hierarchical task analysis (HTA)
- GOMS (la famille)
- UAN (User Action Notation)
- MAD (INRIA)
- Différentes syntaxe (textuel vs graphique)
- Différents niveau de formalité
- Différents opérateurs pour la décomposition des tâches

GOMS : un exemple

GOAL: EDIT-MANUSCRIPT

 GOAL: EDIT-UNIT-Task repeat until no more unit tasks

 GOAL: ACQUIRE-UNIT-TASK

 GET-NEXT-PAGE if at end of manuscript

 GET-NEXT-TASK

 GOAL: EXECUTE-UNIT-TASK

 GOAL: LOCATE-LINE

 [select: USE-QS-METHOD

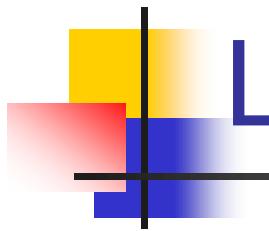
 USE-LF-METHOD]

 GOAL: MODIFY-TEXT

 [select: USE-S-METHOD

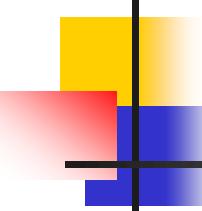
 USE-M-METHOD]

 VERIFY-EDIT



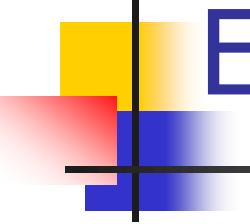
Limitations de GOMS

- Ne prends pas en compte les erreurs utilisateur
- Ne prends pas en compte les interruptions de tâches
- Ne considère que les tâches séquentielles



UAN - User Action Notation

- 2 représentations complémentaires
 - Une hiérarchie de tâches utilisant les opérateurs LOTOS (proche de CTT)
 - Les actions de l'utilisateur et le feedback des utilisateurs sont spécifiés sous forme tabulaire
- Notation textuelle
- Introduit dans un livre en 1992 (Hix & Hartson grand succès *Developing user interfaces Ensuring, Usability Through Product & Process*)

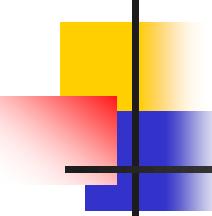


Exemple de description UAN

Task: Obtain cash
 $((Insert_card$
 $Enter_code)$
 $\& (Request_cash$
 $Select_amount))$
 $(Withdraw_card$
 $\& Withdraw_cash))$
 $| (Request_cash$
 $Insert_card$
 $Enter_code$
 $Select_amount$
 $(Withdraw_card$
 $\& Withdraw_cash))$

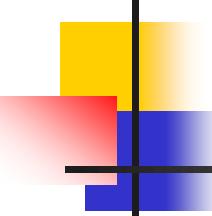
Task: SelApplication

User Action	Interface Feedback	Interface State
$\sim[x,y \text{ in AppICON}]$ $\vee \wedge(t < t_{doubleClick}) \vee \wedge$	w'!: w'-! UnMap(PrevAppliMenu) Map(AppMenu) UnMap(AppICON)	CurAppli=App CurMenu=AppMenu



HAMSTERS

- Met l'accent sur les activités de l'utilisateur
- Structure hiérarchique
- Syntaxe Graphique
- Ensemble de plusieurs opérateur temporels (qualitatifs uniquement)
- Port d'Entrée et Sortie pour les tâches permettant la transmission d'objets
- Attributs pour les tâches
- Opérateurs sont des entités distinctes de tâches, évite les ambiguïtés de description



Opérateurs temporels

Enabling/Activation $T_1 \gg T_2$

Disabling/Désactivation $T_1 [> T_2$

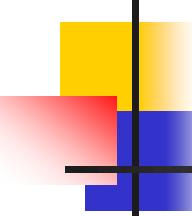
Interruption/Interruption $T_1 |> T_2$

Choice/Choix $T_1 [] T_2$

Concurrency/Concurrence $T_1 ||| T_2$

Order independent/

Ordre indifférent $T_1 |=| T_2$



Opérateurs temporels

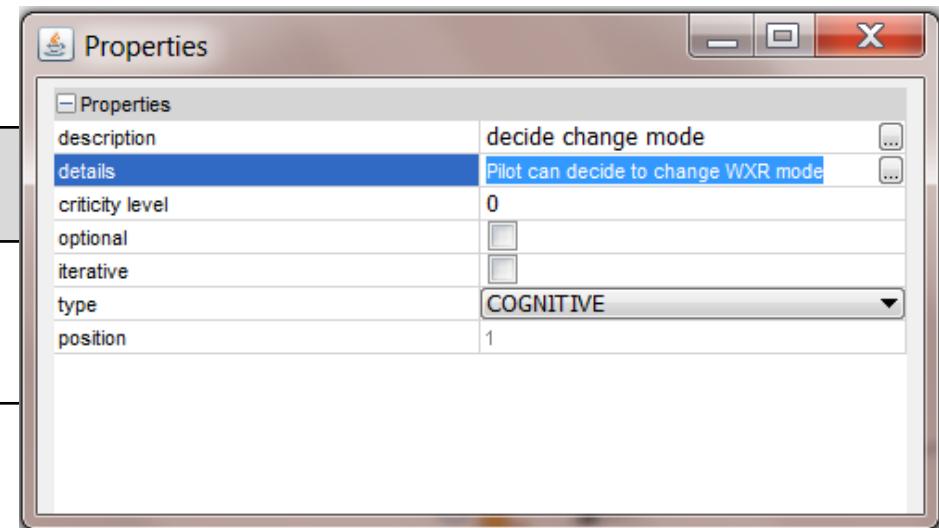
- $T_1 \parallel \parallel T_2$: les actions T_1 et T_2 peuvent être effectuées en même temps
- $T_1 | = | T_2$: les actions de T_1 et T_2 peuvent être effectuées dans n'importe quel ordre
- $T_1 >> T_2$: quand T_1 est terminée T_2 devient active
- $T_1 [> T_2$: quand une action de T_2 se produit, la tâche T_1 est désactivée
- $T_1 | > T_2$: quand une action de T_2 se produit, la tâche T_1 est interrompue, puis reprise lorsque T_2 est terminée

Caractéristique des tâches

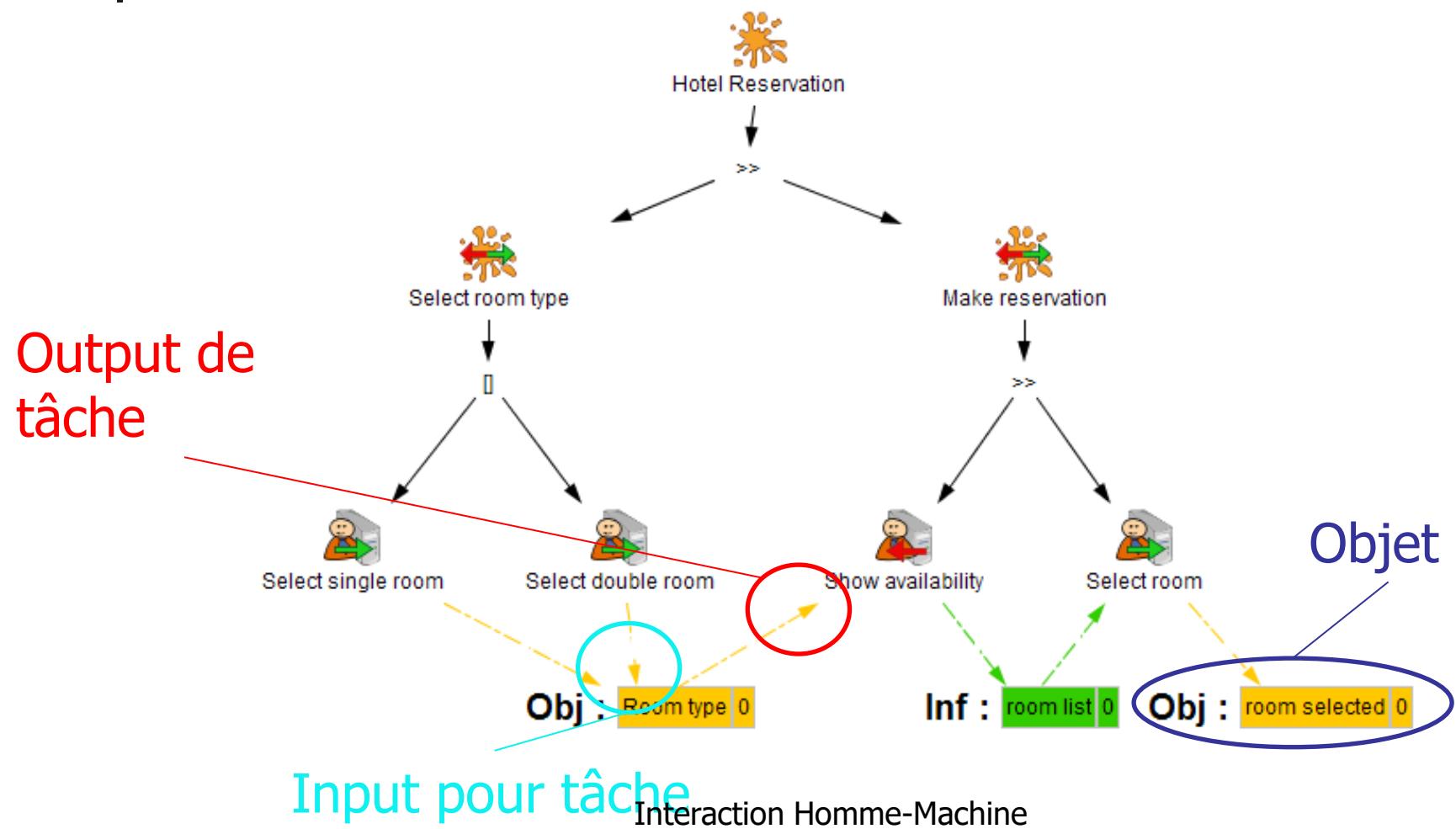
-  est facultative pour atteindre le but (la tâche de niveau supérieur)
-  peut être répétée autant de fois que possible pour atteindre le but
-  facultative et pouvant être répétée

Types de tâches

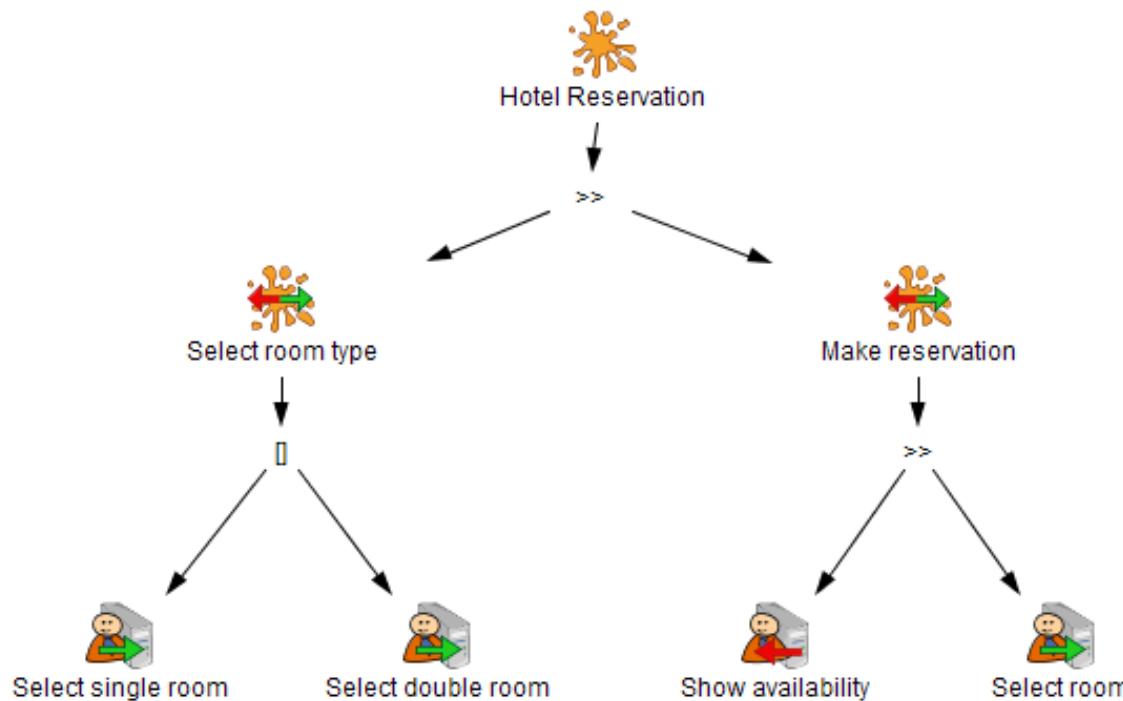
Task type	Icons in the task model				
Abstract Task	 Abstract task				
System Task	 System task				
User Task	    User task Cognitive task Perceptive task Motor task				
Interactive Task	    Interactive task Input task Output task InputOutput task				



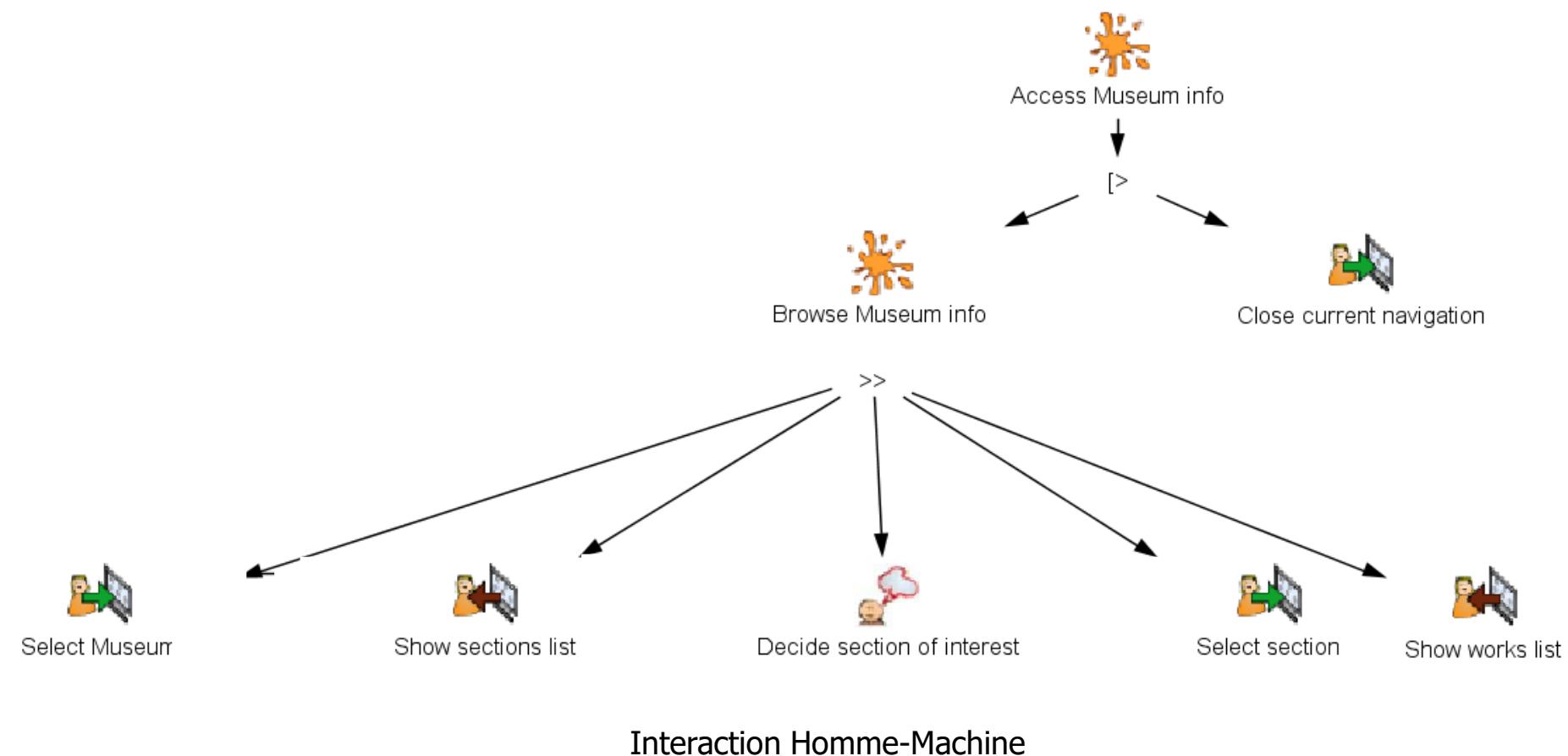
Objets



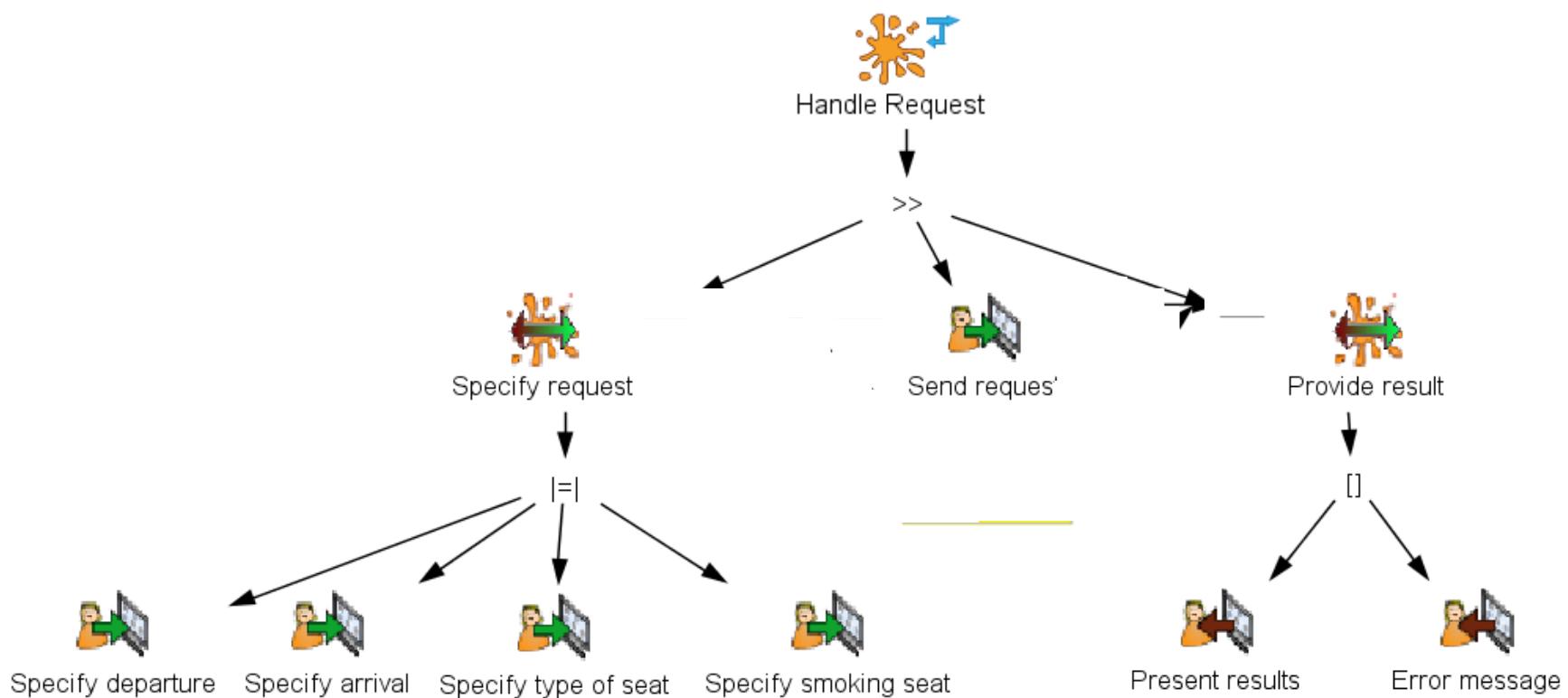
Hiérarchie et relations temporelles



Relations tâches / sous-tâches



Tâches facultatives / choix



Structuring Task Models



Watch Telemetry

Structuring mechanisms

Related icon in HAMSTERS

Sub-routine with no input value and no output value



Reset Flight SW

Sub-routine with at least one input value and no output value



Record failure

Sub-routine with no input value and at least one output value



Find root cause of the switch

Sub routine with at least one input value and at least one output value



Operate communication frequency

Interaction Homme-Machine



Current Situation on Task Modelling

- Foundations identified many years ago
 - Annett & Duncan HTA in 1967
 - Payne & Green TAG 1986
 - Johnson et al. TKS 1988
 - Van der Veer et al. GTA 1996
- Include
 - Behavioural and temporal aspects (procedural)
 - Knowledge
 - Cooperation/collaboration
- Refinement and deeper understanding over the years
 - MAD, CTT, UAN, ... , KMADE, DSA/SAMANTA
- Need for long term detailed smaller refinements

Current Situation on Task Modelling

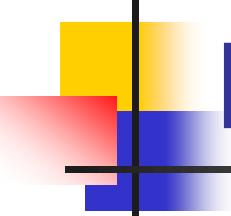
- Task modelling notations focus on procedural knowledge
- Current task modelling approaches overlook declarative knowledge
 - Mainly represented as comments
 - Appear sometimes in pre-conditions (but are not addressed during simulation)
 - Represented in an unstructured way
- Declarative knowledge (objects and information) is part of the real world and of users' activity and must be addressed as carefully as procedural knowledge

Current Situation

	Requirements /notations [reference]	HTA [25]	GOMS [8]	MAD [34]	TKS [17]	TAKD [13]	GTA [44]	UAN [16]	Diane + [41]	VTML [6]	CTT [29]	CWA [45]	AMBOS S [14]	DSA [46]	HAMSTERS
Declarative Knowledge (DK)	REQ_1. Concepts used to represent DK	x	x	x	✓	x	x	x	x	x	x	✓	x	✓	x
	REQ_2. Refinement of concepts into objects and into information	x	x	Partly with objects (explicit)	x	Partly with objects	x	x	x	x	Partly with objects	x	x	Partly with objects	Partly with objects
	REQ_3. Concepts connections should be made explicit	x	x	x	✓	x	x	x	x	x	x	✓	x	✓	x
	REQ_4. Semantic networks to represent the concepts and their relationships	x	x	x	x	x	x	x	x	x	x	x	x	✓	x
	REQ_5. Refinement of DK into strategic knowledge and into situational knowledge	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Procedural Knowledge (PK)	REQ_6 and REQ_7. Routines, subroutines and their interrelationships to represent PK.	x	x	x	x	x	x	x	x	x	x	x	x	x	✓
	REQ_8. Hierarchical structures to represent routines , sub-routines and their interrelationships	✓	✓	✓	x	x	✓	✓	✓	✓	x	✓	✓	✓	✓
	REQ_9. Refinement of PK into strategic knowledge and into situational knowledge	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Integration	REQ_10. All the information above should be integrated in a single model	x	Through pre-condition	Through pre-condition	Through pre-condition	Through pre-condition	Through pre-condition	x	Through pre-condition	Through pre-condition	Through pre-condition	Through constraint	Through pre-condition	Through pre-condition	Through pre-condition
Tool Support		Task-Architect	CAT-HCI and others x	K-MADe	ADEPT	LUTAKD	GTA and EUTERPE	Quantum	TAMOT and Isolde	SpectRM tool Suite	CTTe		AMBOSS	SAMANTA	HAMSTERS

Current Situation

	Requirements /notations [reference]	HTA [25]	GOMS [8]	MAD [34]	TKS [17]	TAKD [13]	GTA [44]	UAN [16]	Diane + [41]	VTML [6]	CTT [29]	CWA [45]	AMBOS S [14]	DSA [46]	HAMSTERS
Declarative Knowledge (DK)	REQ_1. Concepts used to represent DK	x	x	x	✓	x	x	x	x	x	x	✓	x	✓	x
	REQ_2. Refinement of concepts into objects and into information	x	x	Partly with objects (explicit)	x	Partly with objects	x	x	x	x	x	Partly with objects	x	x	Partly with objects
	REQ_3. Concepts connections should be made explicit	x	x	x	✓	x	x	x	x	x	x	✓	x	✓	x
	REQ_4. Semantic networks to represent the concepts and their relationships	x	x	x	x	x	x	x	x	x	x	x	x	✓	x
	REQ_5. Refinement of DK into strategic knowledge and into situational knowledge	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Procedural Knowledge (PK)	REQ_6 and REQ_7. Routines, subroutines and their interrelationships to represent PK.	x	x	x	x	x	x	x	x	x	x	x	x	x	✓
	REQ_8. Hierarchical structures to represent routines , sub-routines and their interrelationships	✓	✓	✓	x	x	✓	✓	✓	✓	✓	x	✓	✓	✓
	REQ_9. Refinement of PK into strategic knowledge and into situational knowledge	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Integration	REQ_10. All the information above should be integrated in a single model	x	Through pre-condition	Through pre-condition	Through pre-condition	Through pre-condition	Through pre-condition	x	Through pre-condition	Through pre-condition	Through pre-condition	Through constraint	Through pre-condition	Through pre-condition	Through pre-condition
Tool Support		Task-Architect	CAT-HCI and others x	K-MADE	ADEPT	LUTAKD	GTA and EUTERPE	Quantum	TAMOT and Isolde	SpectRM tool Suite	CTTe		AMBOSS	SAMANTA	HAMSTERS



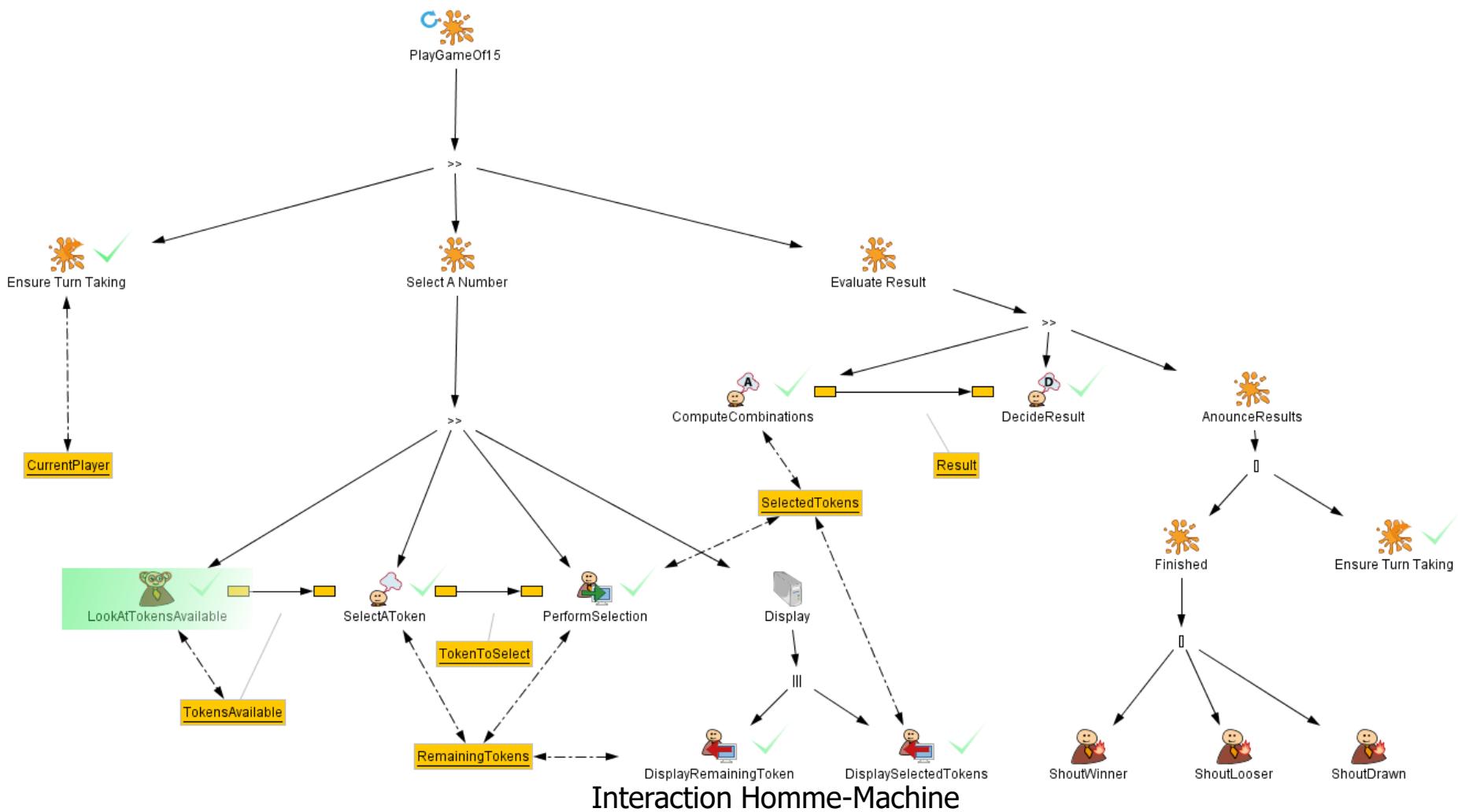
Etude de Cas: Le jeu de 15

- Le jeu se joue à 2
- Vous disposez des jetons 1, 2, 3, 4, 5, 6, 7, 8, 9
- On joue chacun son tour, chaque joueur prend un seul jeton à la fois, un jeton ne peut être pris qu'une fois
- Le premier qui possède 3 jetons (parmi ceux qu'il possède i.e. 3, 4 ou 5) dont la somme est exactement 15 a gagné

Devenez des experts

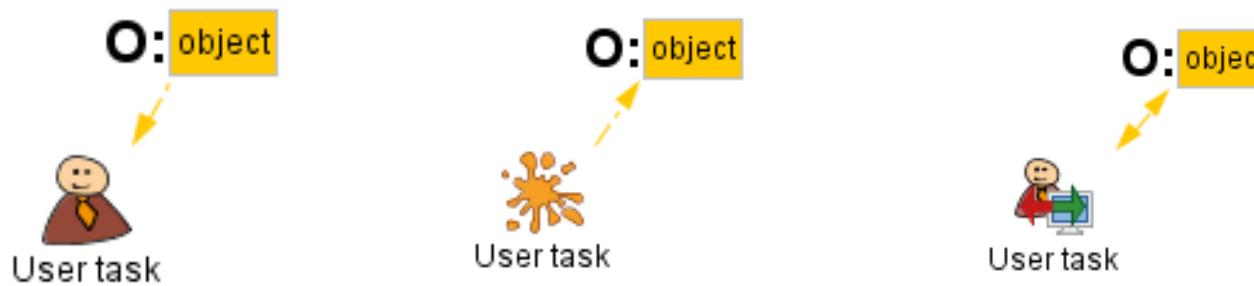
Jouez à ce jeu sans aucun outil (papier, crayon, ...)

