

Interaction Humain-Machine

Bureau d'Etude n°3 :

Projet
Abribus+

Sujet

Projet Abribus+

Comment rendre la ville plus communicante grâce aux abribus ?



Deux sortes d'Abribus



Abribus de Russie sovietique



MIT Researchers Unveil EyeStop

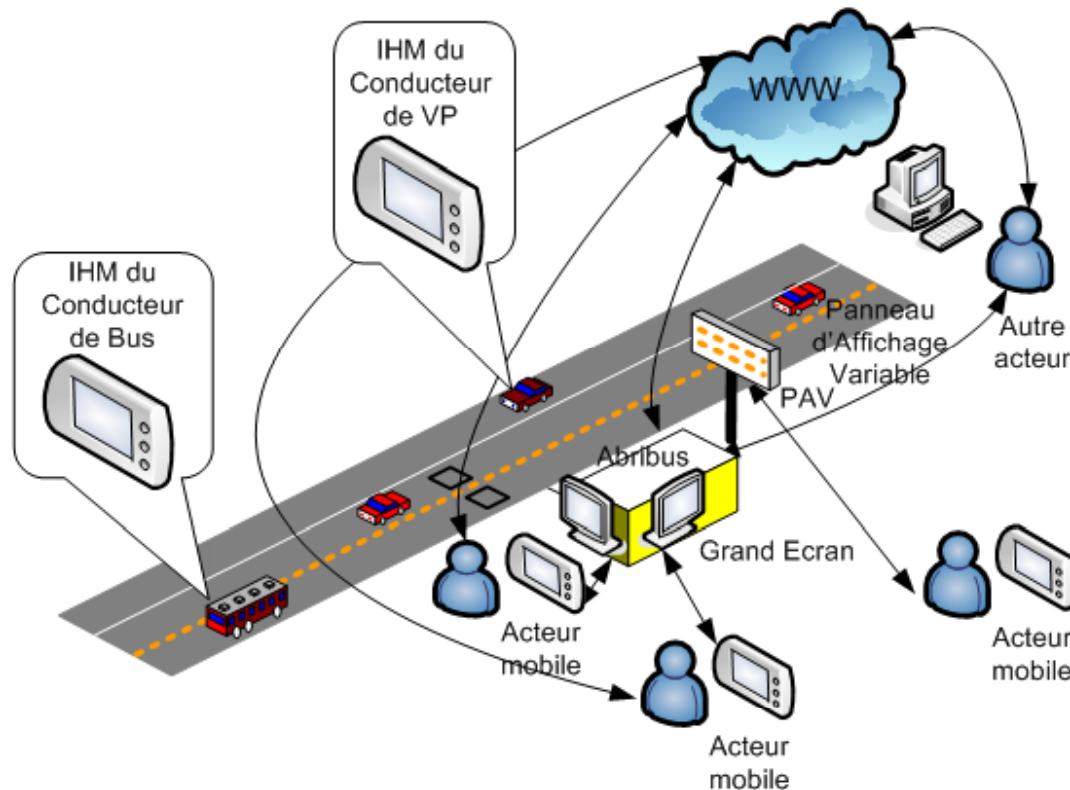
Internet aujourd'hui :

- Sa première version était fondamentalement caractérisée par les utilisateurs statiques (sédentaires) et l'information totalement virtuelle, pouvant se trouver n'importe où.
- La seconde version est actuellement en plein déploiement. Elle se caractérise par les utilisateurs mobiles utilisant des dispositifs légers (PDA, TabletPC, Téléphone mobile) mais avec l'information toujours virtuelle pouvant se trouver m'importe où.
- La troisième génération qui émerge actuellement concerne toujours des utilisateurs mobiles, mais accédant à l'information qui est plus localisée ou au moins dont l'accès est localisé. De cette façon l'information fournie est plus utile, car contextualisée et géo-localisée : LBS-MI (Location Based Services for Mobile Internet).

Abribus+

- Le présent projet a pour but d'explorer cette approche dans le cadre d'un contexte précis, celui d'un abribus communicant.
- Ce que nous souhaitons mettre en œuvre c'est l'utilisation de l'abribus comme support de communication poussée en matière de transport en commun.
- La gestion du co-voiturage (autostop) permettant aux offreurs de places et aux demandeurs de déplacement de se « rencontrer » via un grand panneau d'affichage situé sur l'abribus et supportant l'accès à internet de ces différents usagers constitue une seconde perspective.
- Une autre déclinaison souhaitée concerne le rôle de l'abribus comme point d'accumulation d'informations locales liées aux différents aspects de la vie du quartier (informations culturelles, commerciales, sportives, communautés locales, ...)