First draft of tasks Modelling using HAMSTERS

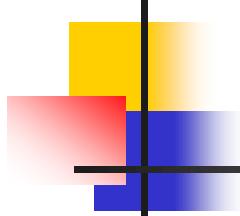


Concepts: Objects, Information & knowledge (1)

- Object (similar to OOP)
 - Structured entity required for (or produced by) the performance of a task



- Information
 - Data required for the performance of a task
 - Data produced by the performance of a task



Concepts: Objects, Information & knowledge (2)

- Knowledge required to perform a task

- Refinement in 2 sub-types
 - Strategic
 - Situational

DK: Declarative knowledge

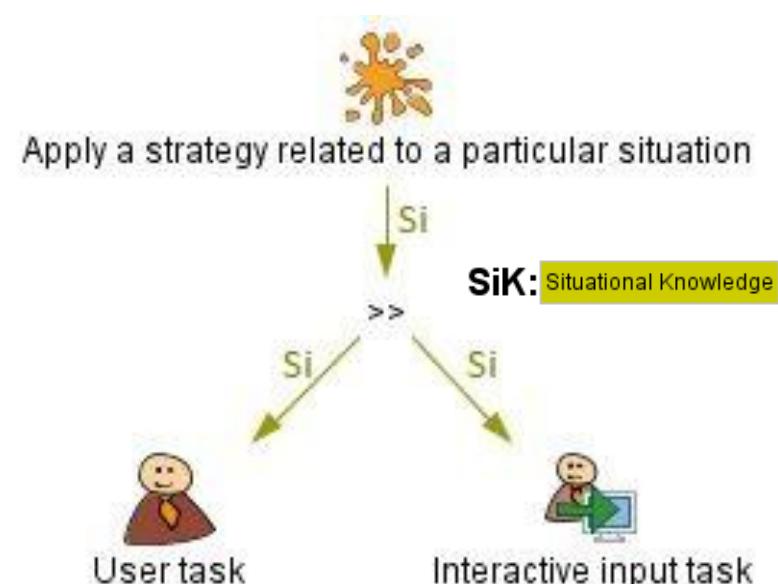
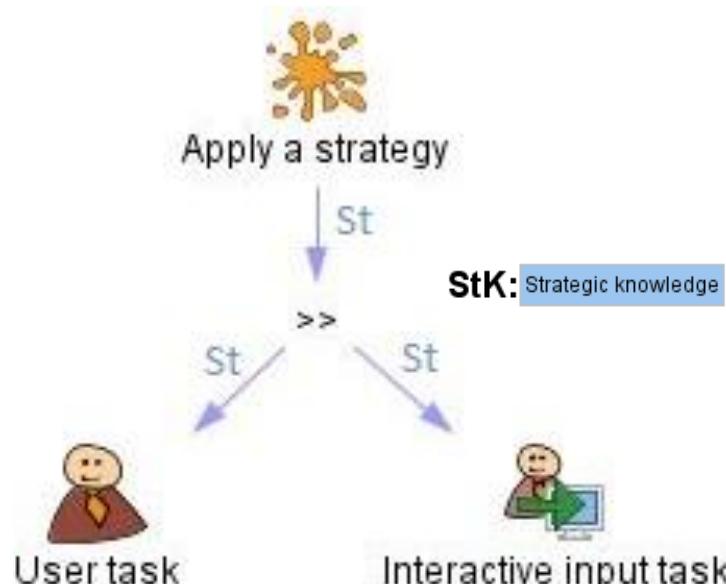
StK: Strategic knowledge

SiK: Situational Knowledge

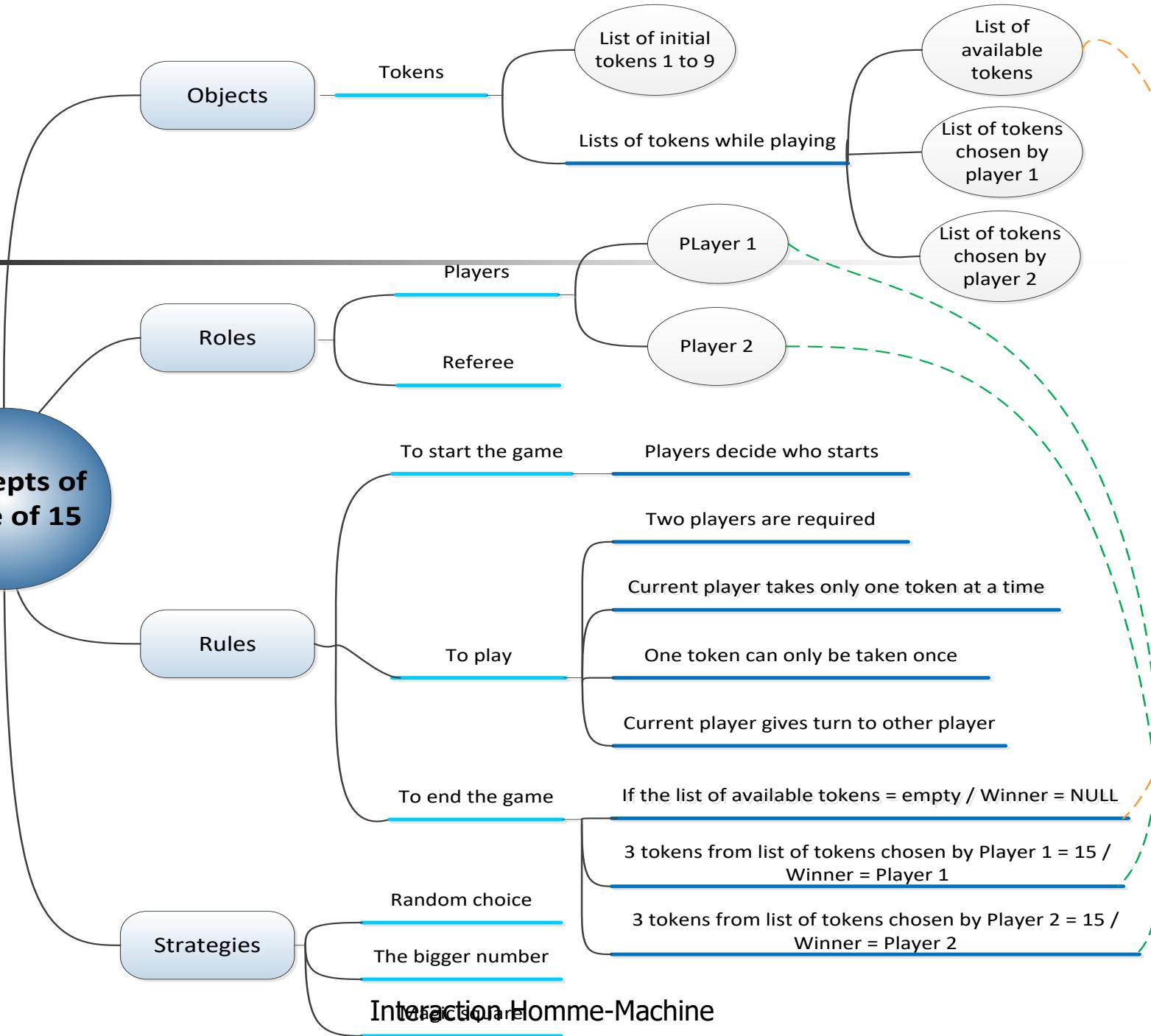
Concepts: Objects, Information & knowledge (3)

Procedural knowledge

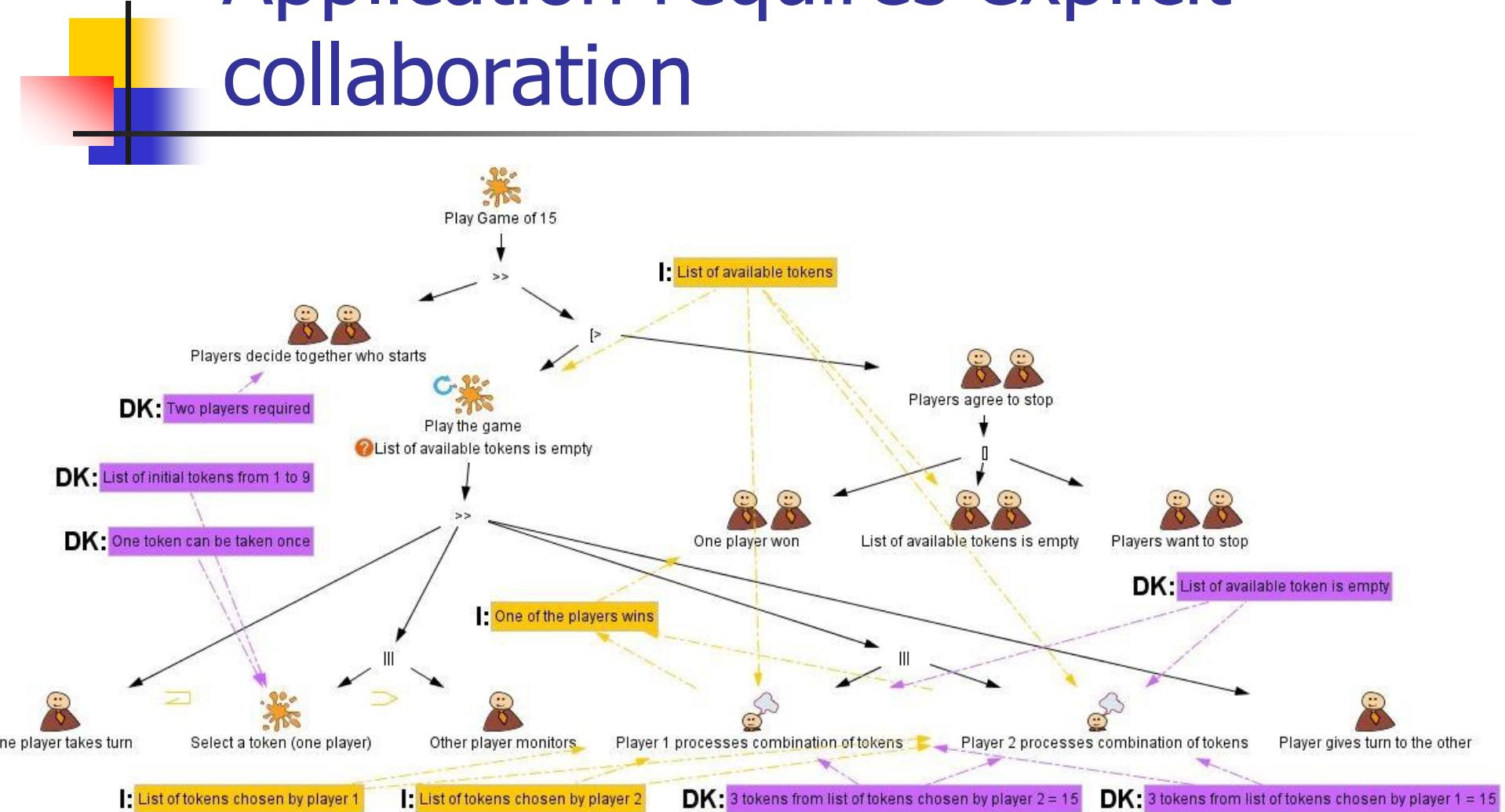
- Strategic (plan)
- Situational (CBR)



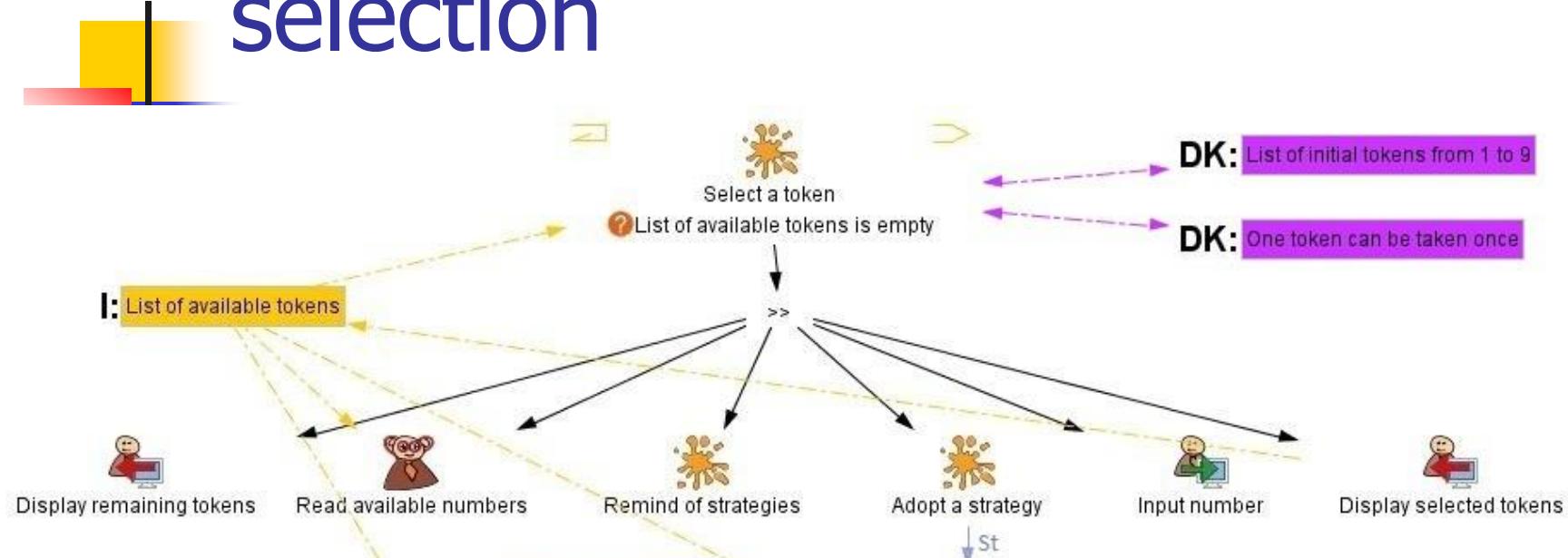
Concepts of game of 15



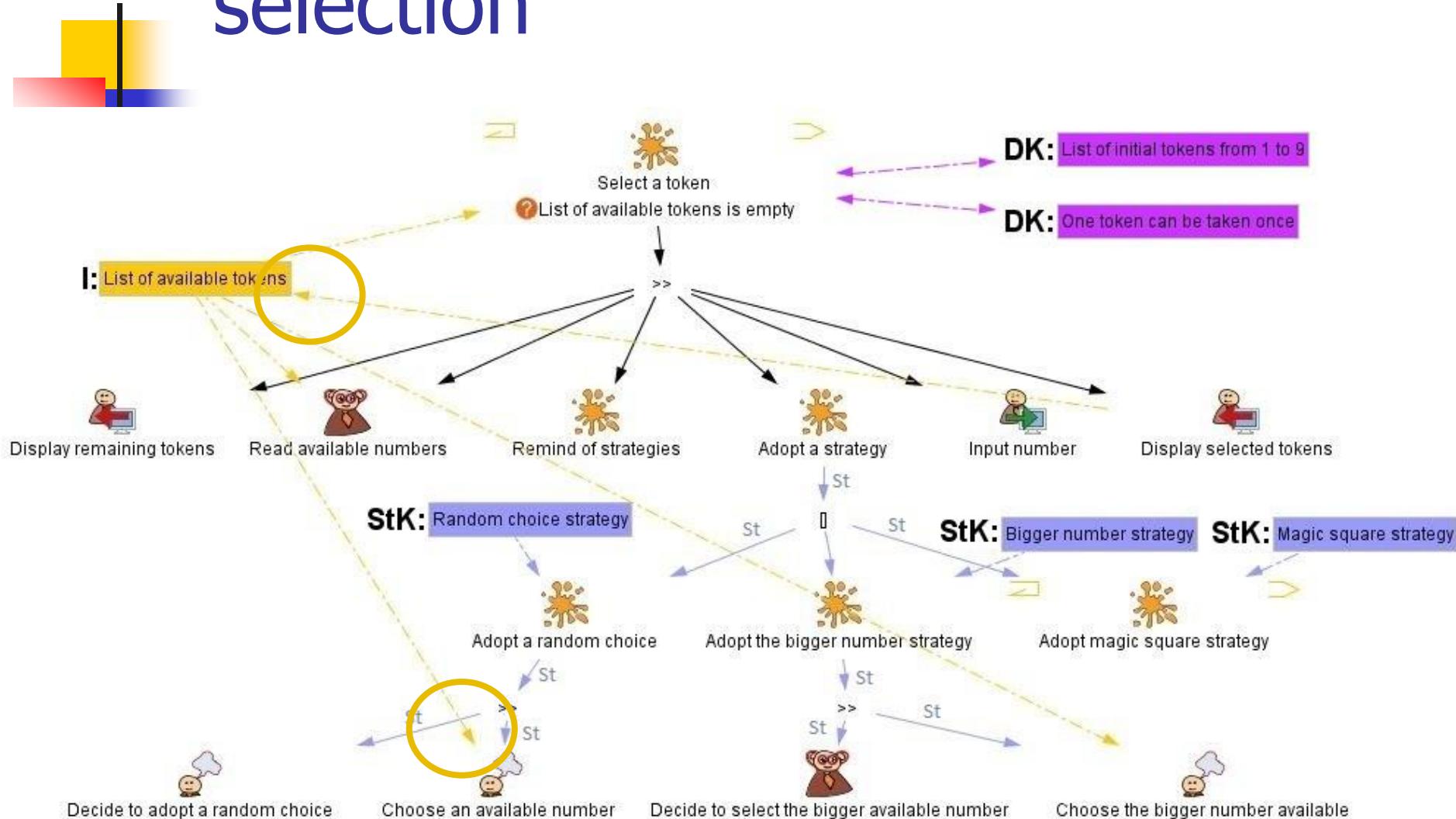
Application requires explicit collaboration

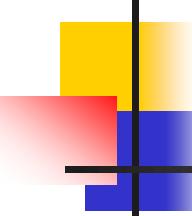


Refinement of the token selection



Refinement of the token selection





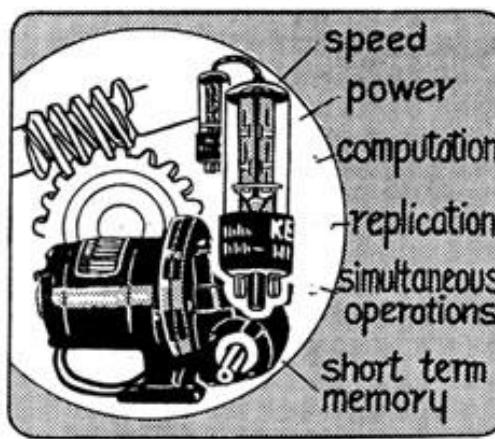
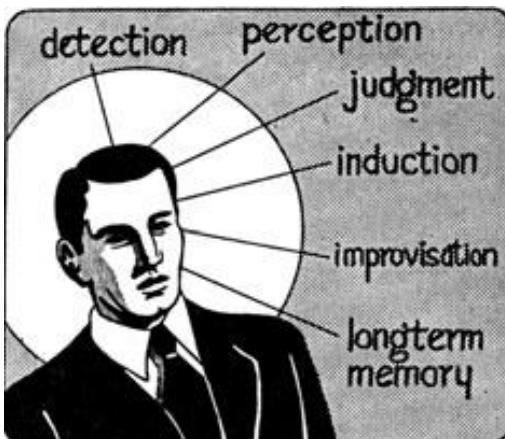
Objectives

- Model-based assessment of tasks complexity (the more information the better)
 - Quantity of concepts, procedural aspects & temporal constraints
 - Performance evaluation (additional information in the tasks)
 - Identification for improvements
 - Identification of training needs
- Use for interactive system design
 - The virtuous task-artefact life cycle (;-) Caroll & Rosson 91)
 - The impact of functions allocation (system-users)
- Use for identifying & managing conflicts between properties
 - Usability, reliability, dependability, safety, ...

Fitts List (p7-8)

Fitts PM (ed) (1951) Human engineering for an effective air navigation and traffic control system. National Research Council, Washington, DC

MABA – MABA
Men Are Better At – Machines Are Better At



Speed	●	●
Memory	●	●
Sensing	●	●
Perceiving	●	●
Reasoning	●	●
Consistency	●	●
Computation	●	●
Power Output	●	●
Information Capacity	●	●

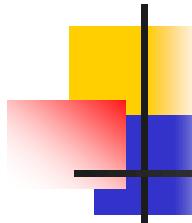
Analyzing Gains and Losses

- Useful for large case studies
- More complex tasks migrations based on information
- Assessment of tasks complexity using scenarios

Sub-goals vs. Task number for each type	Perceptive	Analysis	Decision	Action	System	Interactive output	Interactive input
Non automated Search and Selection (Fig. 6)	i+1	i+1	1	1	1	1	1
Automated Search (Fig. 7)	2	2	1	1	4	1	1
Automated Search and Selection (Fig.8)	1	2	0	0	3	1	0

Objectives

- Model-based assessment of tasks complexity (the more information the better)
 - Quantity of concepts, procedural aspects & temporal constraints
 - Performance evaluation (additional information in the tasks)
 - Identification for improvements
 - Identification of training needs
- Use for interactive system design
 - The **virtuous** task-artefact life cycle (;-) Carroll & Rosson 91)
 - The impact of functions allocation (system-users)
- Use for identifying conflicts between properties
 - Usability, reliability, dependability, safety, ...



Certification Specifications for Large Aeroplanes

CS-25

CS-25 BOOK 1

SUBPART F – EQUIPMENT

GENERAL

CS 25.1301 Function and installation
(See AMC 25.1301)

Each item of installed equipment must –

- (a) Be of a kind and design appropriate to its intended function;
 - (b) Be labelled as to its identification, function, or operating limitations, or any applicable combination of these factors. (See AMC 25.1301(b).)
 - (c) Be installed according to limitations specified for that equipment.
- [Amdt. No.:25/2]

CS 25.1302 Installed systems and equipment for use by the flight crew
(See AMC 25.1302)

This paragraph applies to installed equipment intended for flight-crew members' use in the operation of the aeroplane from their normally seated positions on the flight deck. This installed equipment must be shown, individually and in combination with other such equipment, to be designed so that qualified flight-crew members trained in its use can safely perform their tasks associated with its intended function by meeting the following requirements:

- (a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.
- (b) Flight deck controls and information intended for flight crew use must:
 - (1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.
 - (2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and
 - (3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.
 - (c) Operationally-relevant behaviour of the installed equipment must be:
 - (1) Predictable and unambiguous, and

(2) Designed to enable the flight crew to intervene in a manner appropriate to the task.

(d) To the extent practicable, installed equipment must enable the flight crew to manage errors resulting from the kinds of flight crew interactions with the equipment that can be reasonably expected in service, assuming the flight crew is acting in good faith. This sub-paragraph (d) does not apply to skill-related errors associated with manual control of the aeroplane.

[Amdt. No.:25/3]

CS 25.1303 Flight and navigation instruments

(a) The following flight and navigation instruments must be installed so that the instrument is visible from each pilot station:

- (1) A free-air temperature indicator or an air-temperature indicator which provides indications that are convertible to free-air temperature.
- (2) A clock displaying hours, minutes, and seconds with a sweep-second pointer or digital presentation.
- (3) A direction indicator (non-stabilised magnetic compass).
- (b) The following flight and navigation instruments must be installed at each pilot station:
 - (1) An airspeed indicator. If airspeed limitations vary with altitude, the indicator must have a maximum allowable airspeed indicator showing the variation of V_{MO} with altitude.
 - (2) An altimeter (sensitive).
 - (3) A rate-of-climb indicator (vertical speed).

(4) A gyroscopic rate of turn indicator combined with an integral slip-skid indicator (turn-and-bank indicator) except that only a slip-skid indicator is required on aeroplanes with a third attitude instrument system usable through flight attitudes of 360° of pitch and roll, which is powered from a source independent of the electrical generating system and continues reliable operation for a minimum of 30 minutes after total failure of the electrical generating system, and is installed in accordance with CS 25.1321 (a).

GENERAL

CS 25.1301 Function and installation
(See AMC 25.1301)

Each item of installed equipment must –

- (a) Be of a kind and design appropriate to its intended function;
 - (b) Be labelled as to its identification, function, or operating limitations, or any applicable combination of these factors. (See AMC 25.1301(b).)
 - (c) Be installed according to limitations specified for that equipment.
- [Amendt. No.:25/2]

CS 25.1302 Installed systems and equipment for use by the flight crew
(See AMC 25.1302)

This paragraph applies to installed equipment intended for flight-crew members' use in the operation of the aeroplane from their normally seated positions on the flight deck. This installed equipment must be shown, individually and in combination with other such equipment, to be designed so that qualified flight-crew members trained in its use can safely perform their tasks associated with its intended function by meeting the following requirements:

- (a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

(b) Flight deck controls and information intended for flight crew use must:

- (1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.

- (2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and

- (3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.

(c) Operationally-relevant behaviour of the installed equipment must be:

- (1) Predictable and unambiguous, and

errors resulting from interactions with the reasonably expected in service crew is acting in good faith does not apply to skill-related manual control of the aeroplane

[Amendt. No.:25/3]

CS 25.1303 Flight instr

(a) The following instruments must be installed and visible from each pilot's seat:

- (1) A free-air temperature indicator showing indications that are accurate to within 1°C of air temperature.

- (2) A clock and seconds with a digital presentation.

- (3) A direction magnetic compass.

(b) The following instruments must be installed:

- (1) An airspeed indicator. Airspeed limitations vary with altitude and have a maximum altitude above which the variation cannot be guaranteed.

- (2) An altimeter.

- (3) A rate-of-climb indicator showing speed.

(4) A gyroscopic gyroscope combined with an integrated turn-and-bank indicator. A skid indicator is required. A third attitude instrument is required for flight attitudes of 360° of roll powered from a source electrical generating system reliable operation for a minimum of 10 minutes after total failure of the system, and is installed in accordance with CS 25.1321 (a).

(a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

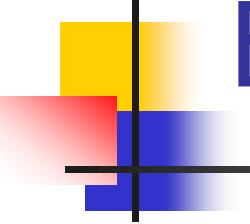
(b) Flight deck controls and information intended for flight crew use must:

(1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.

(2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and

(3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.

(d) To the extent practicable, installed equipment must enable the flight crew to manage errors resulting from the kinds of flight crew interactions with the equipment that can be reasonably expected in service, assuming the flight crew is acting in good faith. This sub-paragraph (d) does not apply to skill-related errors associated with manual control of the aeroplane.