Abribus Communicant support d'échanges entre acteurs

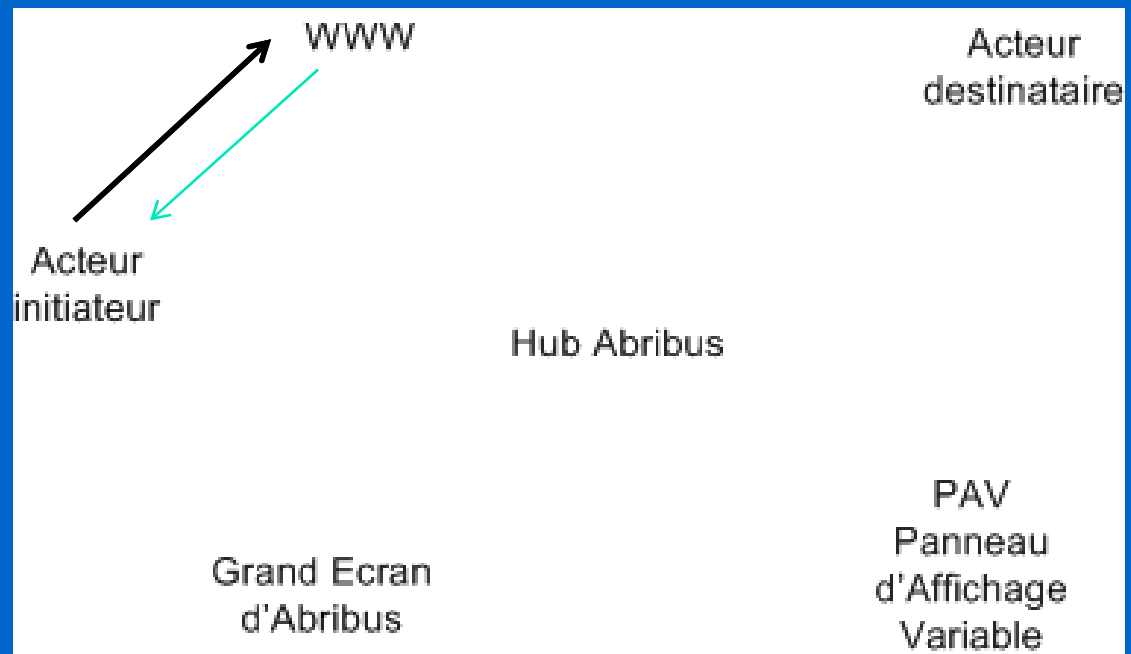


Principaux services :

- Fournir des informations précises et détaillées sur la situation instantanée des transports
- Possibilité de prise en compte de demandes spécifiques (transport de vélo, de poussette, de fauteuil roulant, ou de personnes à déficience visuelle, mobilité réduite, ...).
- Mise ne place dynamique de transport rapide ou lent en relation avec la destination de la majorité des passagers.
- Utilisation de l'abribus comme support du co-voiturage ou d'auto-stop en s'appuyant sur son panneau d'affichage (PAV – Panneau d'Affichage Variable).