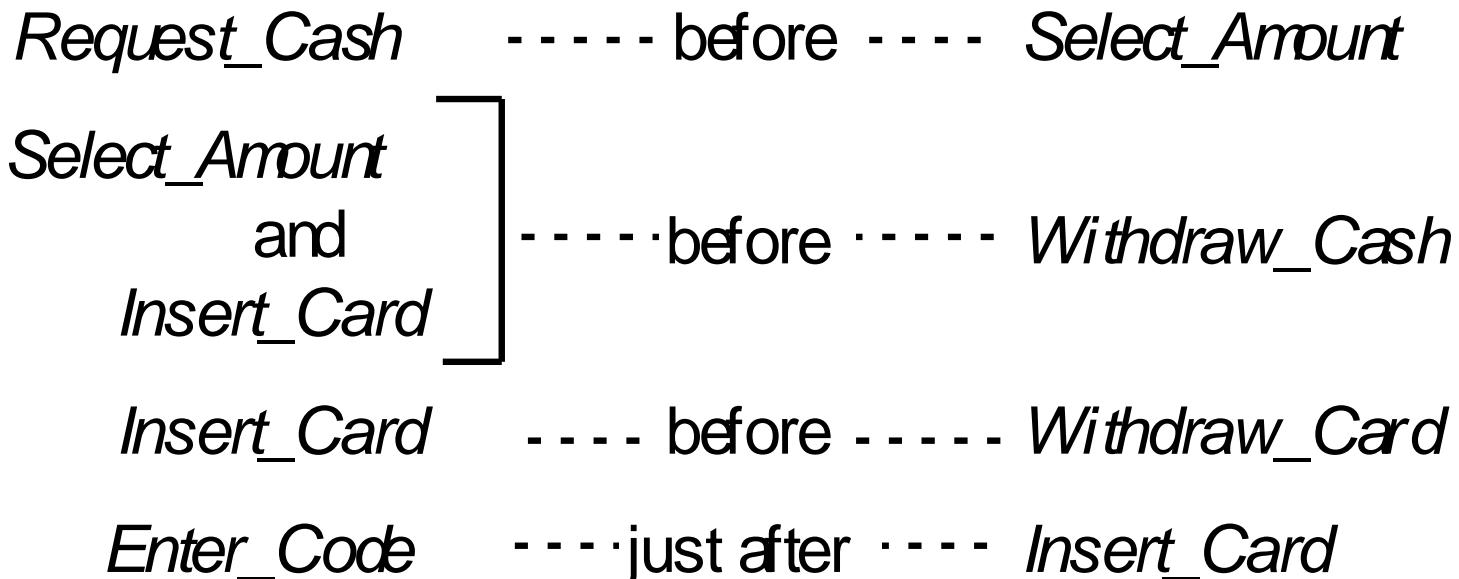


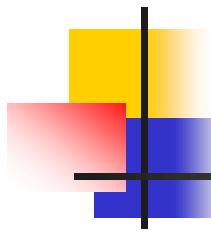
Exercice: GBLS

- Modéliser les tâches d'un utilisateur d'un distributeur de billets automatique

Exercice: GBLS contraintes

- *Insérer_Carte, Entrer_Code, Retirer_Carte*
- *Demander_Cash, Select_Montant, Retirer_Cash*

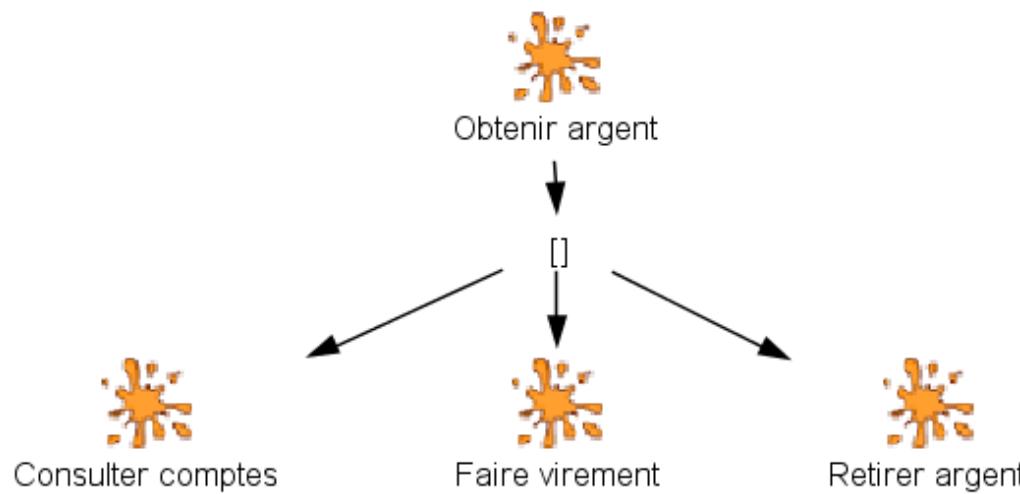




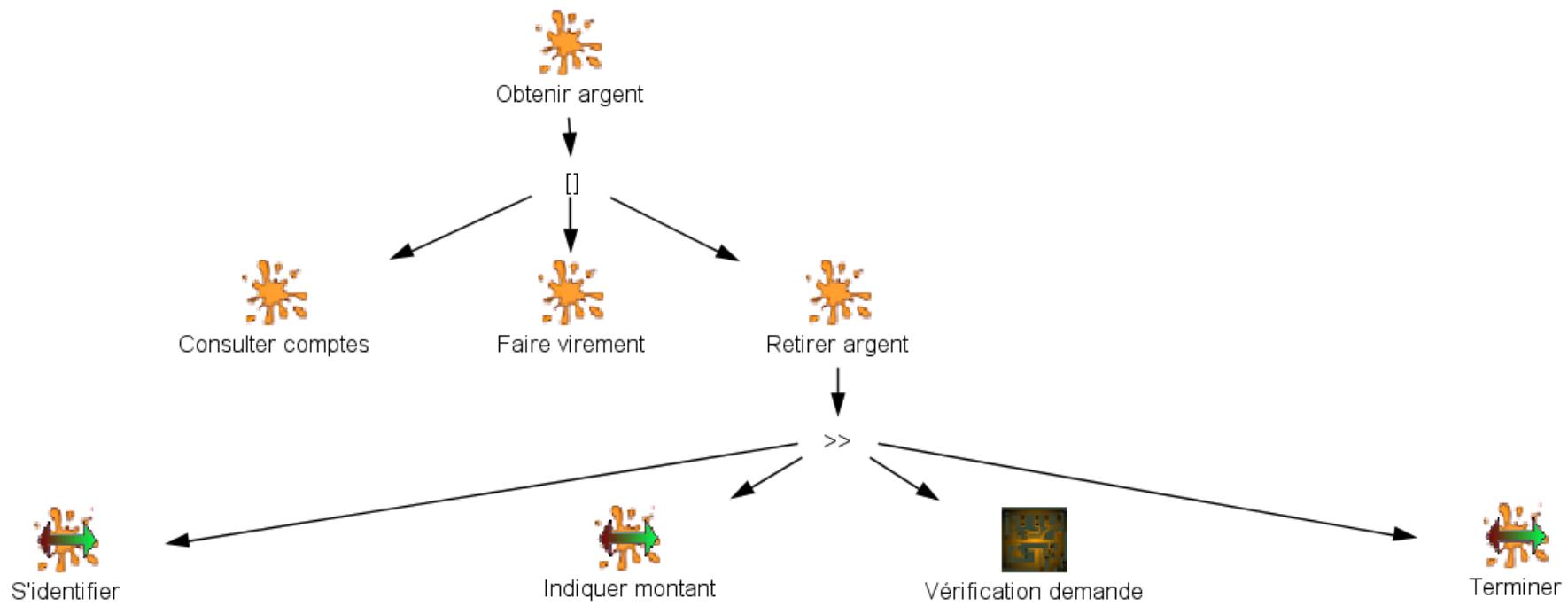
GBLS: tâches

- But Obtenir de l'argent
- Pré-requis:
 - Avoir une carte
 - Connaître le montant que l'on veut retirer
 - Connaître le code secret
- Résultats
 - Prendre l'argent
 - Prendre la carte

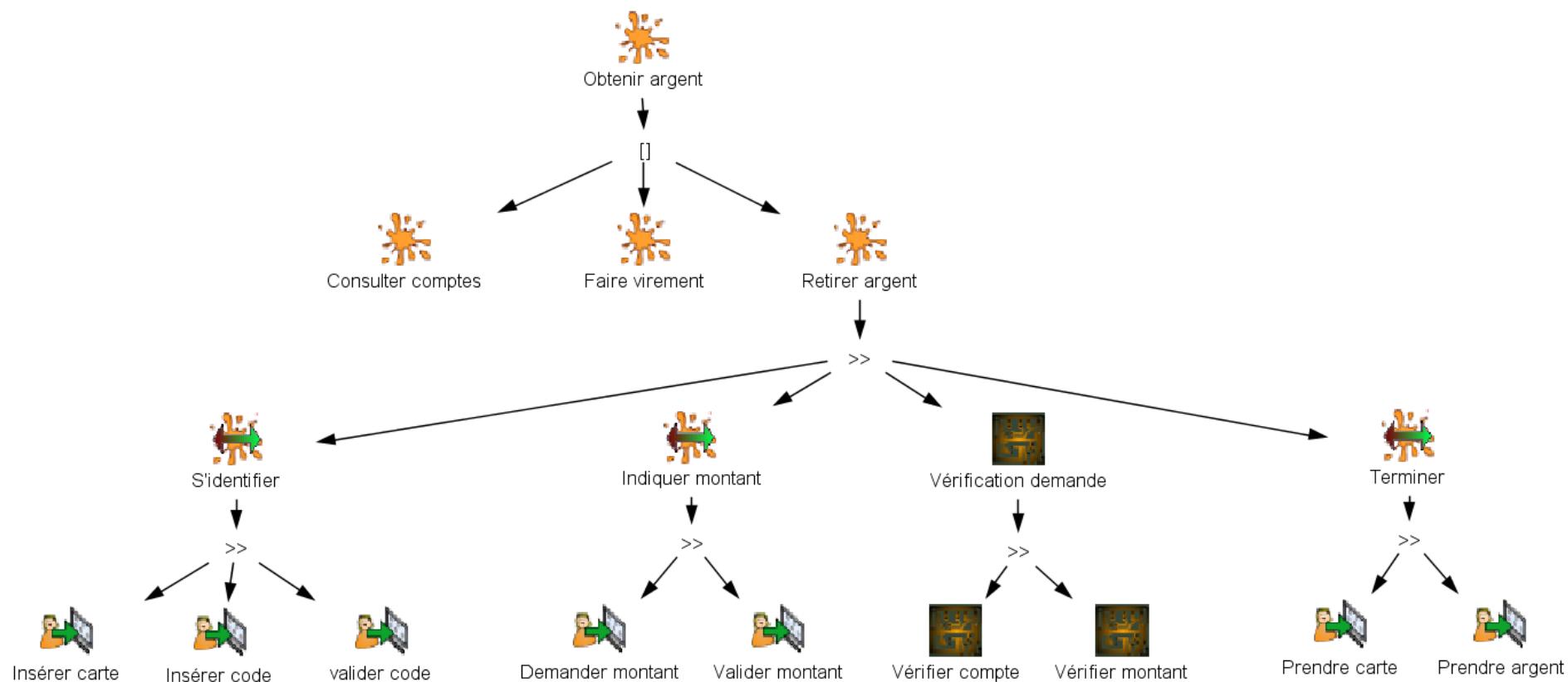
GBLS: Modèle de tâches



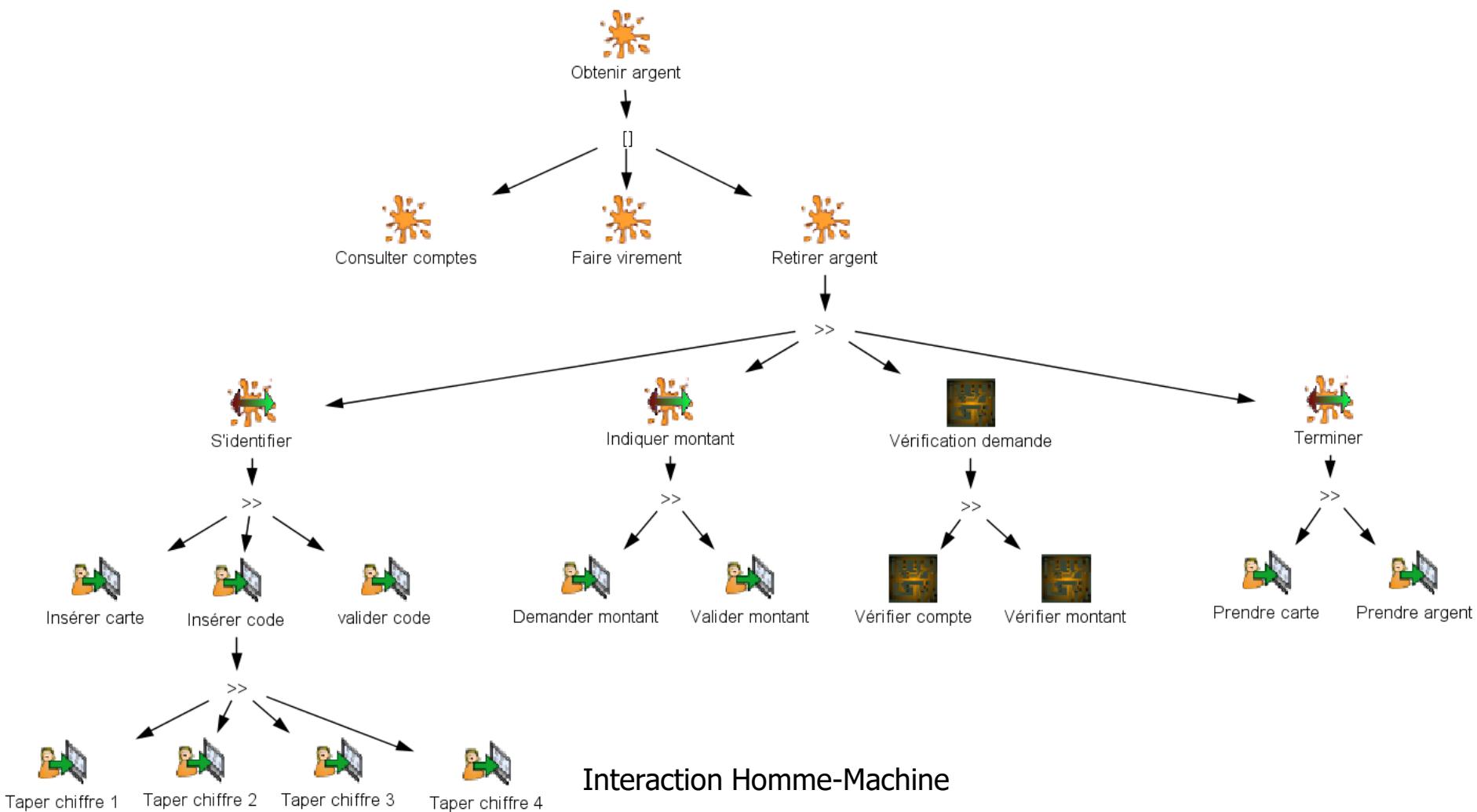
GBLS: Modèle de tâches



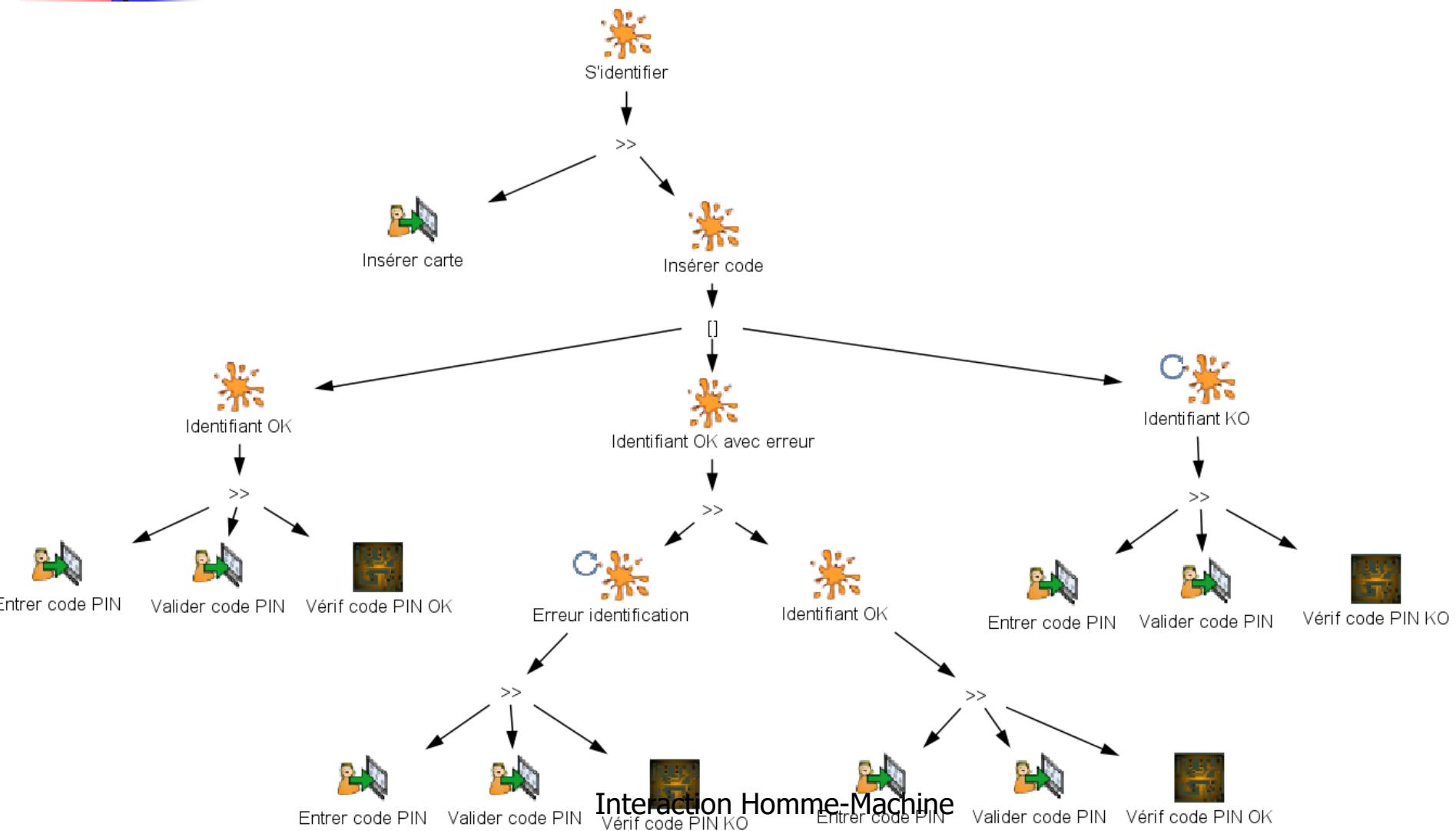
GBLS: Modèle de tâches

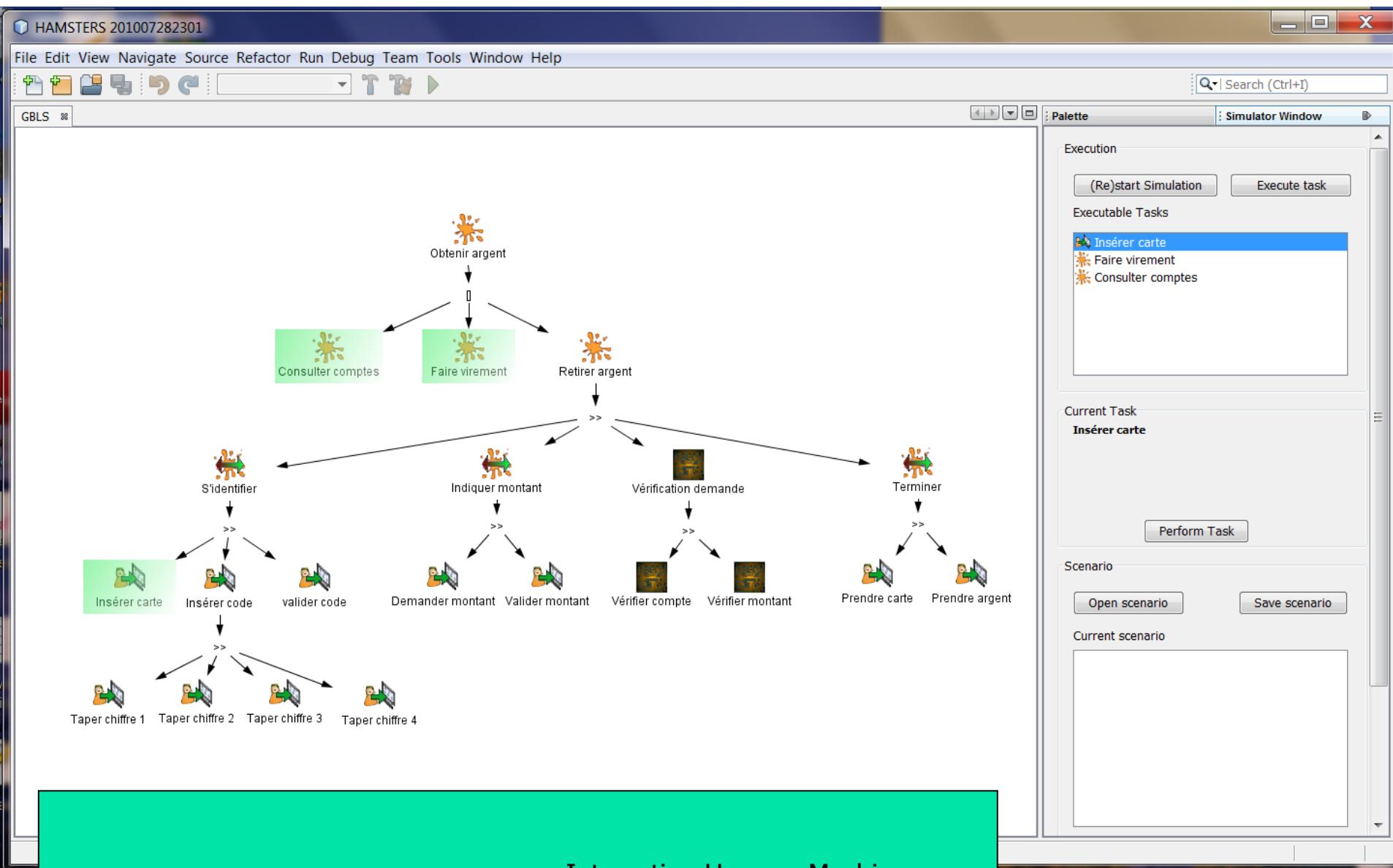


GBLS: Modèle de tâches

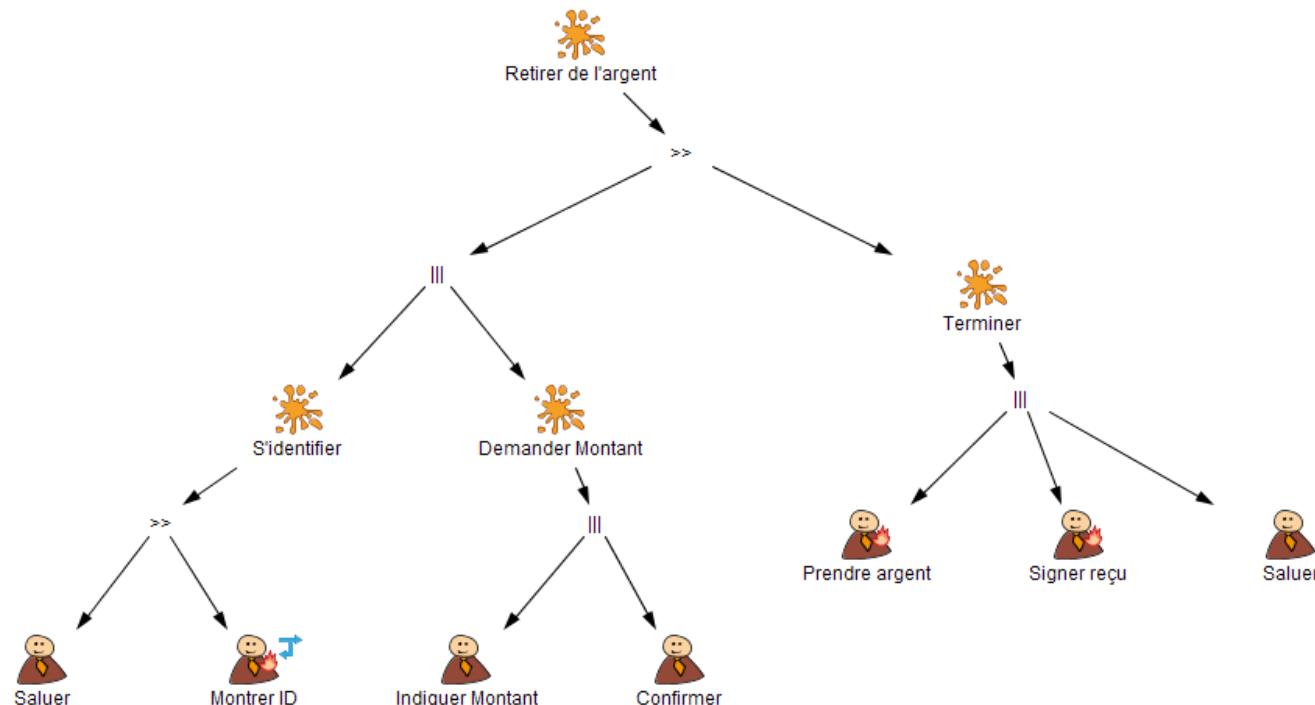


GBLS : Erreurs utilisateur



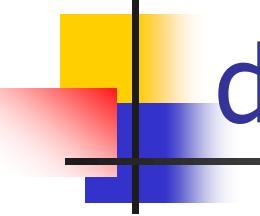


Modèle de tâche au guichet



- Plus d'entrelacement
- Pas d'erreur de 'post completion' (2 modèles de tâche coopératifs)
- Une tâche de politesse

Interaction Homme-Machine



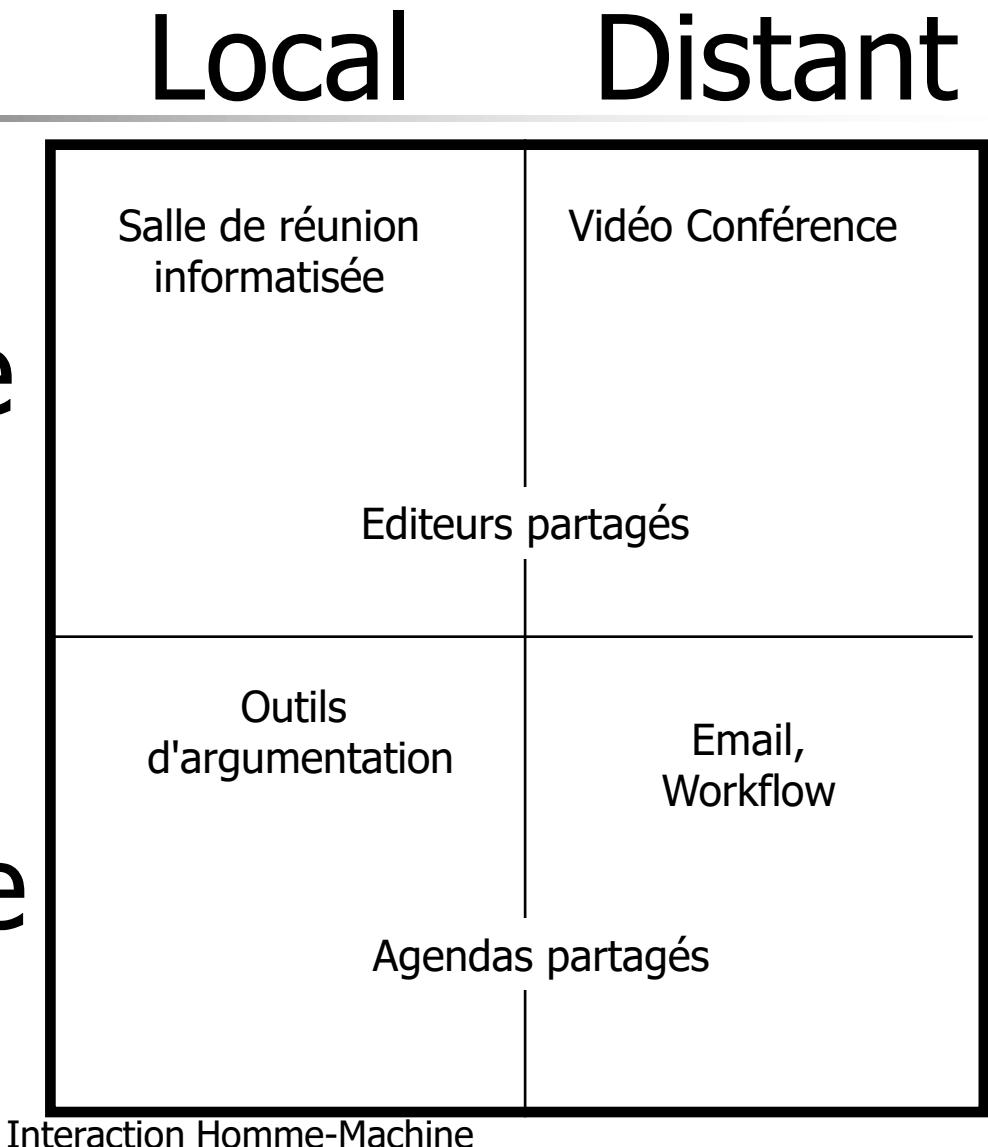
Notation HAMSTERS: catégories des extensions (rappel)

- *Distinction des objets manipulés*
- Fondements
 - Matrice espace/temps
 - Trèfle du collecticiel
 - Taxonomie des tâches de groupe
- Extensions
 - Tâches de groupe
 - Rôles et acteurs associés à une activité
 - Tâches de coordination

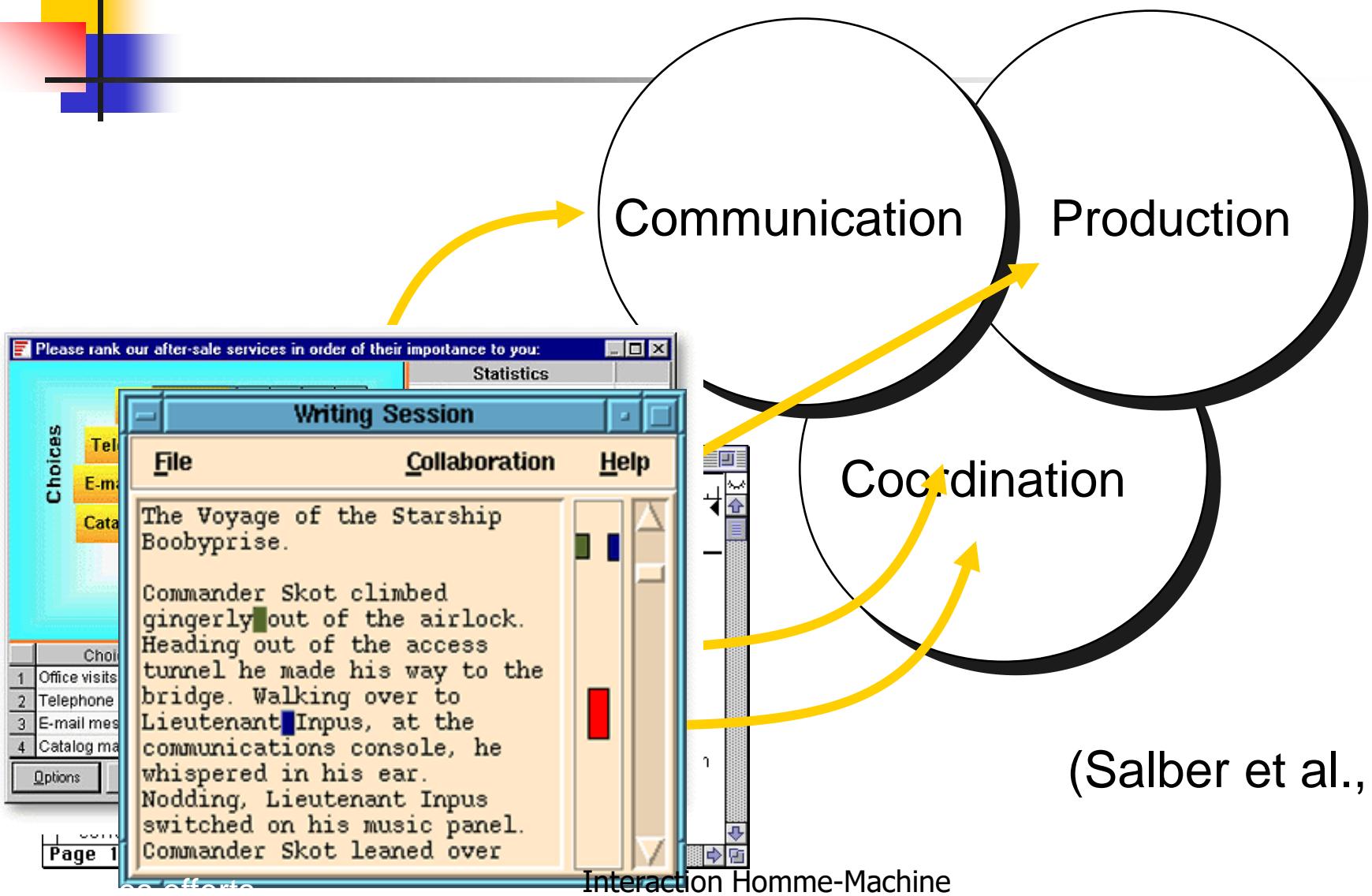
Matrice Espace-Temps

Synchrone
Asynchrone

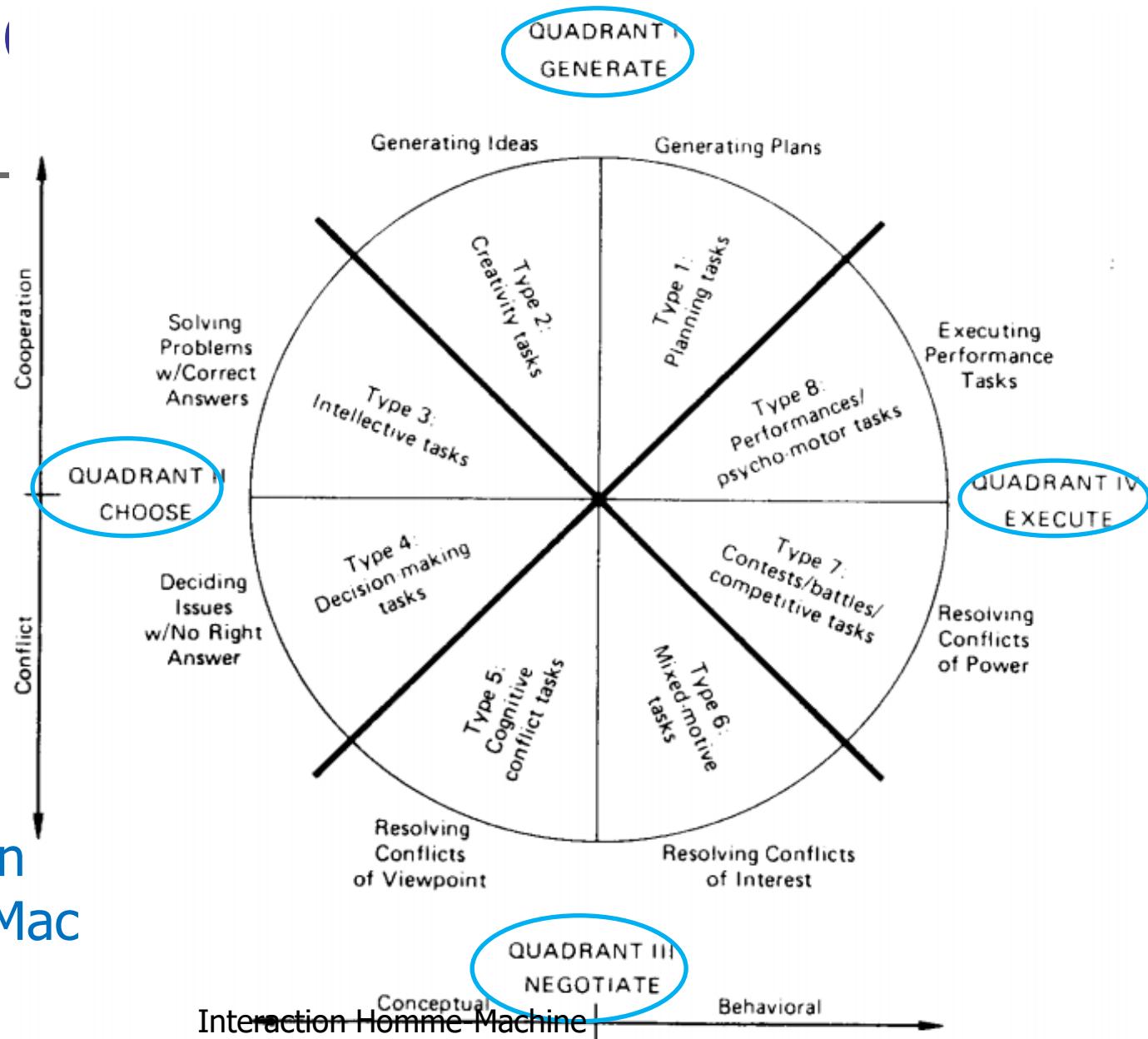
(Ellis C. 1991)



Le trèfle fonctionnel



93 Taxonomie des tâches de groupes



Ref: Group Task circumplex – in Groups: Interaction and performance Mac Grath 1984

Tâche de group: Group task

1 tâche effectuée par plusieurs personnes dans un but commun (provisoire ou définitif)

- Group task
 - Abstract, human, interactive, hybrid
- Aspect espace
 - Synchrone, asynchrone
- Aspect temps
 - Local, distant
- Associations entre tâches de groupe
 - Mise en relation des tâches entre modèles
 - Tableau de correspondances
 - Co-exécution liée à cette mise en correspondance



Abstract Group Task



User Group Task



Interactive Group Task



Hybrid Group Task



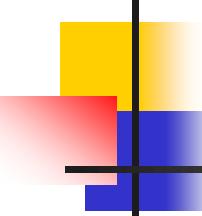
Human group task



Human group task

Raffinement des tâches de groupe

- 1 tâche de groupe abstraite peut être raffinée en 1 ou plusieurs tâches de différents types
- 1 tâche de groupe d'utilisateurs peut être raffinée en 1 ou plusieurs tâches humaines et coopératives



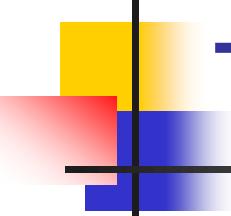
Rôle et Acteur

- Rôle
 - 1 rôle regroupe un ensemble de buts à atteindre (ensemble de modèles de tâche) **dans le cadre une responsabilité**
- Acteur
 - Représente un individu capable d'effectuer un ensemble de tâches
 - Caractéristiques physiques, physiologiques...
 - Connaissances, compétences, expérience...
 - 1 acteur peut se voir assigner 1 ou plusieurs rôles
 - Cette assignation peut être de type principal ou secondaire(s)
 - Statique/dynamique (rôle activé en fonction de conditions)
 - 1 acteur peut se voir assigner 1 ou plusieurs buts

Rôles et acteurs: conséquences

- 1 but à atteindre peut impliquer plusieurs rôles
- L'organisation définit la relation entre acteur(s) et rôle(s)
- 1 but (et ses tâches associées) n'est pas obligatoirement lié à un rôle: cas de la ***subroutine***

→ Aspects méthodologiques liés à la notation



Tâche de coordination

2 ou plusieurs tâches individuelles distinctes, synchronisées et effectuées par différents acteurs

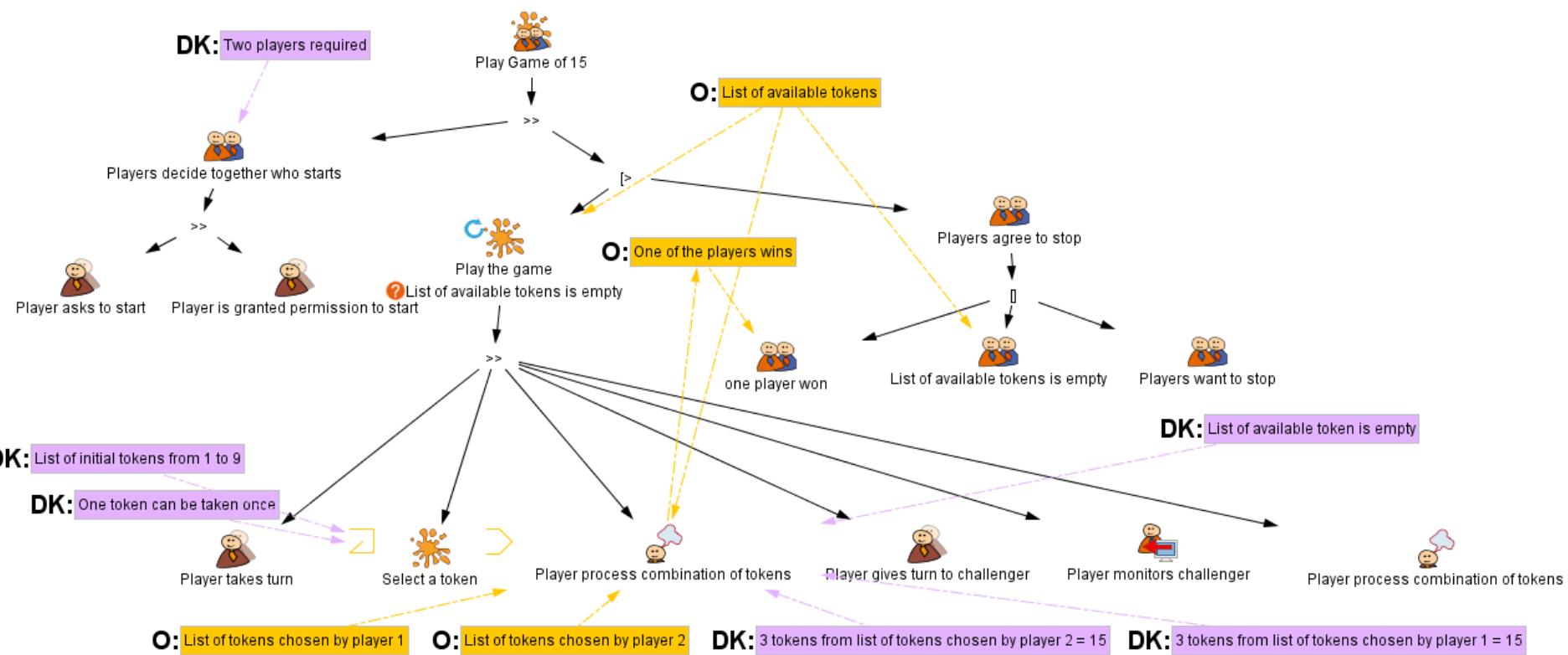
- Bloquante
- Aspects espace et temps



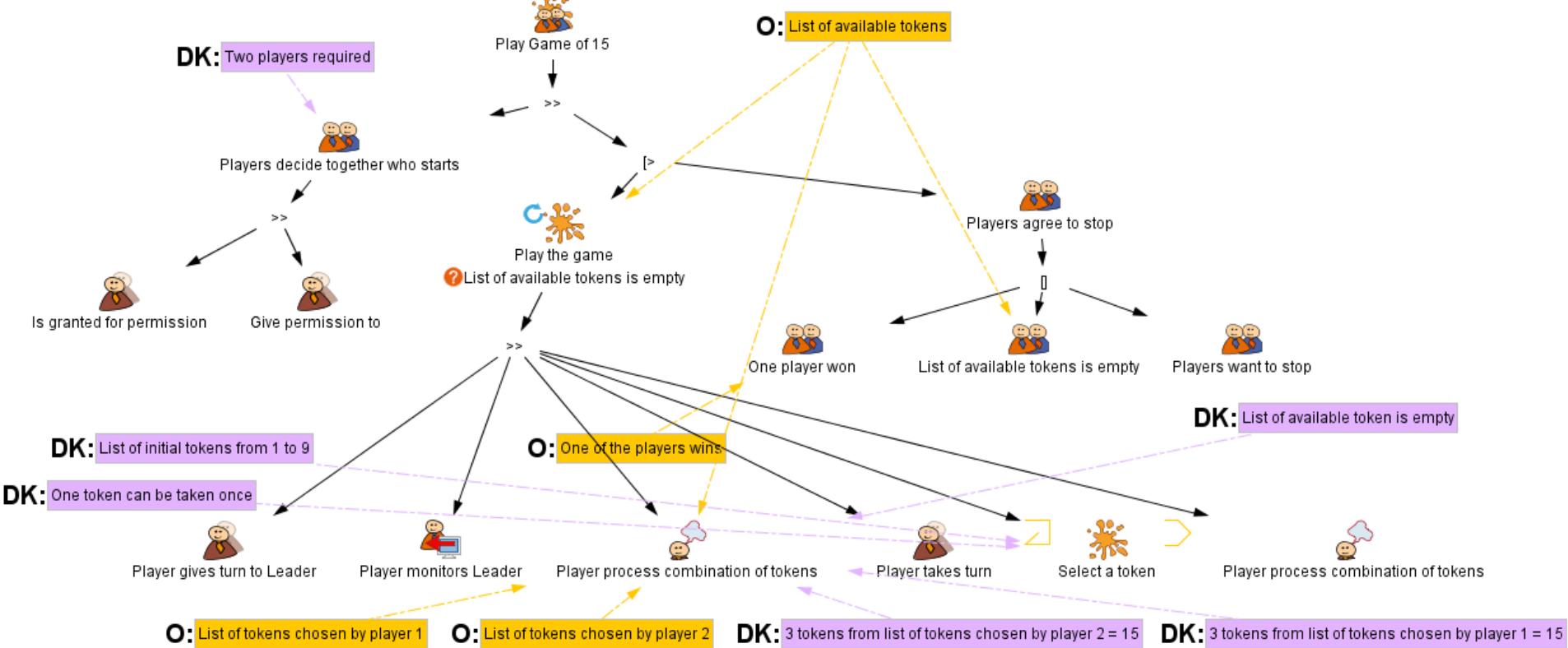
Coordination Task

- Associations entre tâches individuelles coopératives
 - Mise en relation des tâches entre modèles
 - Tableau de correspondances
 - Co-exécution liée à cette mise en correspondance

Leader

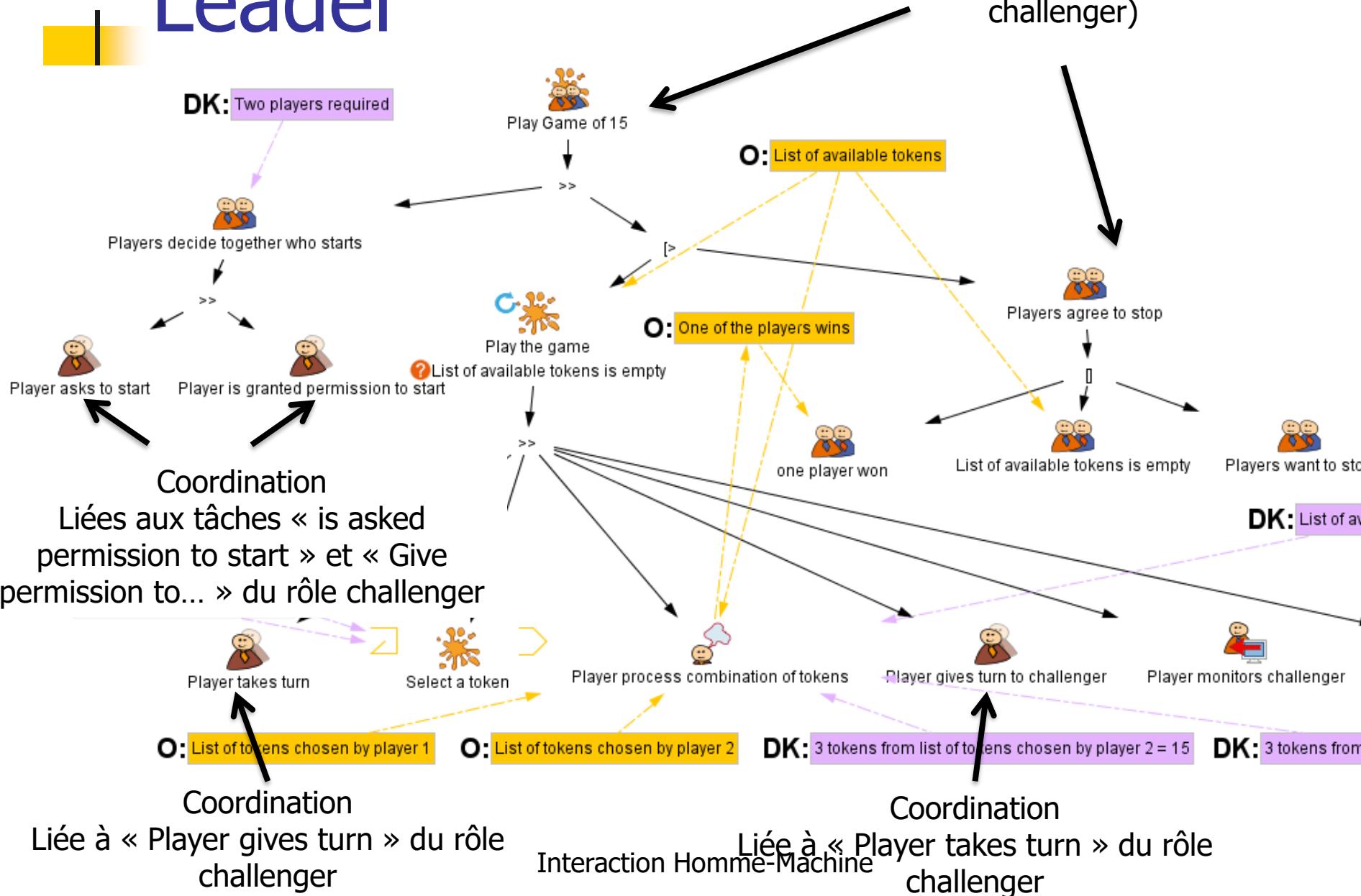


Challenger

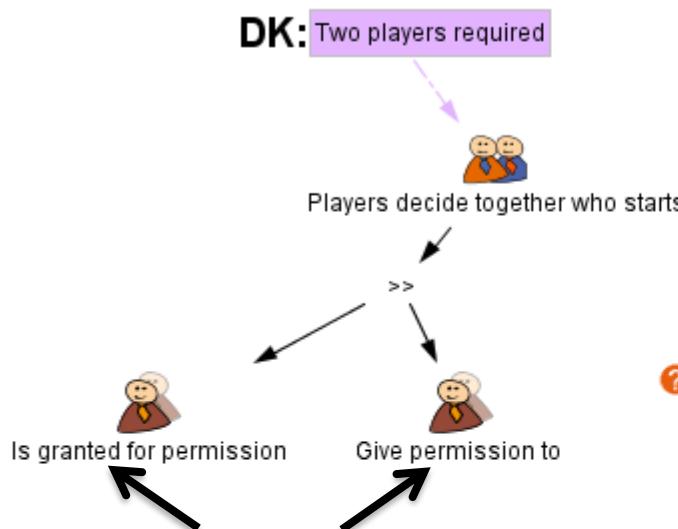


Leader

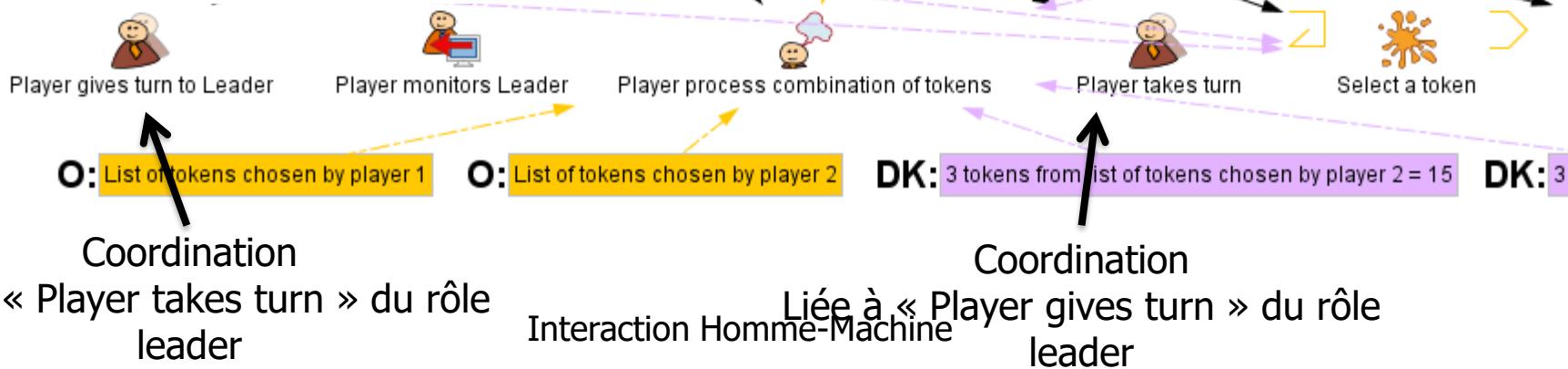
Tâches de groupe
(liées aux tâches équivalentes dans le rôle challenger)

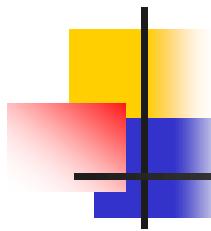


Challenger



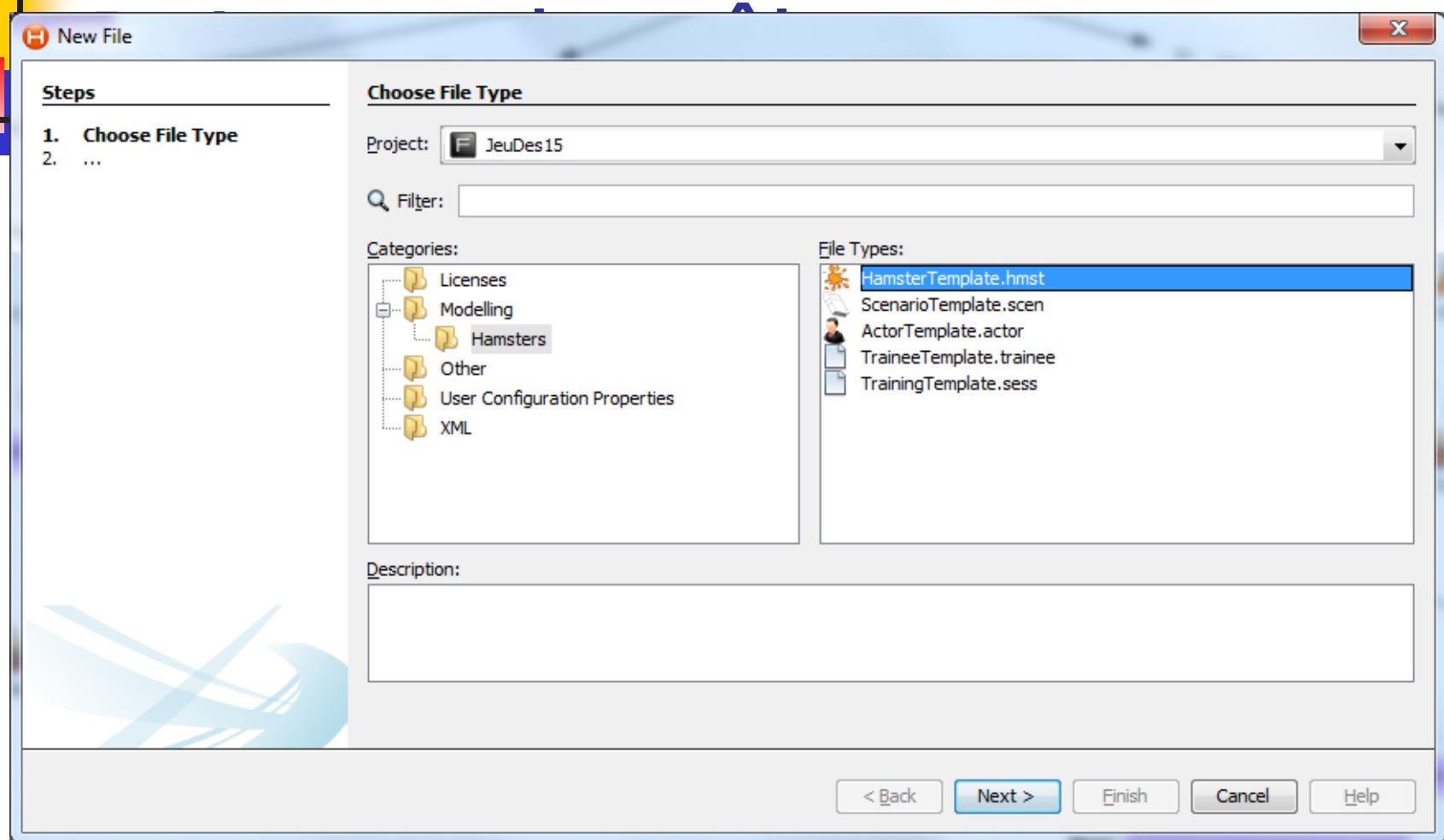
Liées aux tâches « Give permission to... » et « is asked permission to start » du rôle leader

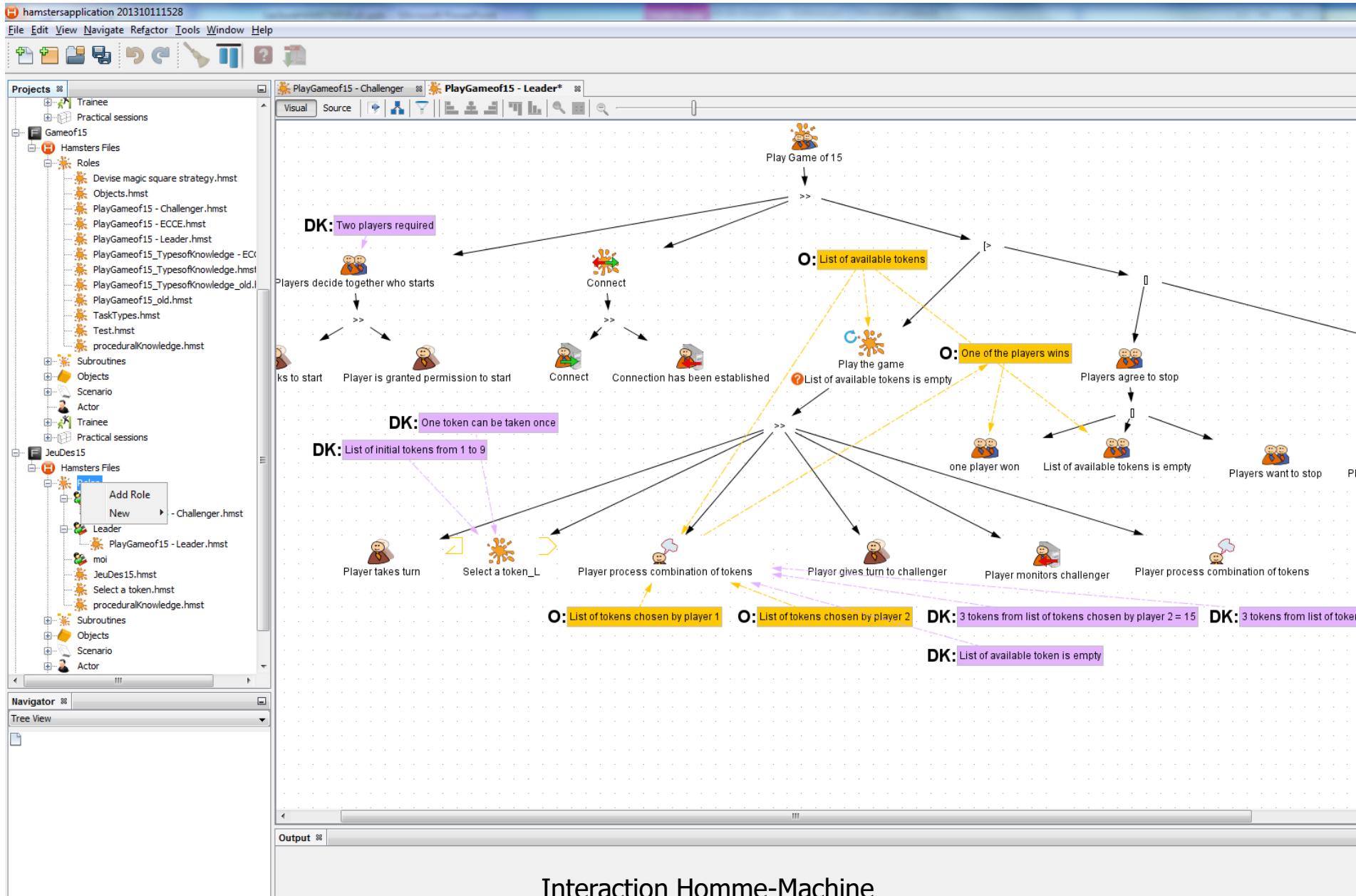




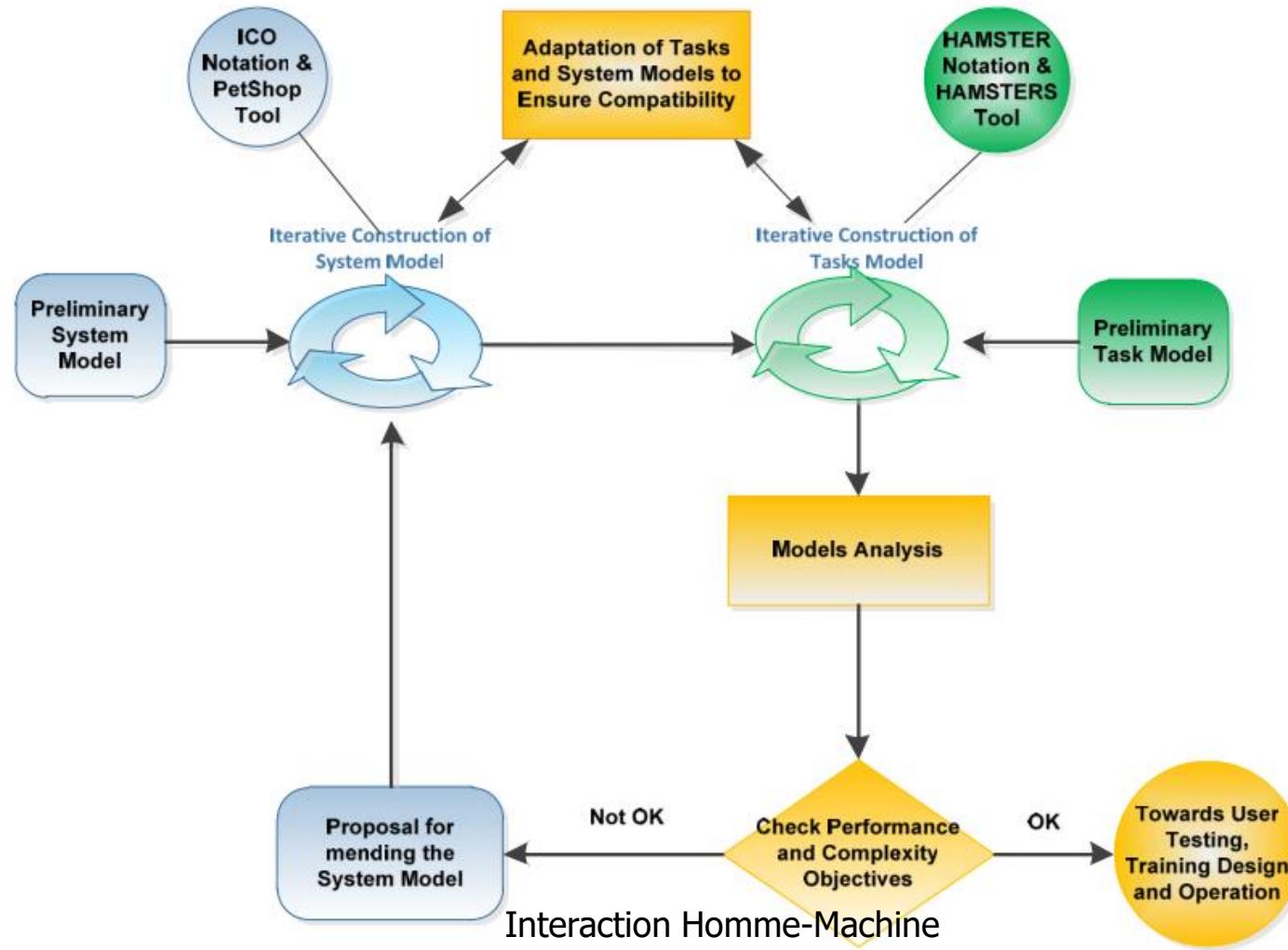
Création des rôles

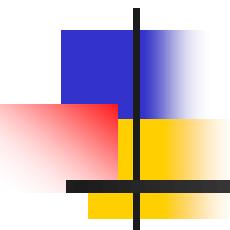
- Ajouter un rôle (bouton droit)
- Ajouter un fichier tempalte HAMSTERS
- Définir le fichier en rôle principal (à chaque fois [Martinie 2013])
- Connecter les tâches de coordination





Process

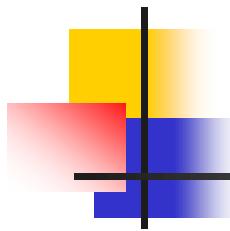




Les problèmes non résolus par la modélisation des tâches

Limites des approches courantes en conception d'IHM

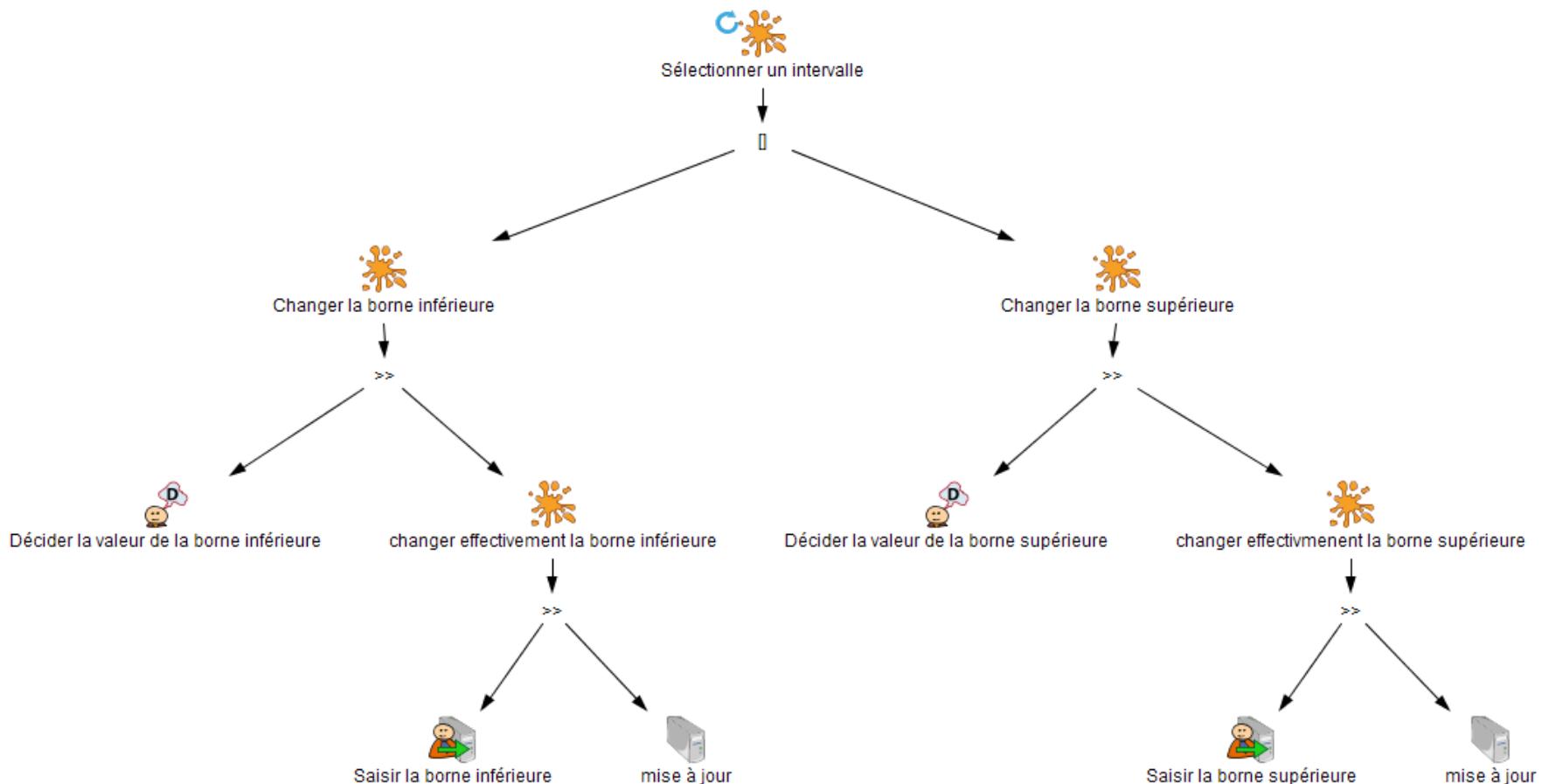
- UML est dirigé uniquement vers la modélisation des composants du systèmes
- Peu d'outils sont disponibles
- Pas d'outils permettant d'exécuter les modèles pour comprendre leur fonctionnement



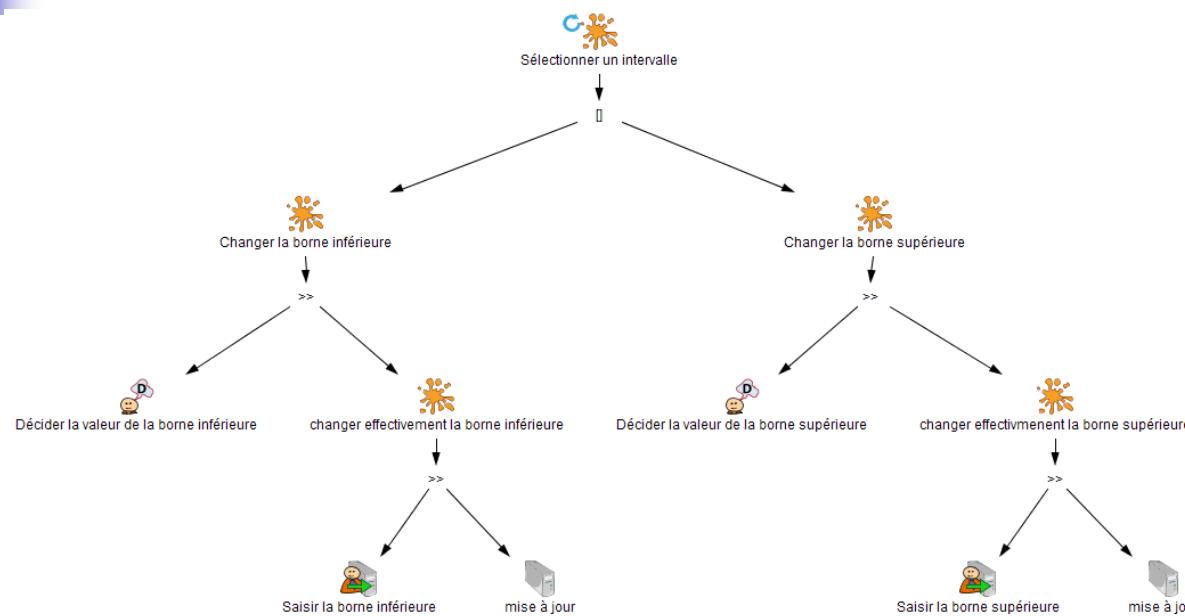
Limites des modèles de tâche (1)

- Définir une fenêtre contenant des objets graphiques (widgets) permettant de sélectionner un intervalle de valeurs
- Valeurs comprises en 1.0 et 2.0
- Faire l'analyse de tâche

Limites des modèles de tâche (2): Arbre abstrait sel. intervalle



Limites des modèles de tâche (3): Première conception



Form1

Current min value

Current max value

Range Selector

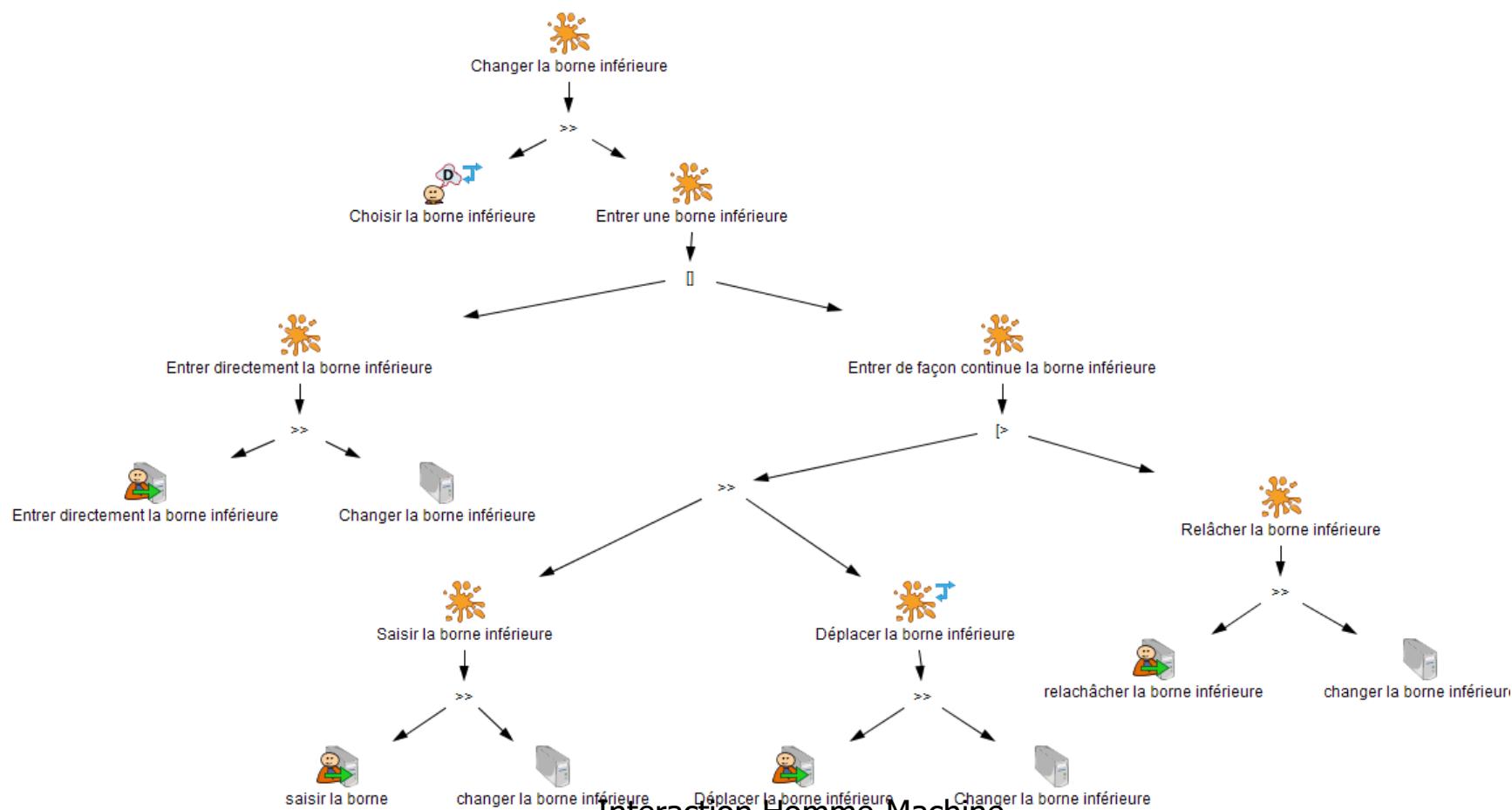
1.0 2.0

Form1

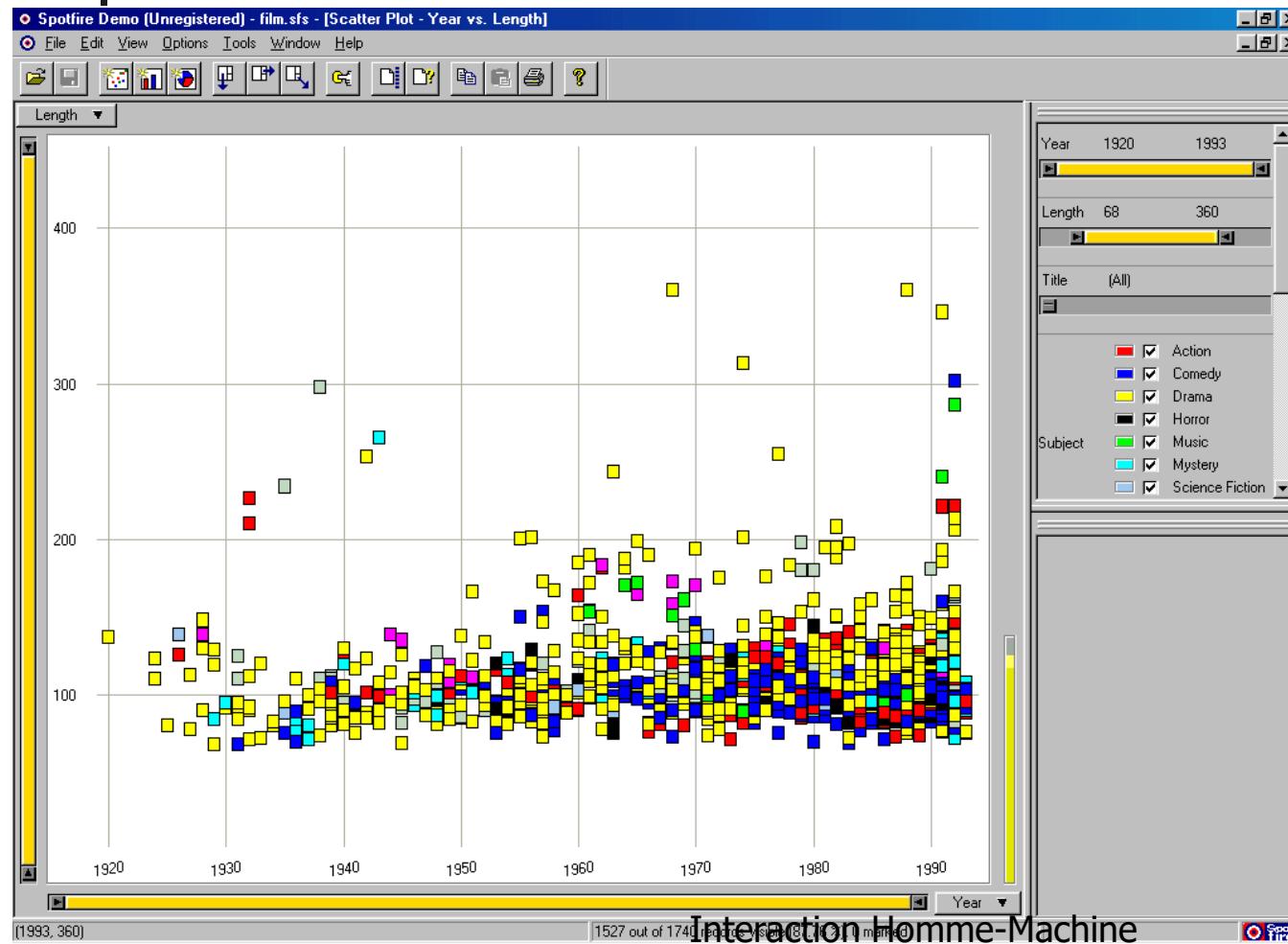
Current min value

Current max value

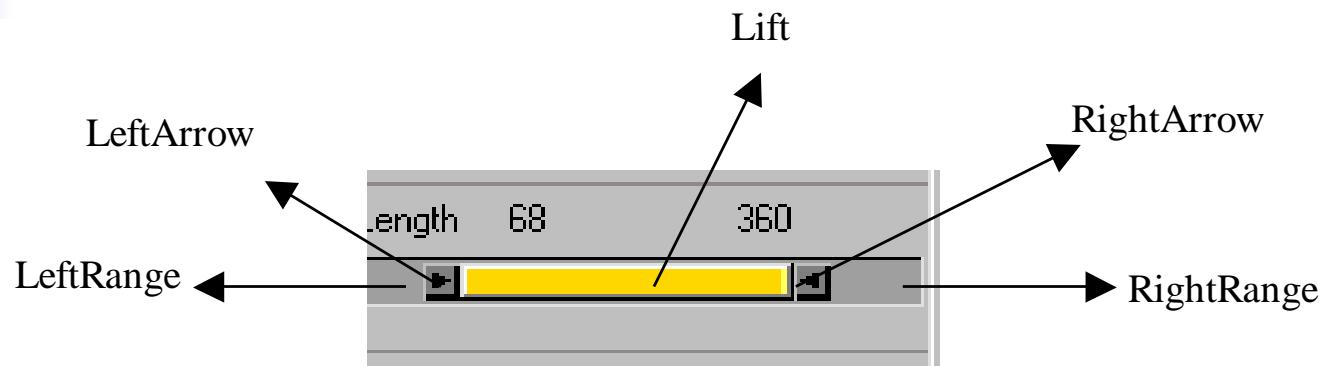
Limites des modèles de tâche (4): Sélection borne inférieure



Limites des modèles de tâche (5)

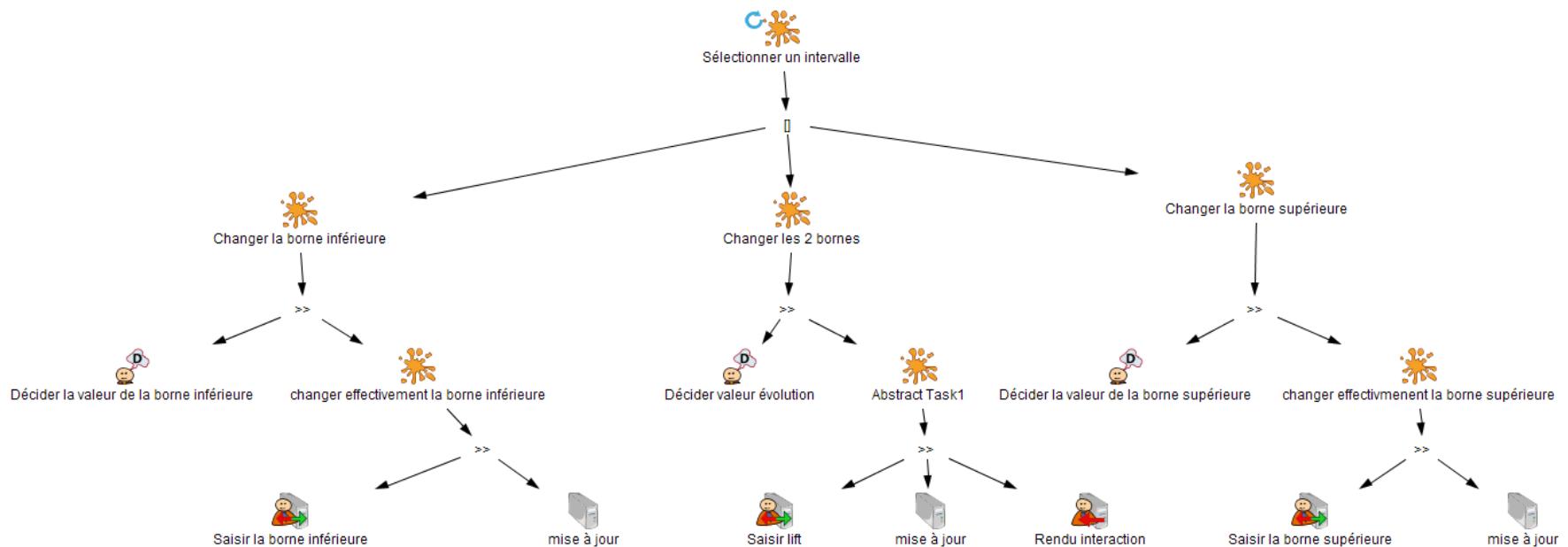


Limites des modèles de tâche (7)



- Down sur LeftRange en X,Y ;
- Down sur RightRange en X,Y ;
- Drag sur LeftArrow en X,Y ;
- Drag sur RightArrow en X,Y ;
- Drag sur Lift en X,Y.

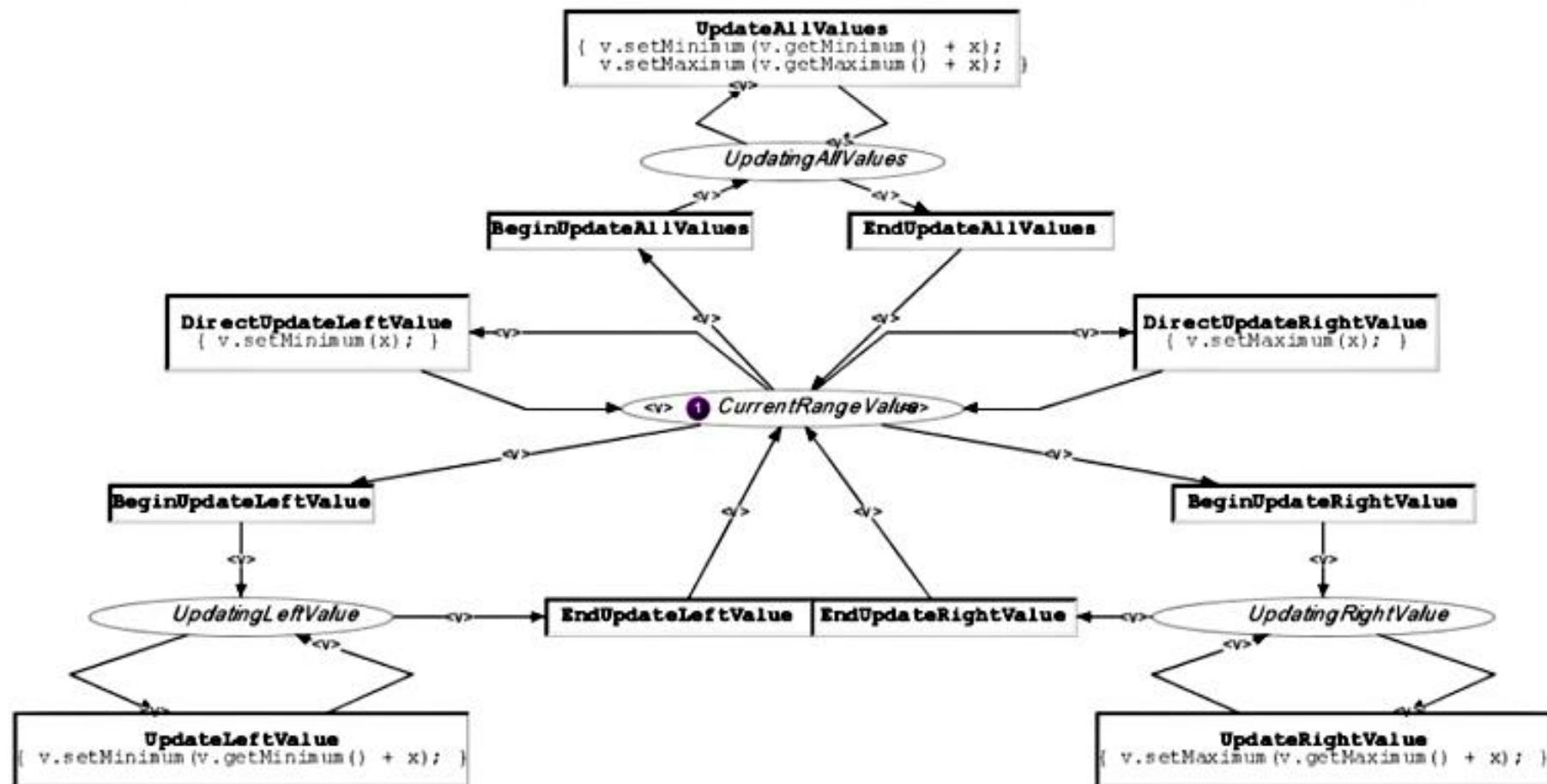
Limites des modèles de tâche (6): Modélisation 2 bornes (lift)



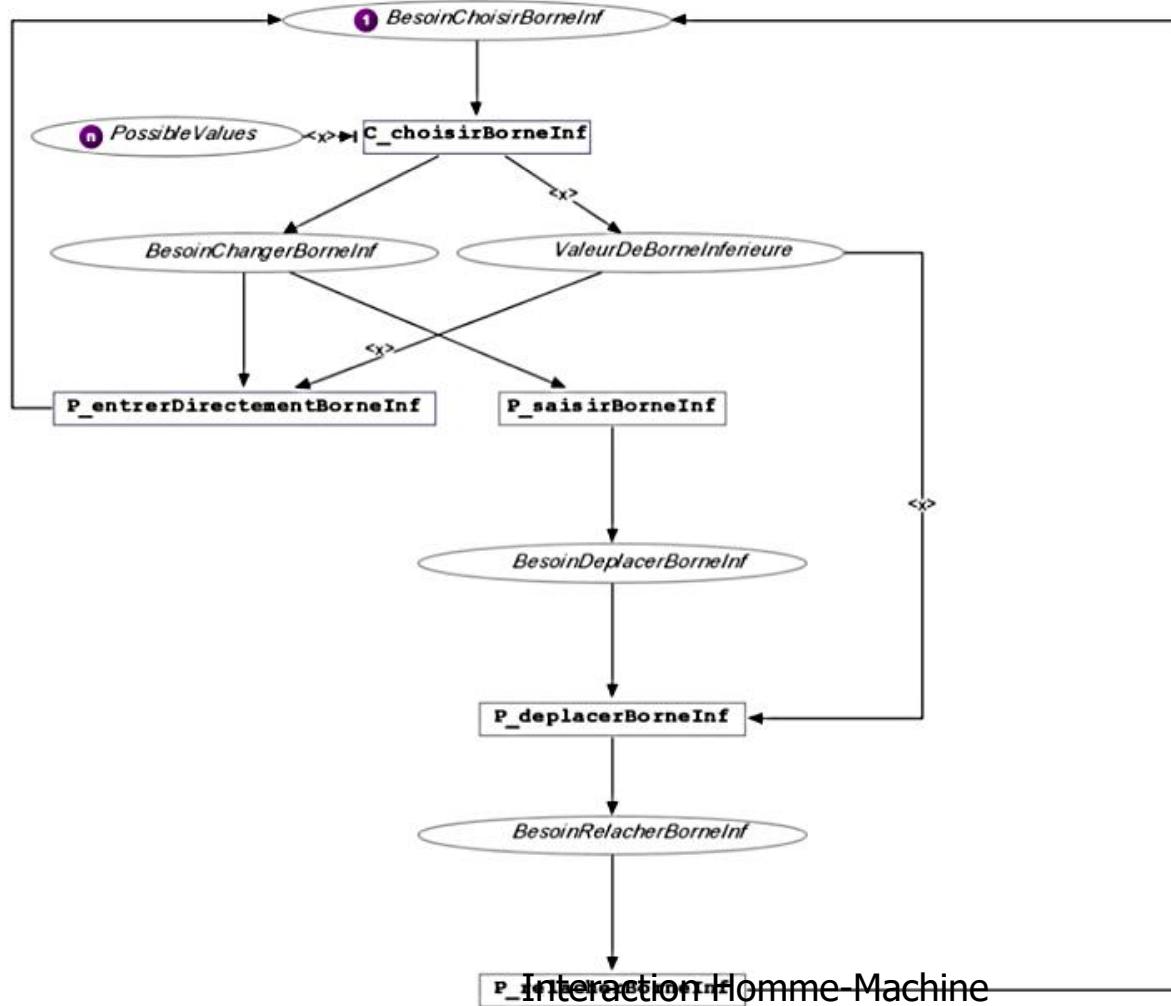
Approche de modélisation (top Down – Bottom Up)

- Démarche descendante
 - On commence par le travail des utilisateurs
 - On modifie les modèles en raffinant
 - On crée l'interface à partir de ces informations
- Démarche ascendante
 - On part des objets
 - Documents papier
 - Interfaces applications interactives
 - On abstrait pour chercher les tâches et les buts
- Dans tous les cas on poursuit par une conception centrée utilisateur

Limites des modèles de tâche (8)

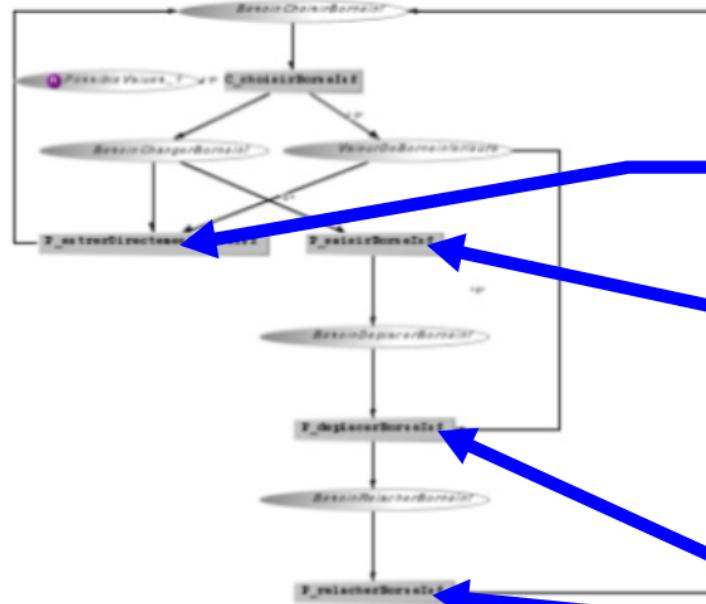


Lien entre modèle de tâches et comportement du système (1)

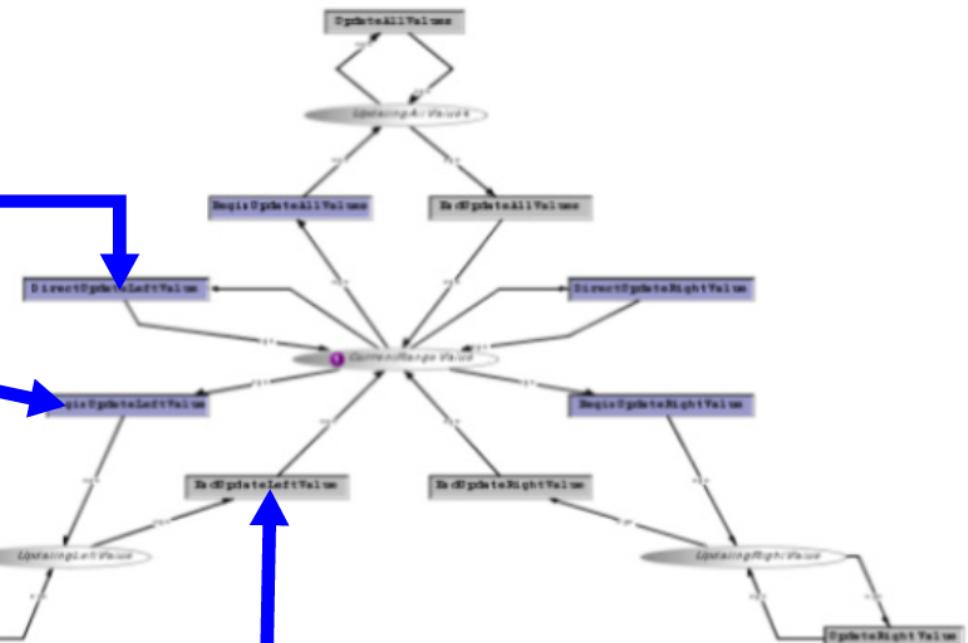


Lien entre modèle de tâches et comportement du système (2)

Modèle de tâche

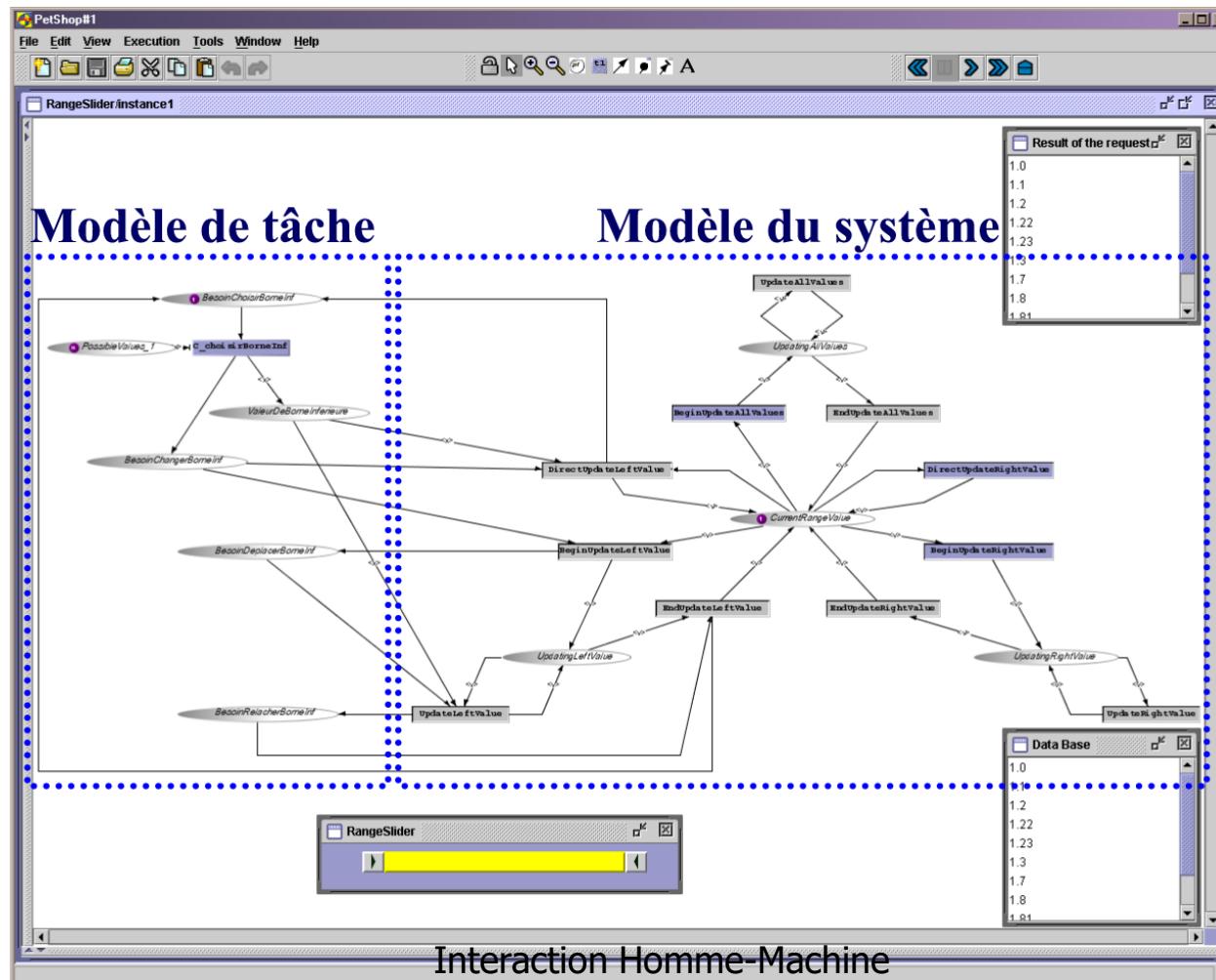


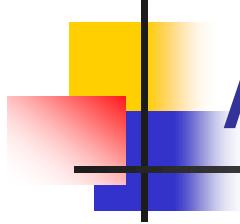
Modèle du système



Interaction Homme-Machine

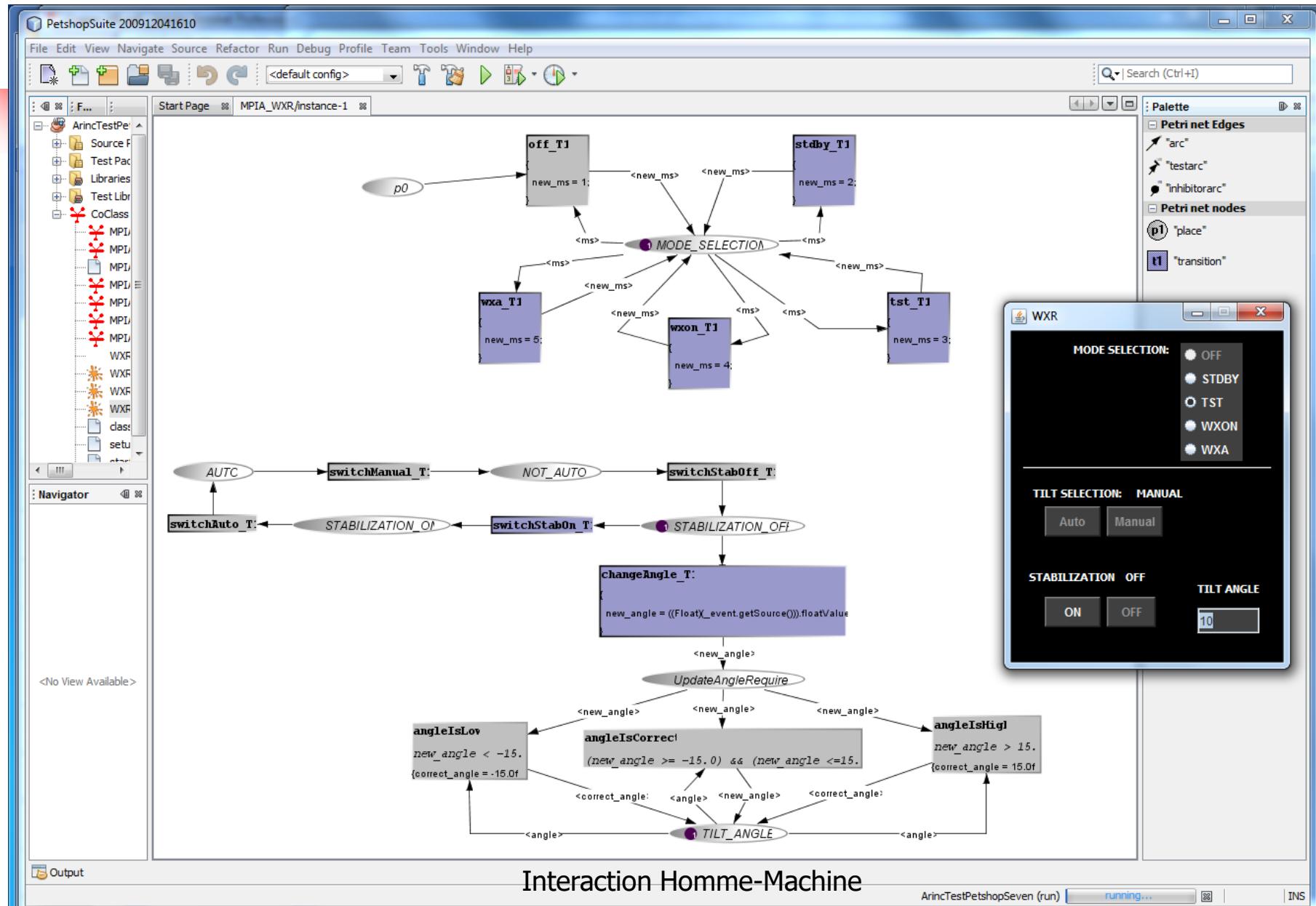
Lien entre modèle de tâches et comportement du système (2)



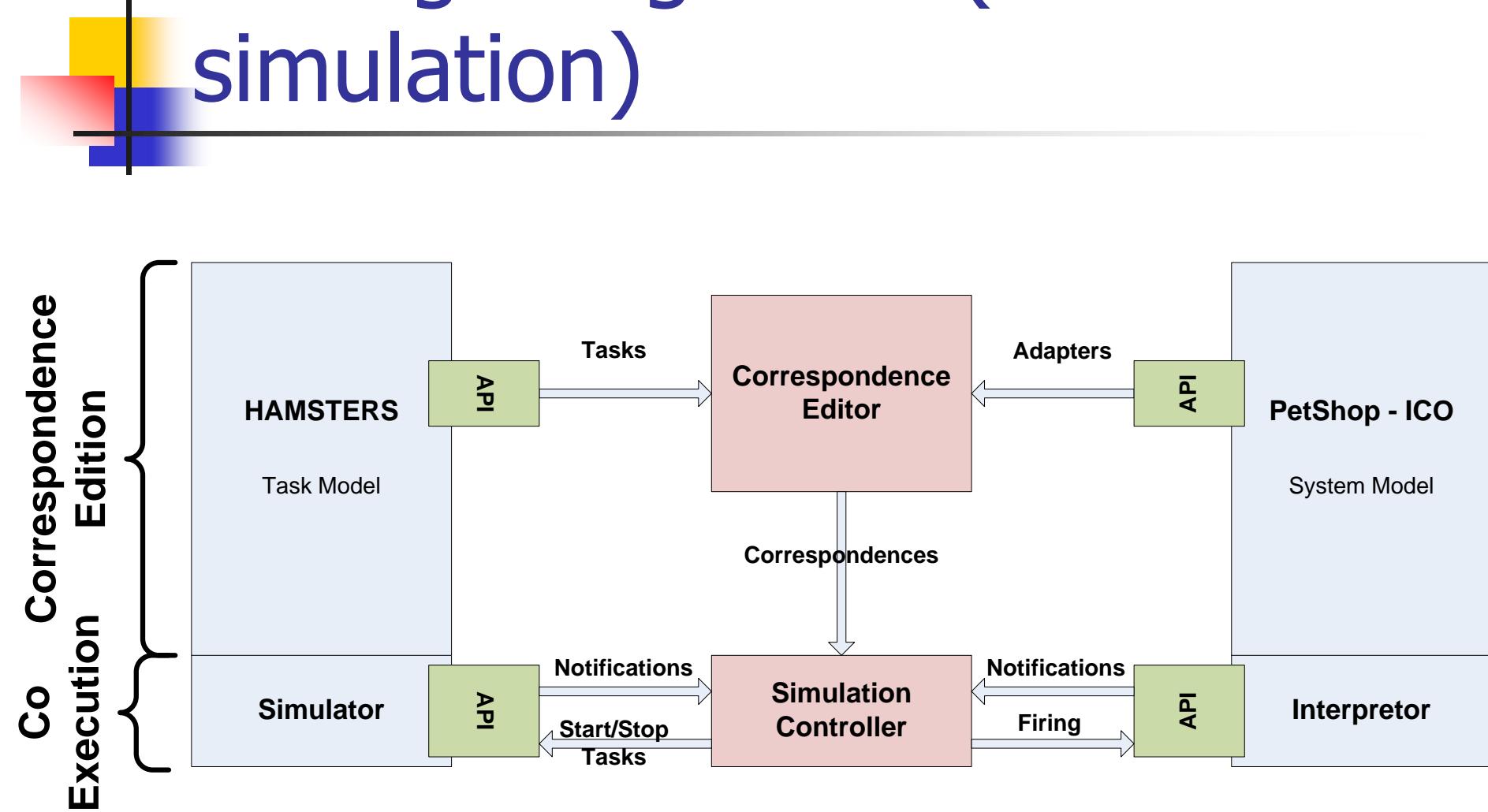


Approche Co-exécution

122 PetShop and the system model



Strong integration (co-simulation)



Correspondance Definition

Input correspondences [Add Input Correspondence](#) [Remove Selected](#)

Interactive Input Tasks	Event Handler
<input type="checkbox"/> switch to WXON	wxon
<input type="checkbox"/> edit angle	changeAngle
<input type="checkbox"/> switch OFF	off
<input type="checkbox"/> select auto	switchAuto
<input type="checkbox"/> stabilization off	switchStabOff
<input type="checkbox"/> switch to TST	tst
<input type="checkbox"/> switch to WXA	wxa
<input type="checkbox"/> switch to STDRY	stdhv

Output correspondences [Add Output Correspondence](#) [Remove Selected](#)

Interactive Output Tasks	Place	Place Event
<input type="checkbox"/> check updated value	TILT_ANGLE	Token added

Correspondence coverage  75 %

Warnings	Unused Interactive Input Tasks	0 out of 10
	Unused Interactive Output Tasks	0 out of 1
	Potentially Unused Event Handlers	0 out of 10
	Potentially Unused Rendering	7 out of 8

 [Output](#)

Interaction Homme-Machine

ArincTestPetshopSeven (run) | running... | INS |

PetshopSuite 200912041610

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Start Page MPIA_WXR/instance-1 WXR_TaskModel

ArincTestPetShop

Source Pack Test Pack Libraries Test Library CoClass MPIA MPIA MPIA MPIA MPIA MPIA MPIA WXR WXR WXR WXR classi setup start

mode selection

off_T1 standby_T1 wxa_T1 wxon_T1 stat_T1

new_ms = 1 new_ms = 2 new_ms = 5 new_ms = 4 new_ms = 3

new_ms = 1 new_ms = 2 new_ms = 5 new_ms = 4 new_ms = 3

AUTO switchManual_T NOT_AUTO switchStabOff_T switchStabOn_T STABILIZATION_ON STABILIZATION_OFF changeAngle_T

new_angle = (float)event.getSource().floatValue()

angleIsLow new_angle < -15.0; correct_angle = -15.0f; angleIsCorrec new_angle >= -15.0 && new_angle <= 15.0; angleIsBig new_angle > 15.0; correct_angle = 15.0f; TILT_ANGLE

angleIsLow new_angle < -15.0; correct_angle = -15.0f; angleIsCorrec new_angle >= -15.0 && new_angle <= 15.0; angleIsBig new_angle > 15.0; correct_angle = 15.0f; TILT_ANGLE

mode selection

manage WXR setup WXR decide WXR is ready

manage modes

decide mode is correct

change mode

decide change mode

switch to WXA switch to TBT switch to WXX switch to STDBY

select manual stabilization off angle editing

select auto stabilization on

modify angle

decide angle And edit angle check updated value

Interactive Tasks

User Tasks

Perceptive Task

Motor Task

Cognitive Task

CorrespondanceExecution Window

Warnings

Simulation Mode

- System driven
- Task driven

(Re)start Simulation

Current Scenario

Available tasks

- decide change mode
- decide mode is correct
- decide change tilt angle
- decide WXR is ready

Current Task

No task selected

Perform Task

Output

WXR

MODE SELECTION:

- OFF
- STDBY
- TBT
- WXX
- WXA

TILT SELECTION: AUTO

Auto Manual

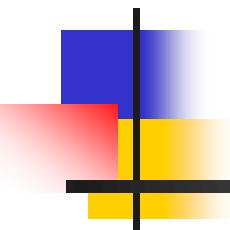
STABILIZATION ON

ON OFF

TILT ANGLE

Angle: 0.0°

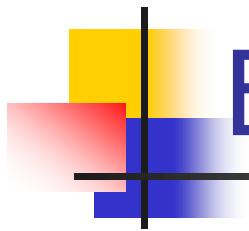
ArincTestPetShopSeven (run) running... INS



Des tâches au système

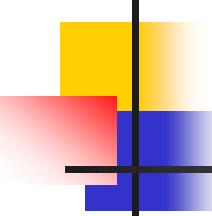
Articles

F. Paterno EHCI 1998 & C. Paris
et al. EHCI 1998



Exemple D.OM de Atos Origin

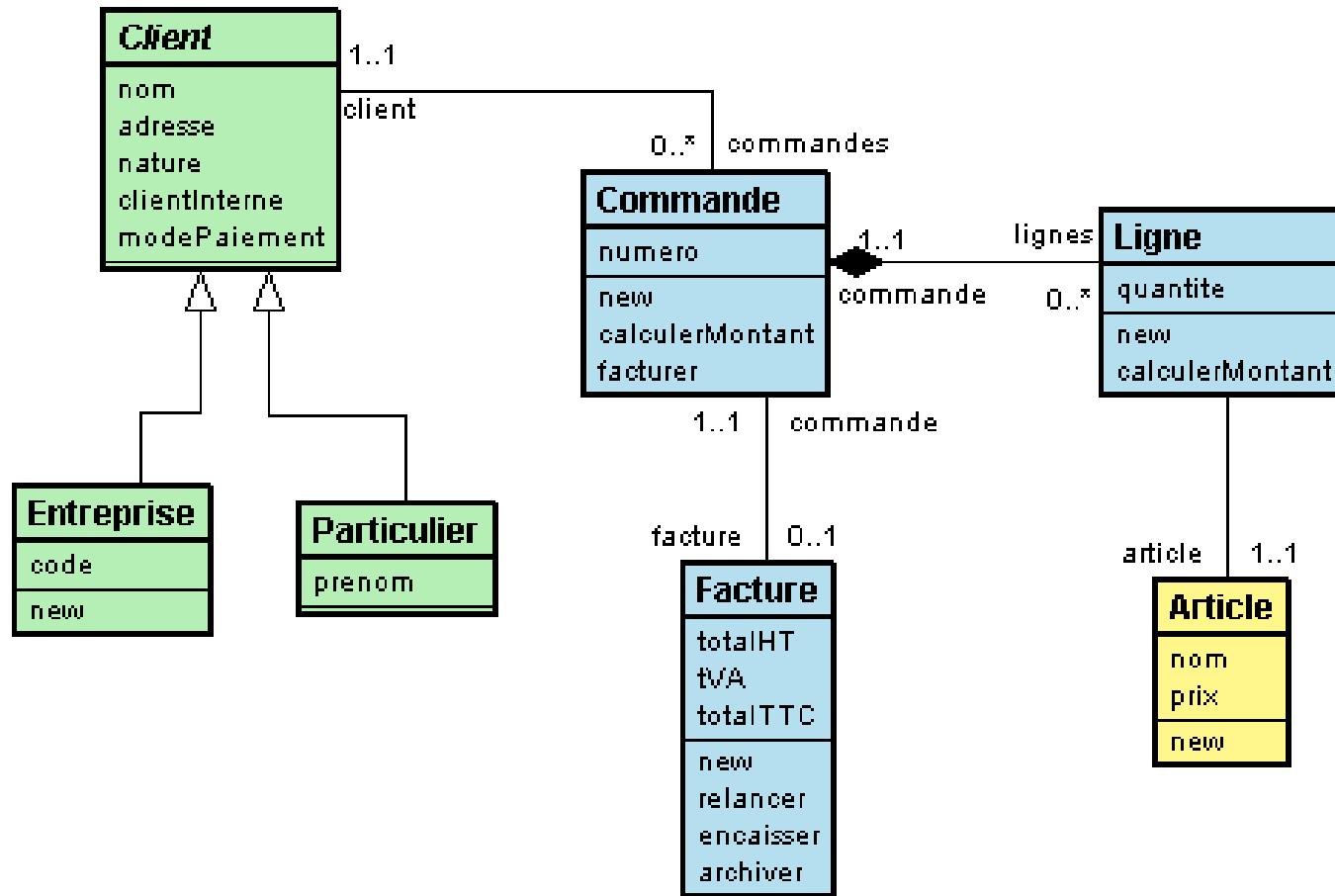
- <http://www.ii.atos-group.com/rhone-alpes/Dom/index.html>
- Génération de l'interface à partir du modèle de classes UML



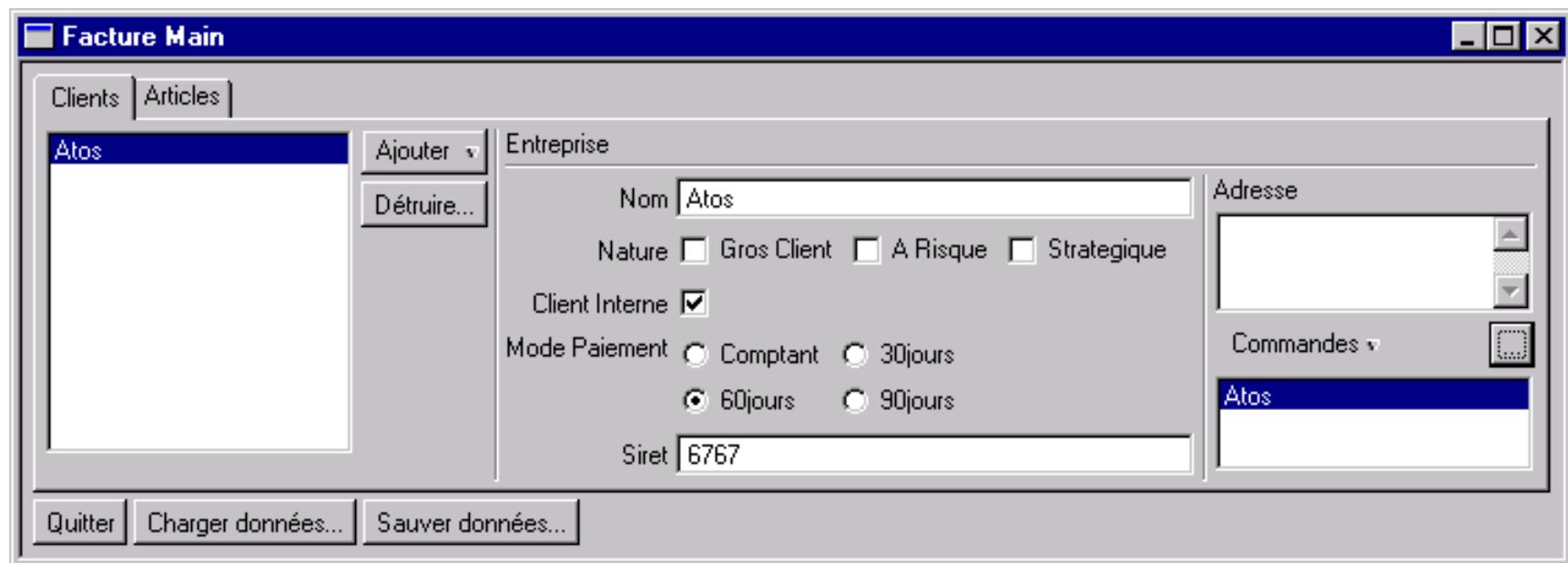
La Pub

- A partir du seul modèle objet (modèle métier), D•OM génère automatiquement et intégralement un prototype fonctionnel permettant de créer et de manipuler les instances d'objets et leurs associations. Les comportements dynamiques sont conformes aux spécifications du modèle.
- **Validation du modèle**
- L'utilisateur dispose de toutes les fonctions lui permettant de faire vivre les instances de son modèle. L'objectif du prototype est de valider le modèle, tout au long de sa conception, par expérimentation. Ceci permet au concepteur de se rendre compte des conséquences de ses choix de modélisation.
- Le prototype est aussi un excellent moyen pour s'assurer au plus tôt que le concepteur et l'utilisateur se sont bien compris. Il permet de vérifier que l'application en cours de développement correspond bien aux besoins.
- **Interface utilisateur**
- L'interface utilisateur graphique est entièrement déduite des caractéristiques du modèle grâce à une base de connaissances. L'interface obtenue est très ergonomique et parfaitement homogène. Elle est complète et matérialise l'intégralité des fonctionnalités présentes dans le modèle objet.
- Bien sûr, le prototype ne préfigure en aucun cas l'interface utilisateur définitive de l'application. Si toutefois l'ergonomie par défaut ne convient pas, il est possible d'enrichir le modèle avec des annotations concernant l'interface utilisateur. Ces annotations sont mémorisées et réutilisées pour reconstruire automatiquement l'interface, après chaque évolution du modèle. On obtient ainsi un prototype très simple à utiliser, parfaitement présentable à l'utilisateur et toujours cohérent avec le modèle.

Modèle de classe

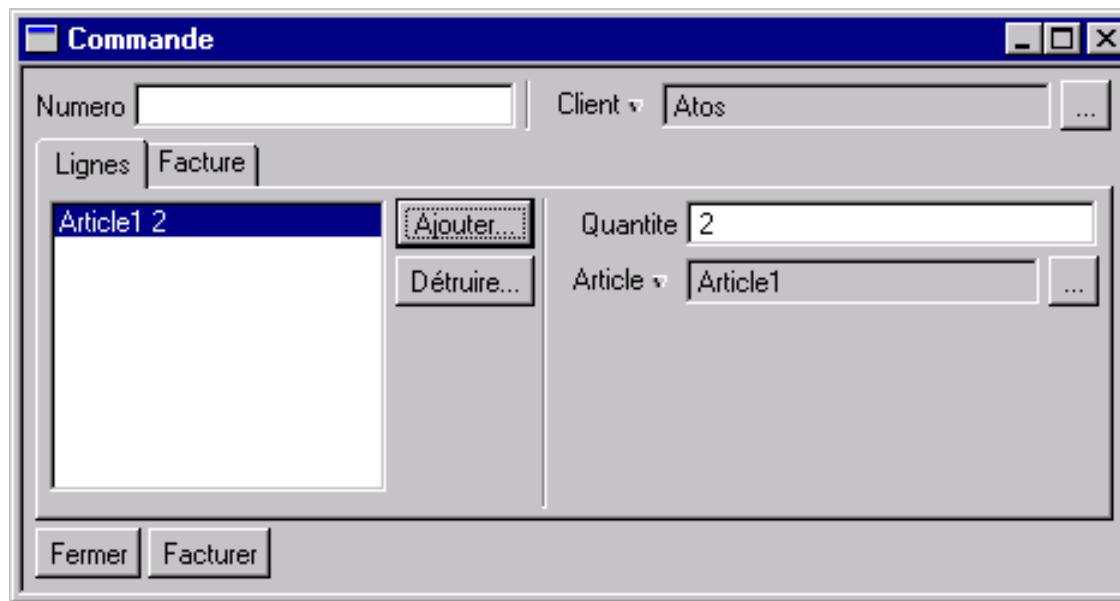


Interface Générée (1)

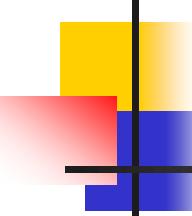


La gestion des clients et des articles s'effectue dans la fenêtre principale

Interface Générée (2)



La fenêtre Commande accessible depuis la fenêtre principale, permet de travailler sur une commande mais aussi sur ses lignes de commandes



Conclusions

- **Jeux d'essais**

Le prototype généré permet de sauvegarder et de recharger des bases d'objets, pour créer des jeux d'essais réutilisables par les versions successives du modèle. Cette gestion de données automatique permet de s'affranchir de l'utilisation d'un SGBD durant la phase de prototypage.

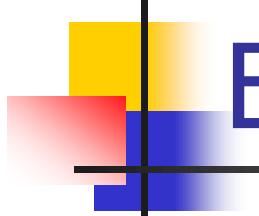
Le composant Storage permet de décrire les entités du modèle objet à sauvegarder. Il permet de remonter des sauvegardes effectuées sur une version précédente du prototype alors que des modifications importantes ont été faites au niveau du modèle objet, (changement de type d'un attribut, suppression ou ajout d'attributs, ...).

- **Diffusion des prototypes**

Les prototypes peuvent être diffusés librement auprès des différents expérimentateurs. D•OM offre une fonction pour délivrer sous une forme compacte les fichiers nécessaires à l'exécution du prototype. Pour l'expérimentation, il n'y a donc pas besoin de disposer de D•OM.

Conclusions et autres sources

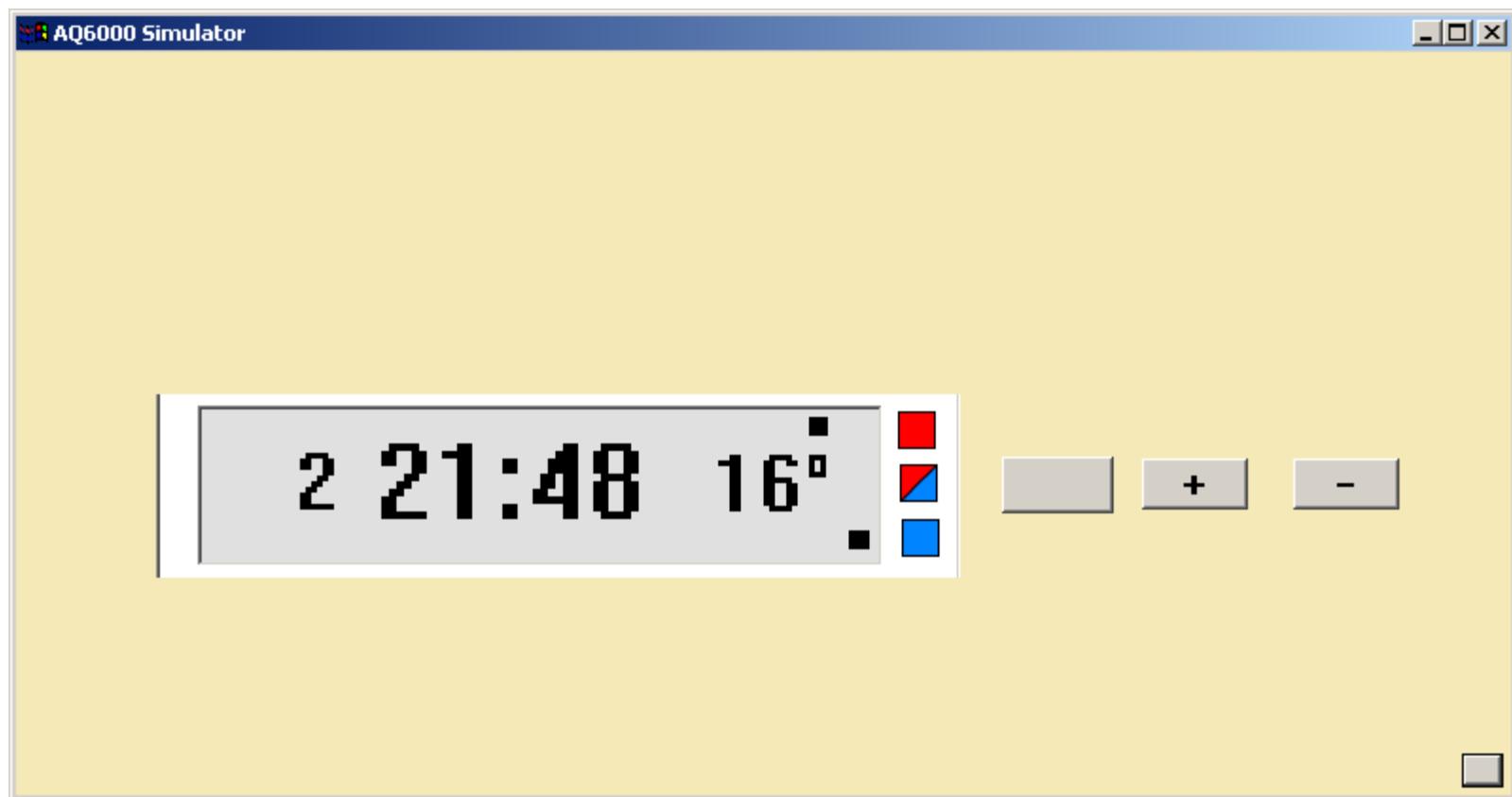
- [Http://giove.cnuce.cnr.it/ctte.html](http://giove.cnuce.cnr.it/ctte.html)
- GUITARE Project
<http://giove.cnuce.cnr.it/guitare.html>
- MEFISTO Project
<http://giove.cnuce.cnr.it/mefisto.html>
- Book on Model-Based Design and Evaluation of Interactive Applications,
Springer Verlag
- fabio.paterno@cnuce.cnr.it



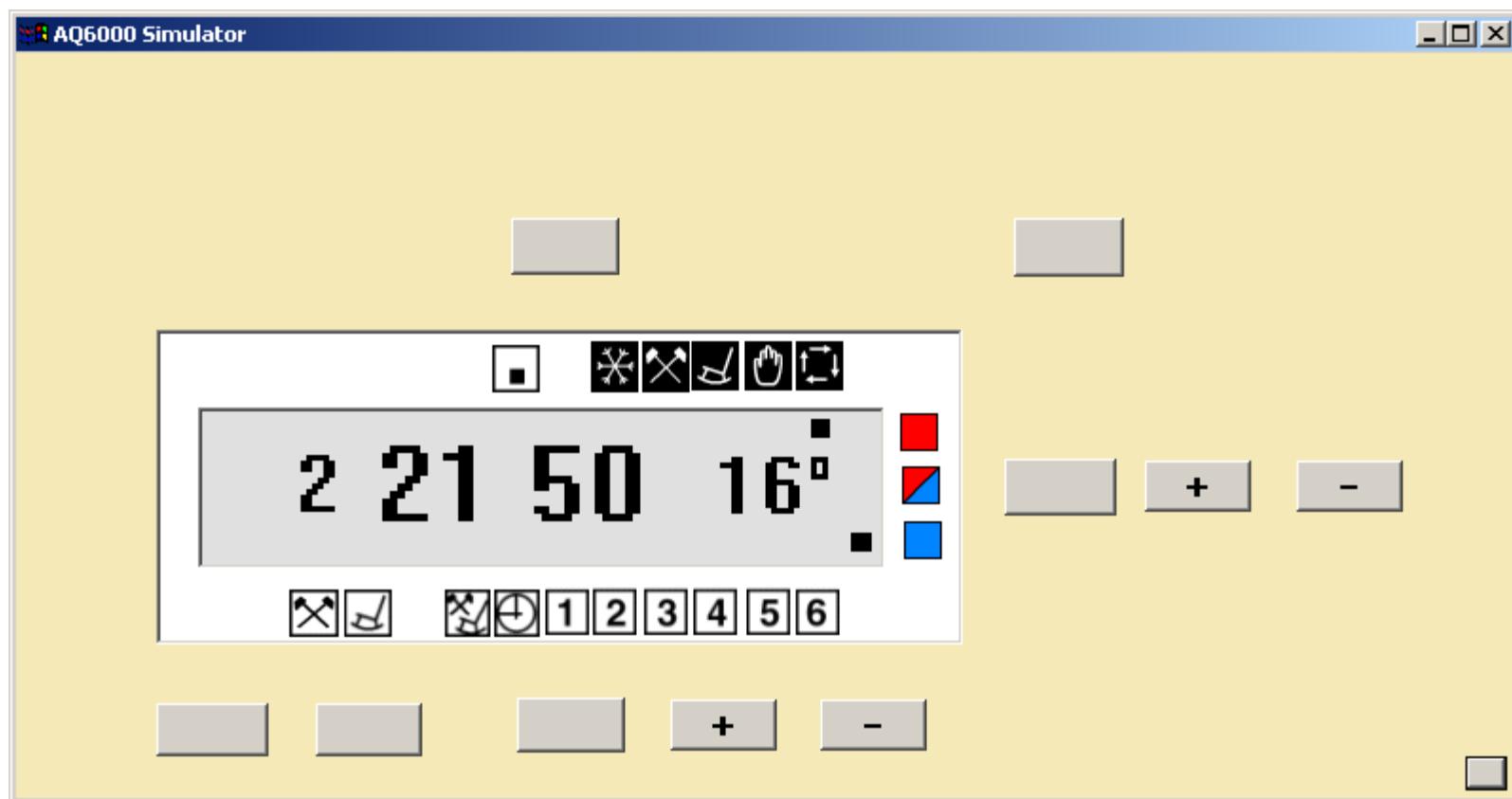
Exercice 2

- Conception d'un système de contrôle de chauffage central
- Version initiale
- Proposer des améliorations

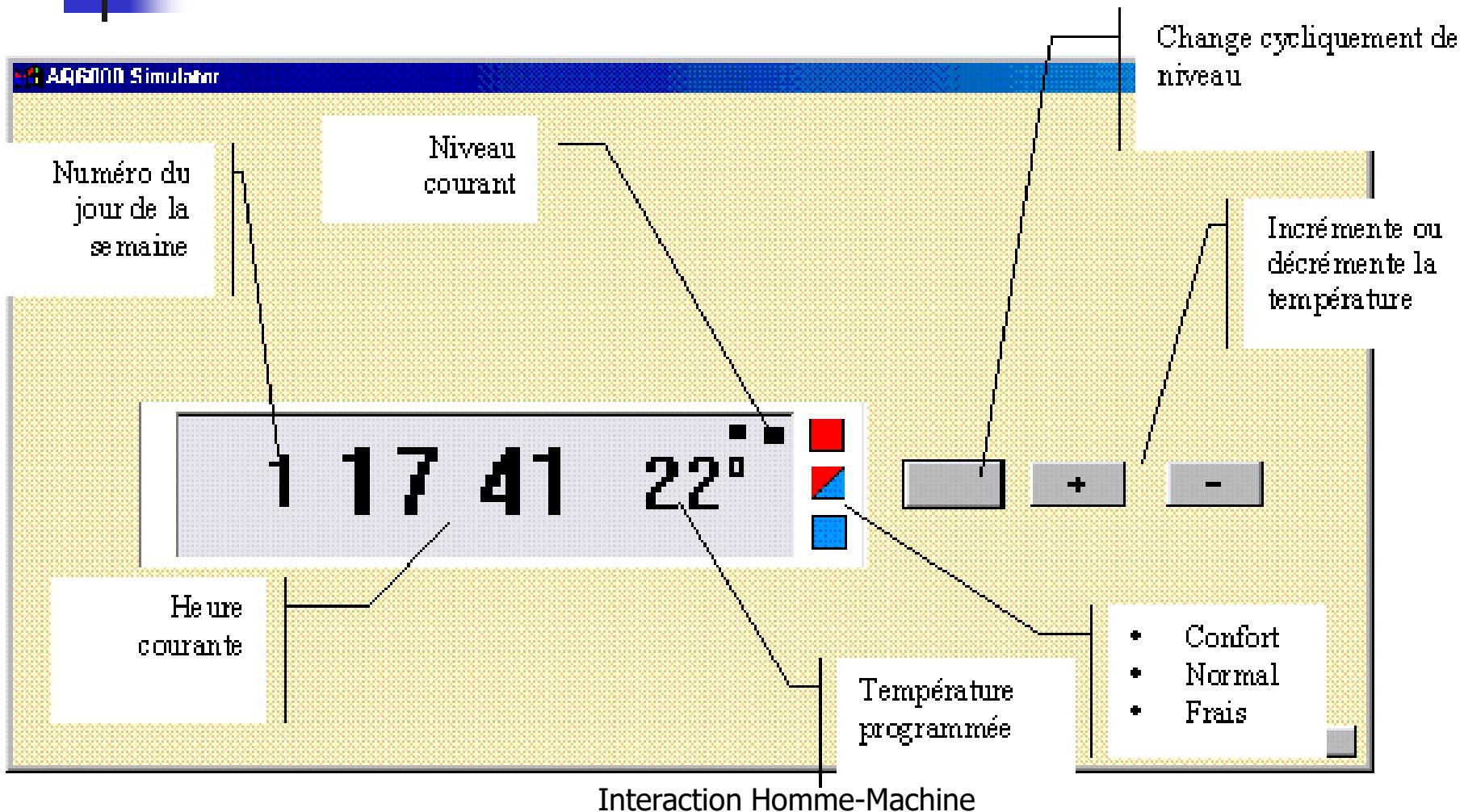
AQ6000



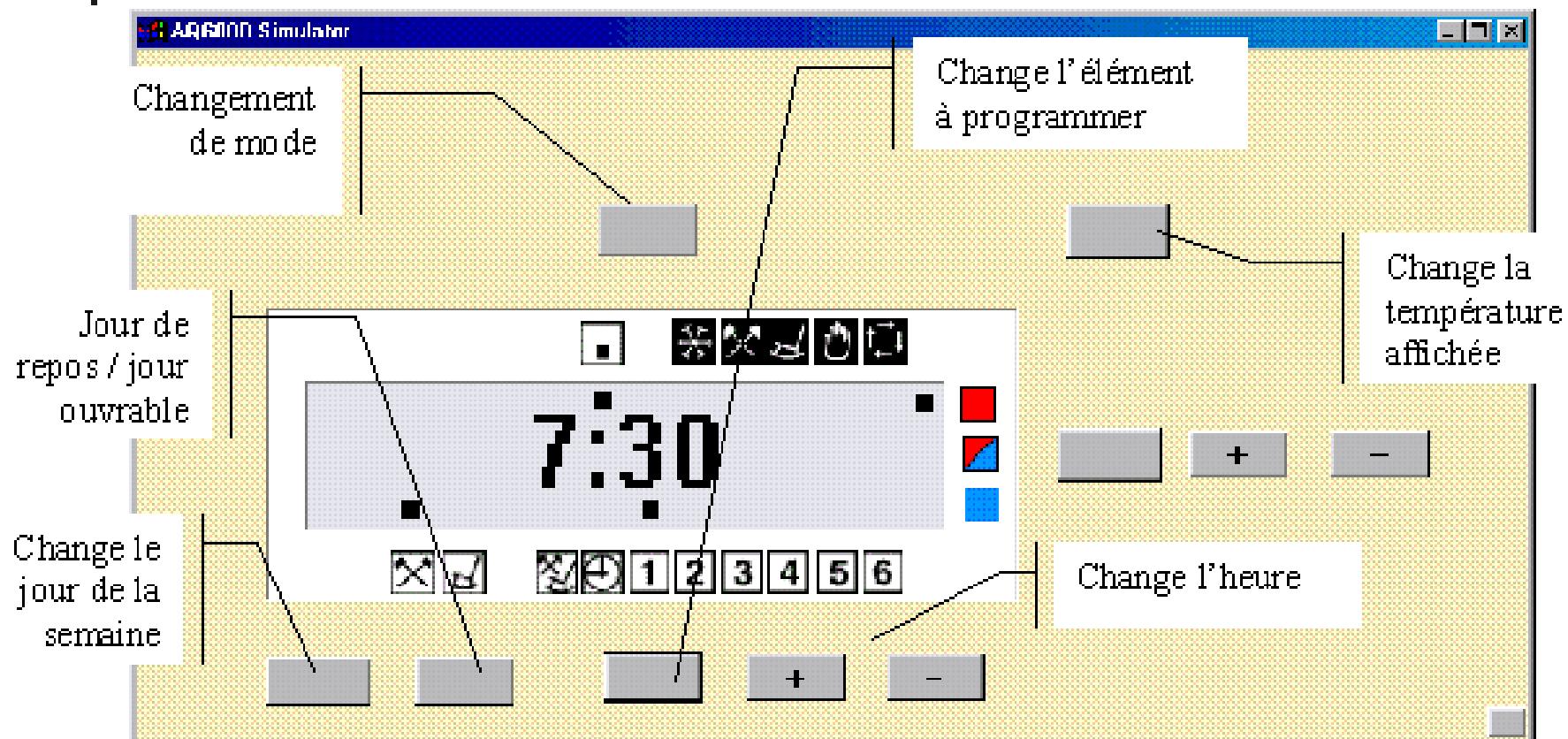
AQ6000

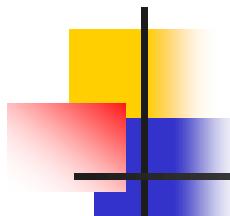


AQ6000



AQ6000





Migration de tâches

- Faire un modèle de tâches correspondant à l'activité de « laver la vaisselle »
 - 4 personnes
 - Repas normal
- Faire un modèle de tâche correspondant à l'activité de « je fais la vaisselle avec un lave vaisselle »

Améliorations?

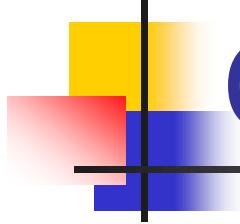
- <http://en.wikipedia.org/wiki/DishDrawer>



Interaction Homme-Machine

Un challenge en modélisation: changer de vitesse

- Wait until engine sounds ready for a change
- Depress the left-most pedal with your left foot while at the same time taking your right foot off the rightmost pedal
- Once the pedal is fully depressed work out what the new position of the gear lever has to be
- Move the gear lever to that new position
- Gently depress the rightmost pedal with your right foot while simultaneously releasing the leftmost pedal with your left foot until the leftmost pedal is fully released and the car engine sounds about right
- Be careful about temporal aspects (especially if driving up a mountain)
- Envisage errors and how to recover

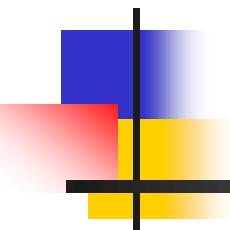


Other Models

- Rasmussen (SRK)
- Reason (Swiss cheese)
- Holnagel (Barriers and resilience)
- ...

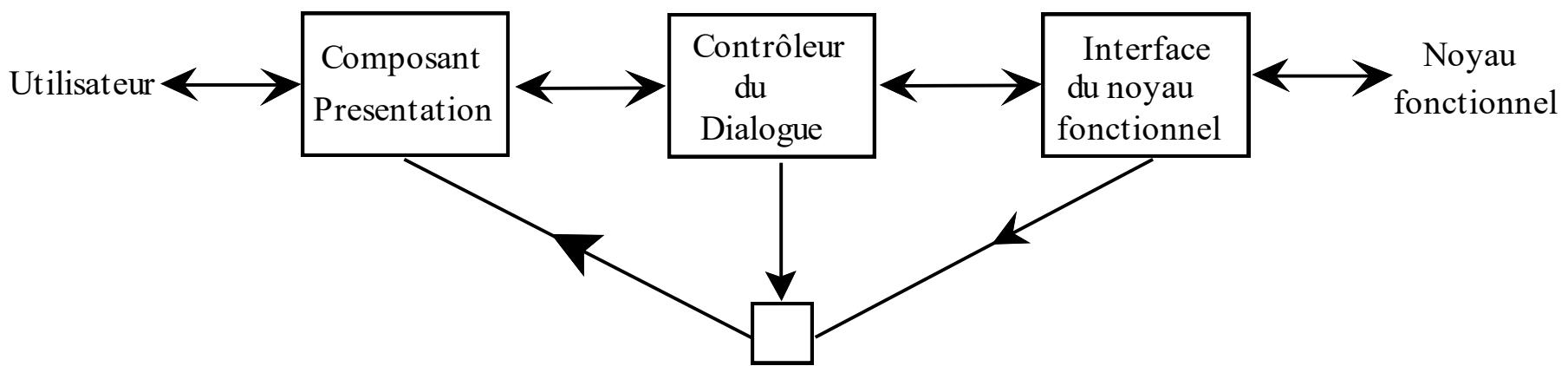


SPÉCIFICATION DES IHM

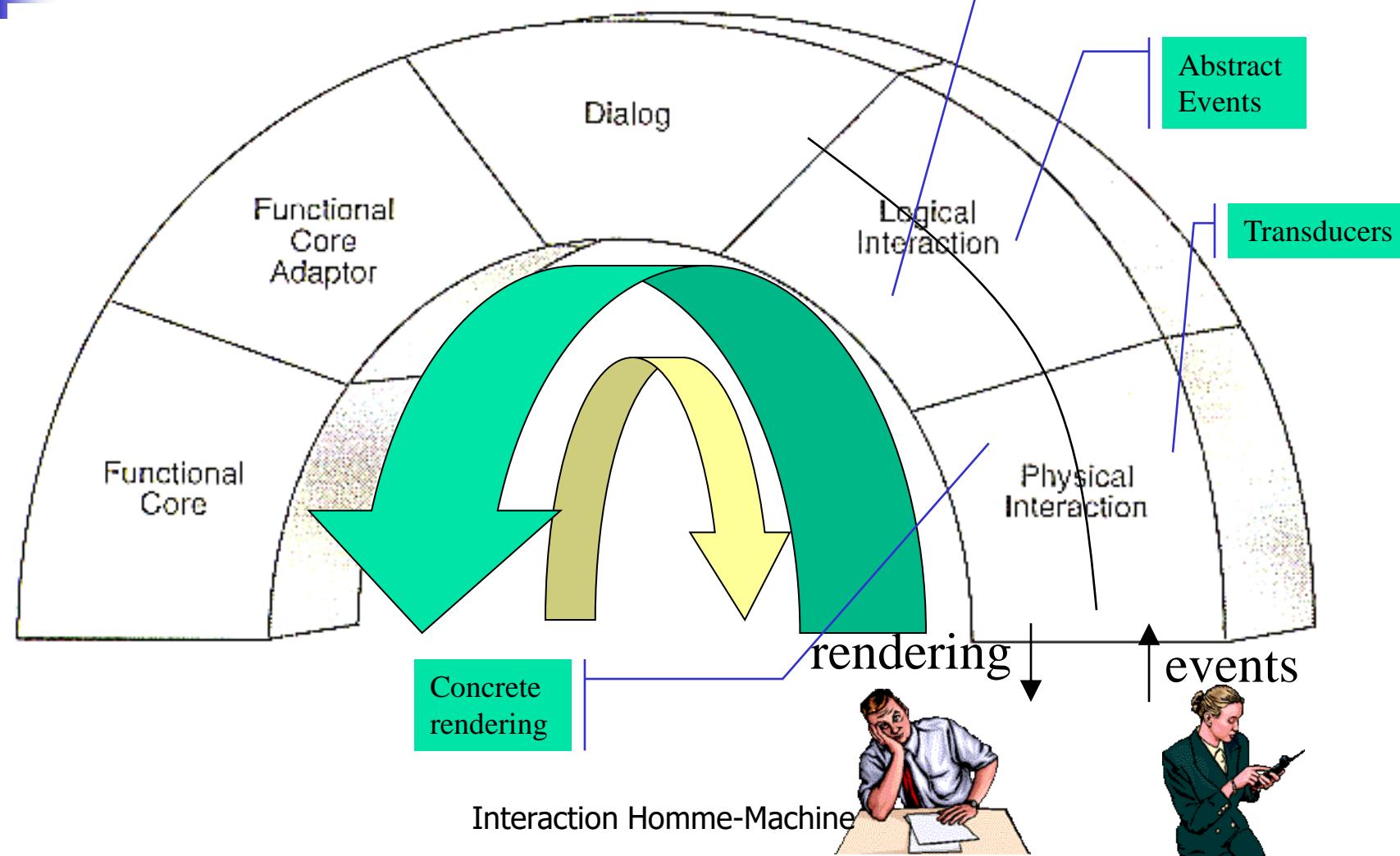


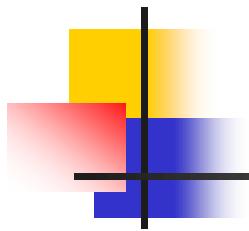
Du Design à la construction

Modèle de Seeheim 83



Modèle Architectural





Conception des IHM

Il faut concevoir les trois parties du modèle de Seeheim:

La présentation : ce que l'utilisateur voit de l'application

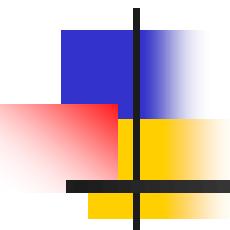
Le dialogue :

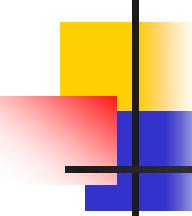
- qu'est-ce que l'utilisateur a la possibilité de faire
- comment l'utilisateur agit sur la présentation
- l'influence de son action sur ce qu'il pourra faire ensuite

Le noyau fonctionnel :

- les fonctions réalisées par l'application
- les données manipulées par l'application

Fonctionnement des systèmes par événements





Structure d'une appli. classique

A tout instant l'application est en attente d'une entrée de l'utilisateur.

Dans les autres cas elle fait des calculs et l'utilisateur doit attendre

Début
choix = '1';

Tantque choix <> '9' faire

affiche-menu;

lire(choix);

case choix of

1 : ajouter;

2 : modifier;

3 : supprimer;

9 : Quitter;

Fin Case

Fint Tantque

Fin

Procédure Ajouter;
début

rep = 'o';

Tantque rep <> 'n';

dessin-écran;

lire(nom);

lire(prenom);

...

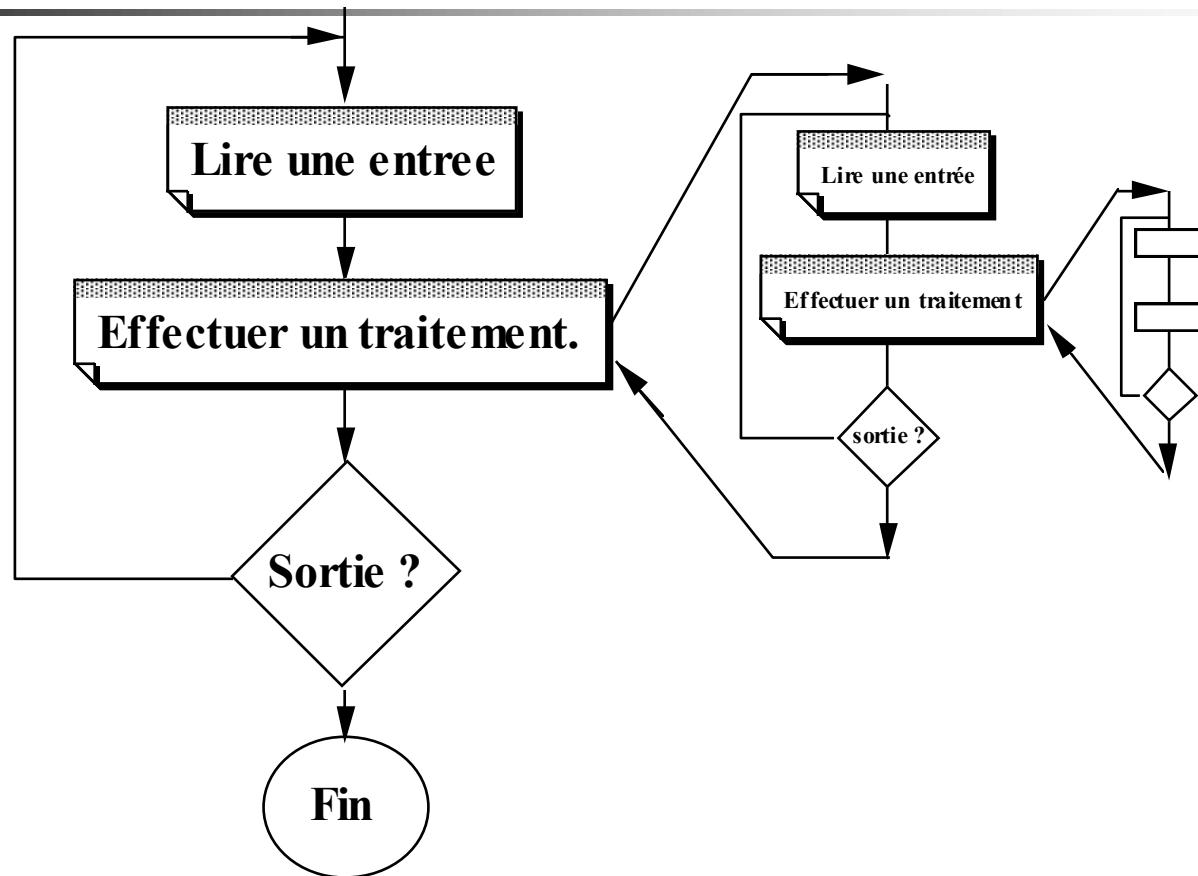
écrire('voulez-vous
continuer ?');

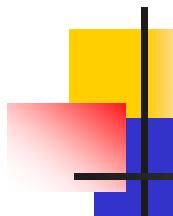
lire(rep)

Fint Tantque

Fint

Fonctionnement classique





Structure appli. par evt.

- La boucle d'événement (main event loop) : reçoit chaque événement produit par l'utilisateur
- Les gestionnaires d'événements : sont des procédures associées à chaque couple (widget, action sur un widget) et appelées par la main event loop dès que une action a été réalisée.

Tous les event handlers ont la même structure :

EH1;

 Précondition;

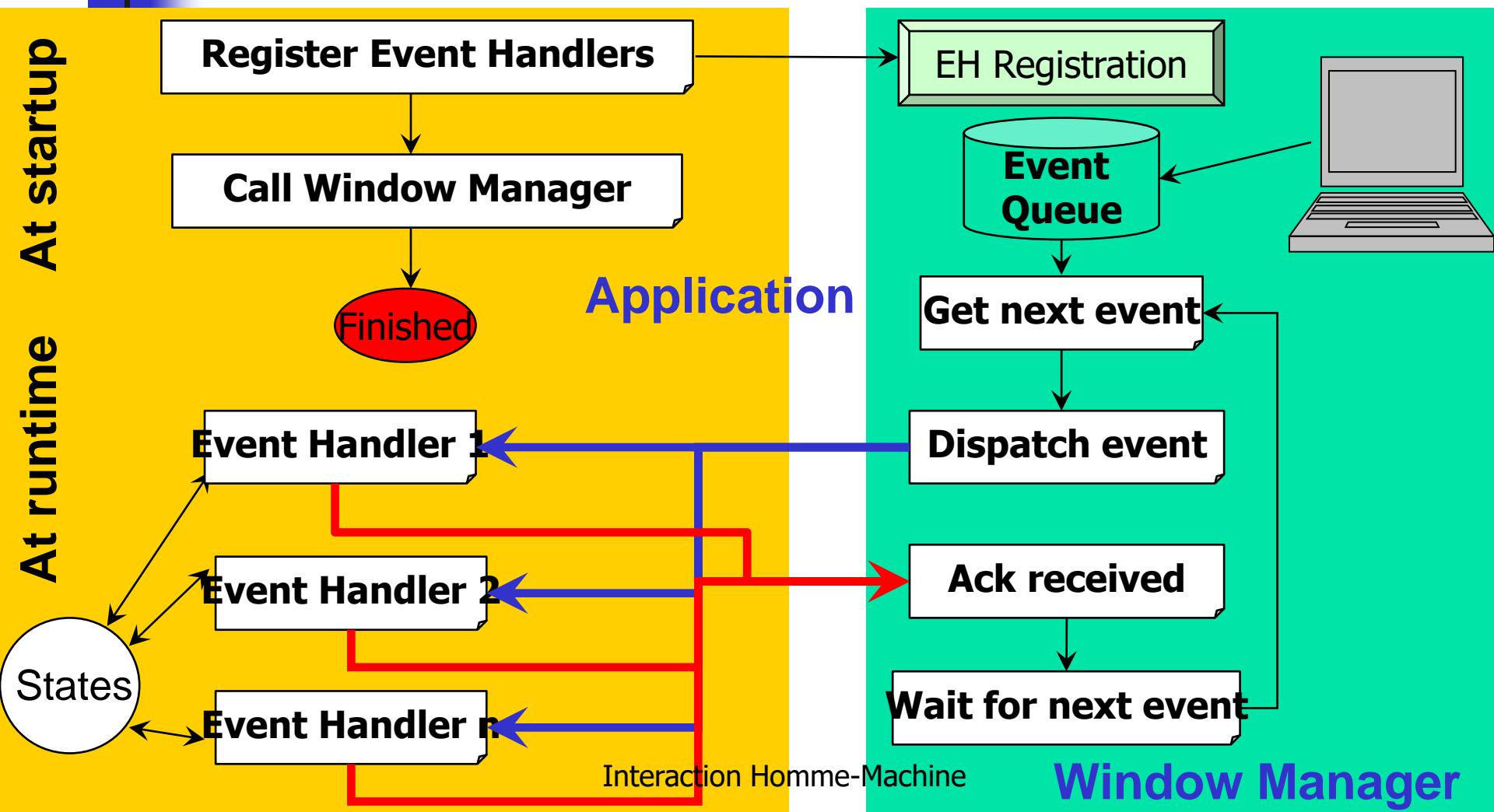
 Action;

 Modification de l'état du dialogue;

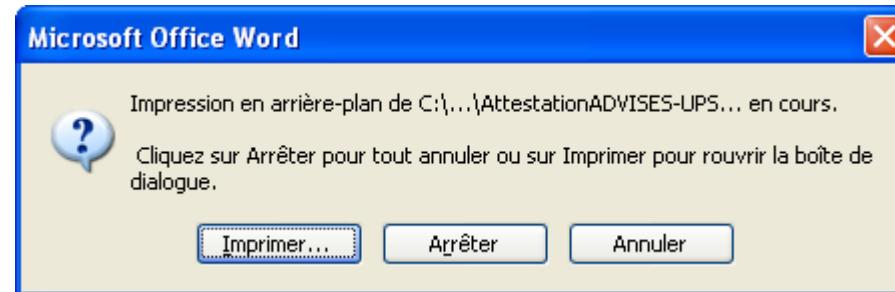
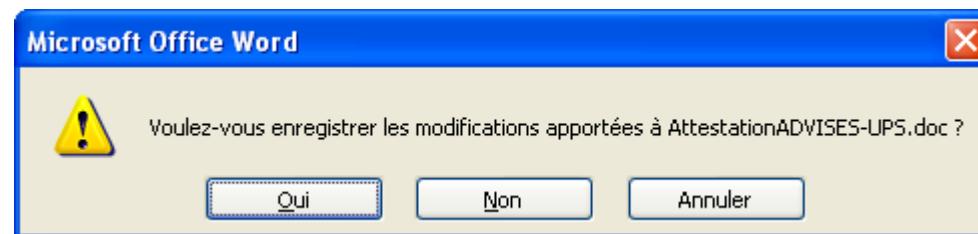
 Rétroaction graphique;

Fin EH1;

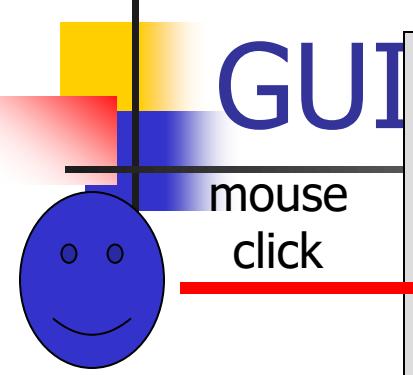
Event-based Functioning



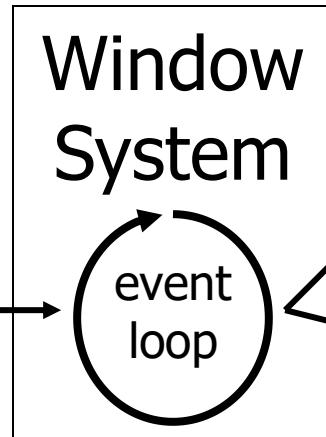
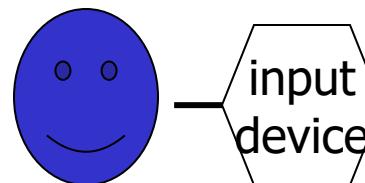
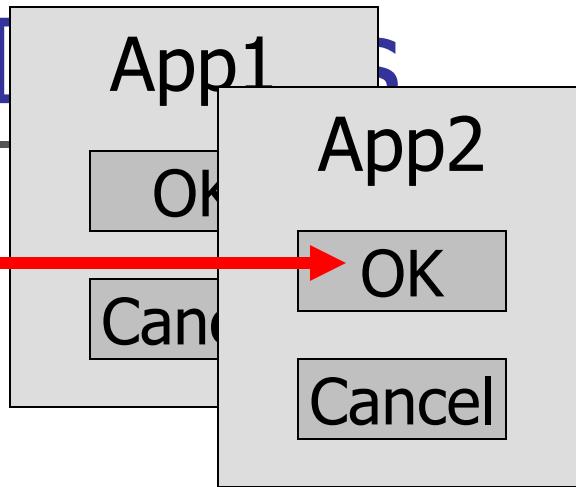
Comportement basé sur les états



GUI

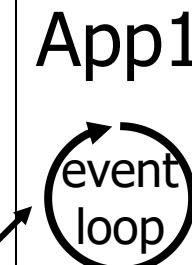


mouse click



Interaction Homme-Machine

which app?



which control?

App2 code:

```

OKbtn_click()
{
    do stuff;
}
CancelBtn_click()
{
    do different stuff;
}
App2Form_click()
{
    do other stuff;
}

```

Fonctionnement

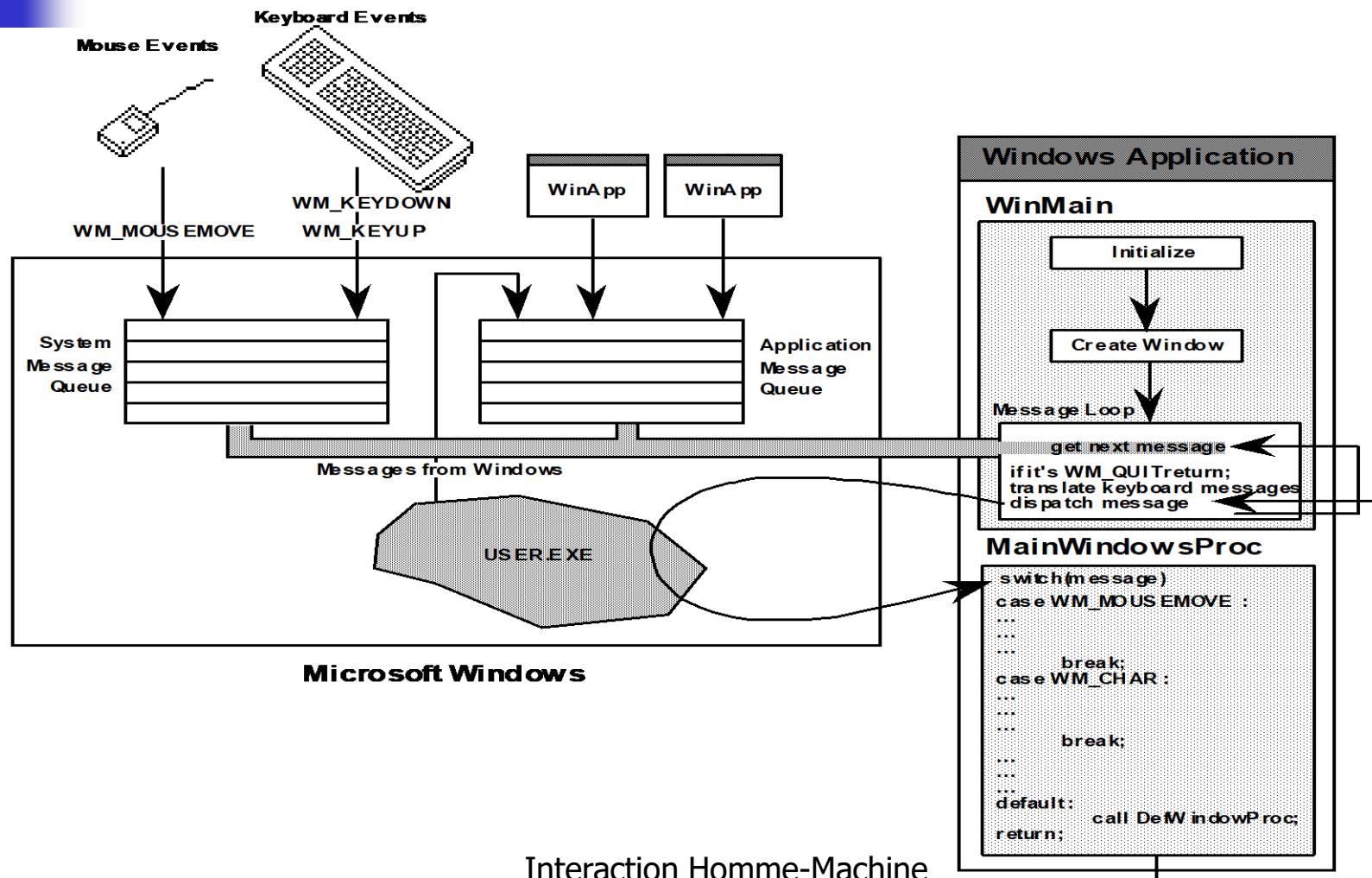
- “delegates” = callbacks
- Java: **Listeners**



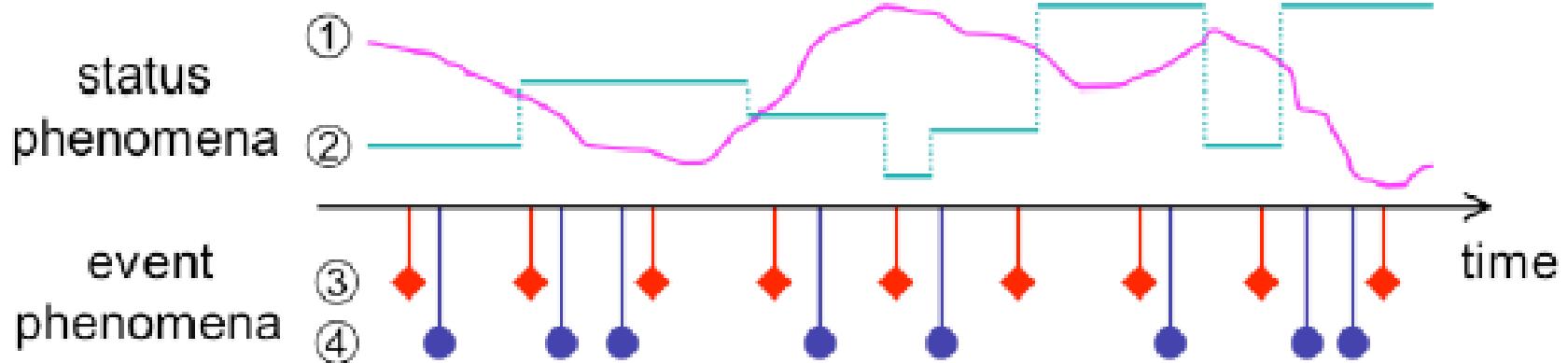
GUI App

```
Class{  
    decl data storage;  
  
    constructor(){  
        initialization code;  
        create GUI controls;  
        register callbacks;  
    }  
    main(){  
        Run(new )  
    }  
    callback1(){  
        do stuff;  
    }  
    callback2(){  
        do stuff;  
    }  
}
```

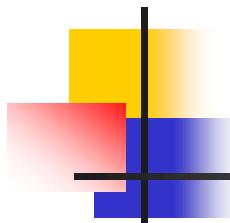
Evénements dans Windows



Etats et événements

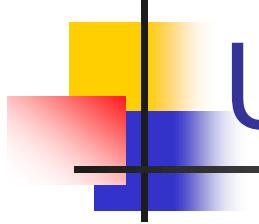


- 1- le monde réel évolue de façon continue
- 2- les variables représentent des variations par palier
- 3- les événements peuvent avoir une origine périodique
(regarder sa montre toutes les 30s)
- 4- les événements arrivent et ont un impact sur l'état



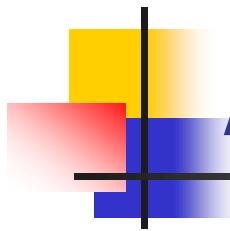
Une démarche de conception

- Une démarche de conception
- Une notation les automates
- Un processus proche de E/A (conception de bases de données)
- Un cheminement vers le code de l'application
- Pas de fossés à combler intellectuellement



Une démarche de conception

- 1) Analyse
 - a) conception de l'interface (design, choix des objets, ...)
 - b) liste des événements
 - c) liste des actions
- 3) automate de comportement
- 4) Matrice états/événements
- 5) Event-handlers

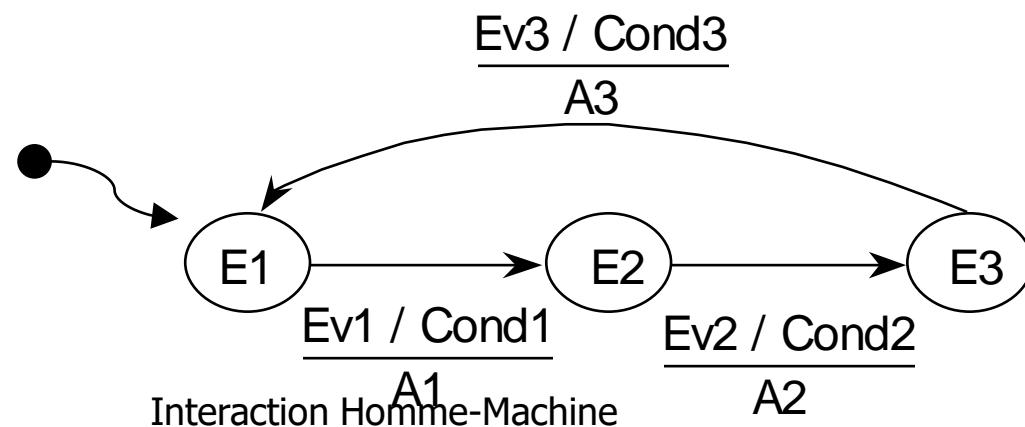


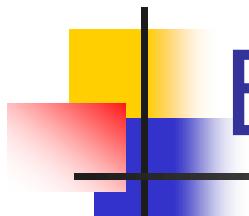
Avantages

- Description complète et non ambiguë
- Analyse de propriétés
 - Comportementales
 - D'utilisabilité
- Génération de code
- Il est plus facile de prouver que de tester

Les Automates Etendus

- Un automate étendu est un automate à états pouvant posséder :
 - des événements déclenchant des actions
 - des conditions de déclenchement des actions
 - des registres (variables numériques)
 - effectuer des actions sur les registres (affectation)
- Les événements, les conditions et les actions sont représentés
- sur les arcs sous la forme suivante :





Exemple par événement

```
Handler Ev1() {  
    switch(v) {  
        case 1 :  
            A1;  
            v:=2; ev1 actif ev2 actif  
        case 2 :  
            A2;  
            v:=3; ev1 actif ev2 actif  
        case 3 :  
            A2;  
            v:=4; ev1 inactif ev2 actif  
        case 4 :  
            'Interdit  
    }  
}
```

```
Handler Ev2() {  
    switch(v) {  
        case 1 :  
            'Interdit  
        case 2 :  
            A4;  
            v:=1; ev1 actif ev2 inactif  
        case 3 :  
            A3;  
            v:=2; ev1 actif ev2 actif  
        case 4 :  
            A3;  
            v:=3; ev1 actif ev2 actif  
    }  
}
```

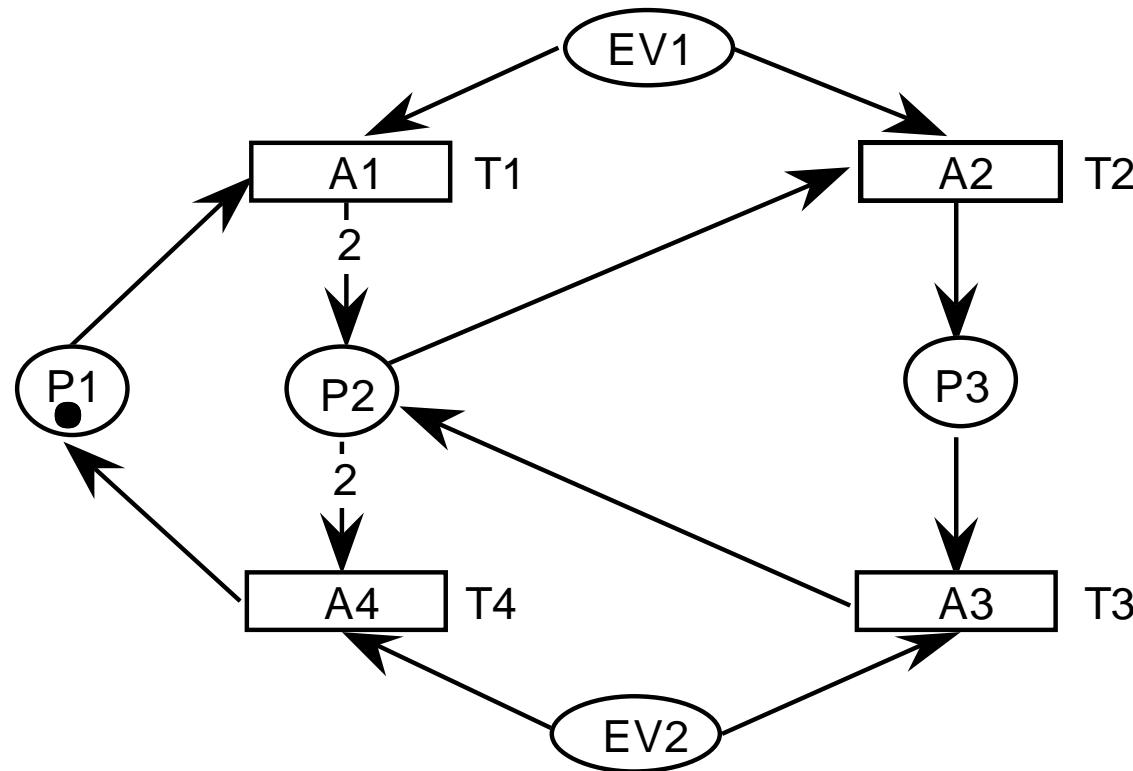
$$V = \{v / v_0 = 1 \text{ and } v : \text{integer}\}$$



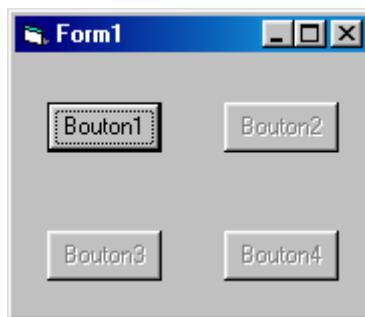
Tableau récapitulatif

Nature de l'application	Transformationnelle	Inter(ré)active
Protocole de communication (Application / Utilisateur)	Application = Client Utilisateur = Serveur	Application = Serveur Utilisateur = Client
Contrôle	Impératif	Déclaratif
État du dialogue	Historique	Valeur des variables d'état

Des Réseaux de Petri ???



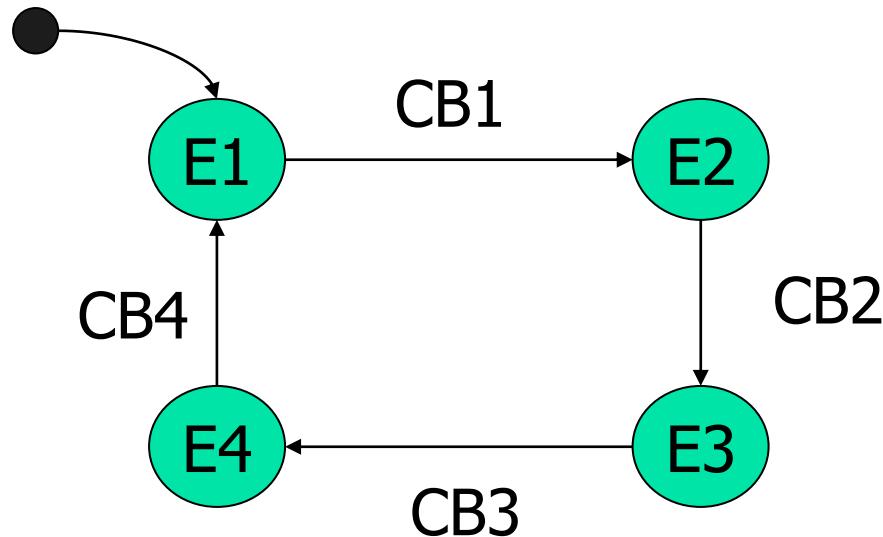
Exemple: les 4 boutons



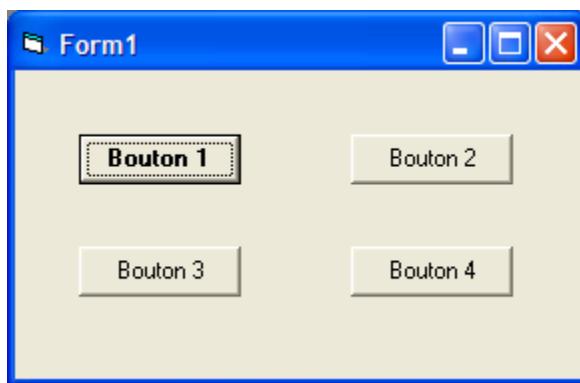
- Spécification du comportement d'une application avec 4 boutons cycliques

Automate Exercice 1

- 4 événements CB1, CB2, CB3, CB4



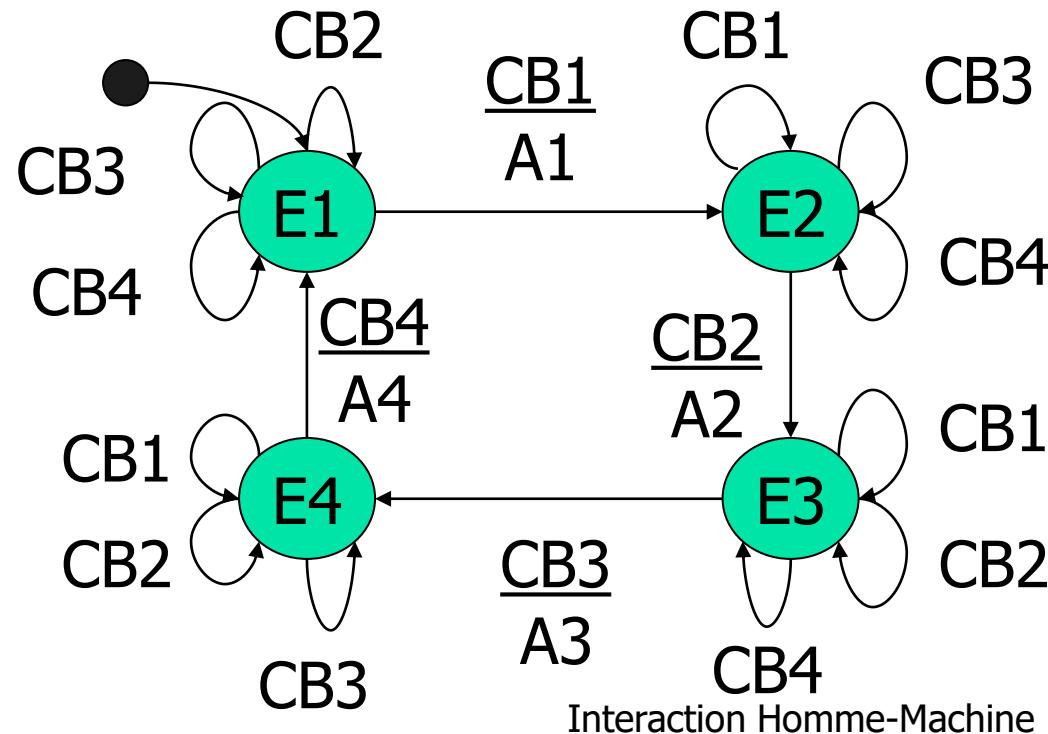
Exemple: les 4 boutons cycliques



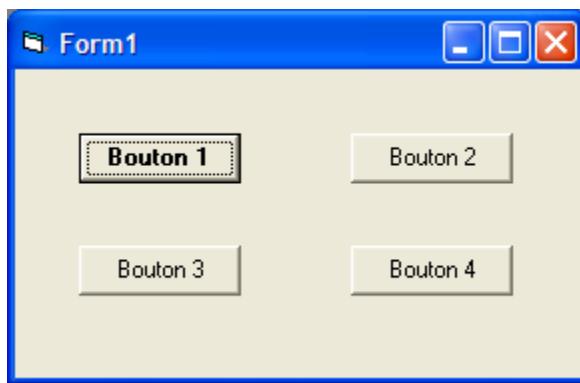
- Spécification du comportement d'une application avec 4 boutons cycliques toujours actifs

Automate Exercice 1 (2/3)

- 4 événements CB1, CB2, CB3, CB4
- 4 actions (chgt apparence boutons)



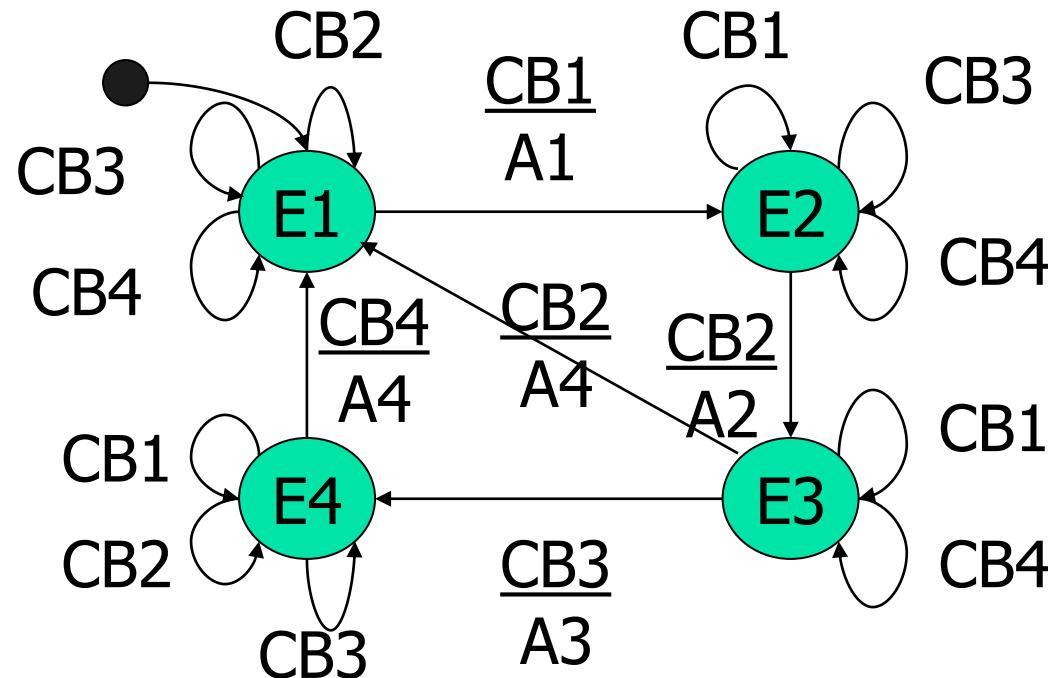
Exemple: les 4 boutons cycliques (3/3)



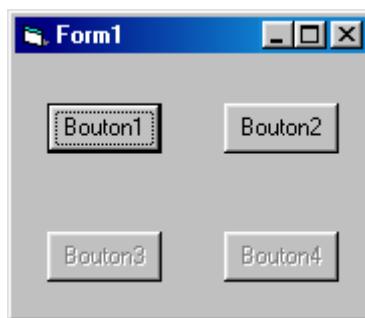
- Spécification du comportement d'une application avec 4 boutons cycliques toujours actifs et en ajoutant un raccourci (dans l'état où B3 est en gras, on peut cliquer sur B2 ce qui conduit dans l'état initial)

Automate Exercice 1 (3/3)

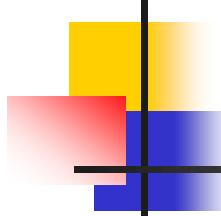
- 4 événements CB1, CB2, CB3, CB4
- 4 actions (chgt apparence boutons)



Exemple: les 4 boutons



- Spécification du comportement d'une application avec 4 boutons alternatifs



Autres exemples

- Compteur
- Feux tricolores
- Rubber banding
- Outils de dessin