Comportement n°1

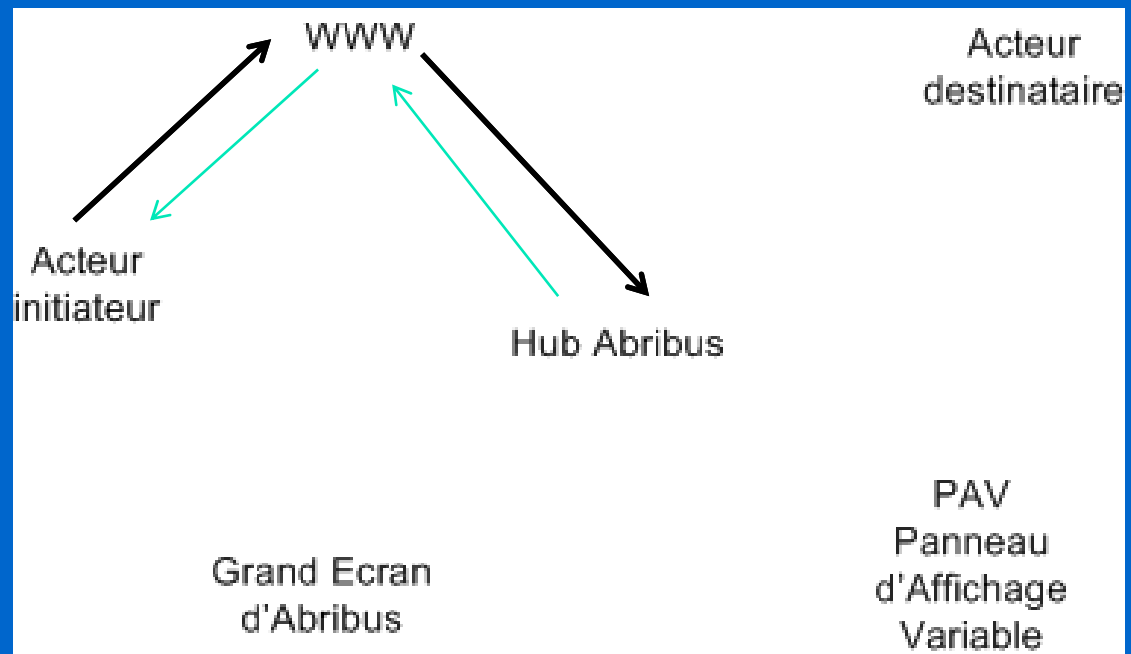
■ Interaction globale avec Internet



Exemple : Consultation d'Internet classique

Comportement n°2

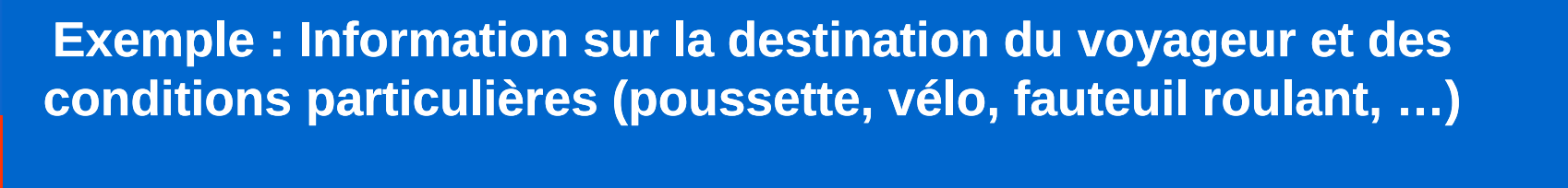
■ Interaction locale avec le Hub d'Abribus



Exemple : Consultation d'informations localisées: Consultation des horaires du Transport en Commun, des spectacles, matches, ...

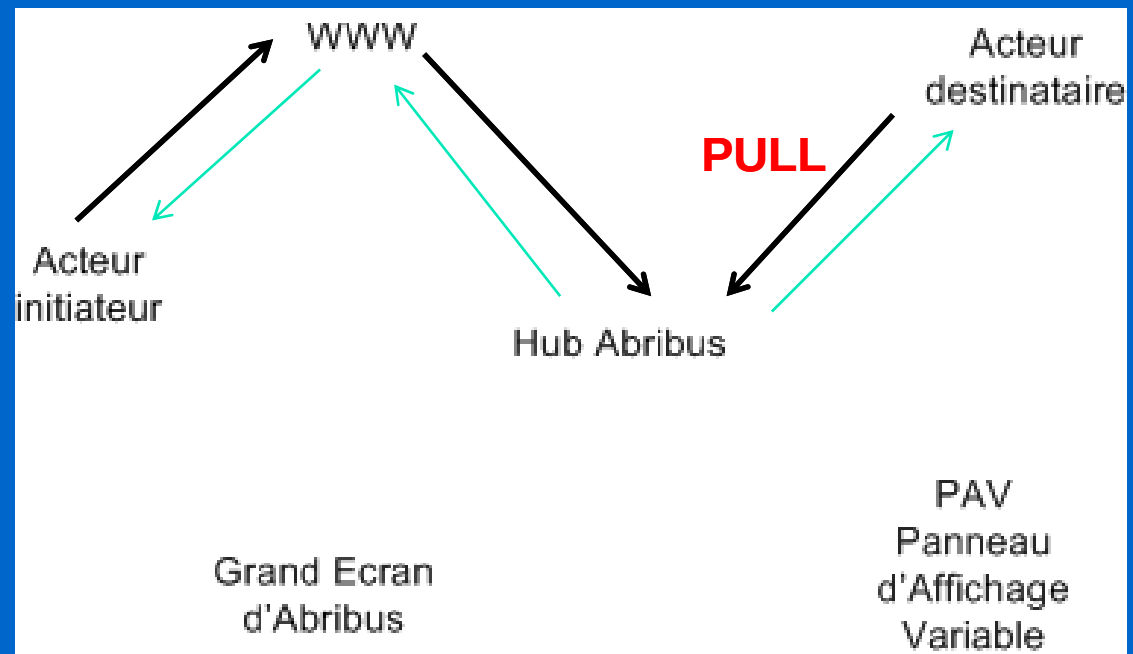
Comportement n°3

- Interaction locale avec le Hub d'Abribus et propagation vers l'acteur destinataire
-
- The diagram shows the following components and interactions:
- WWW** (top center)
 - Acteur initiateur** (bottom left)
 - Hub Abribus** (bottom center)
 - Acteur destinataire** (bottom right)
 - PAV** (bottom right, below Acteur destinataire)
 - Grand Ecran d'Abribus** (bottom left, below Acteur initiateur)
- Interactions (indicated by arrows):
- A black arrow points from **Acteur initiateur** to **WWW**.
 - A red arrow points from **WWW** to **Hub Abribus**.
 - A red arrow points from **Hub Abribus** to **Acteur destinataire**.
 - A red arrow points from **Acteur destinataire** to **PAV**.
 - A red arrow points from **PAV** to **Grand Ecran d'Abribus**.
 - A red arrow points from **Grand Ecran d'Abribus** to **Acteur initiateur**.



Comportement n°4

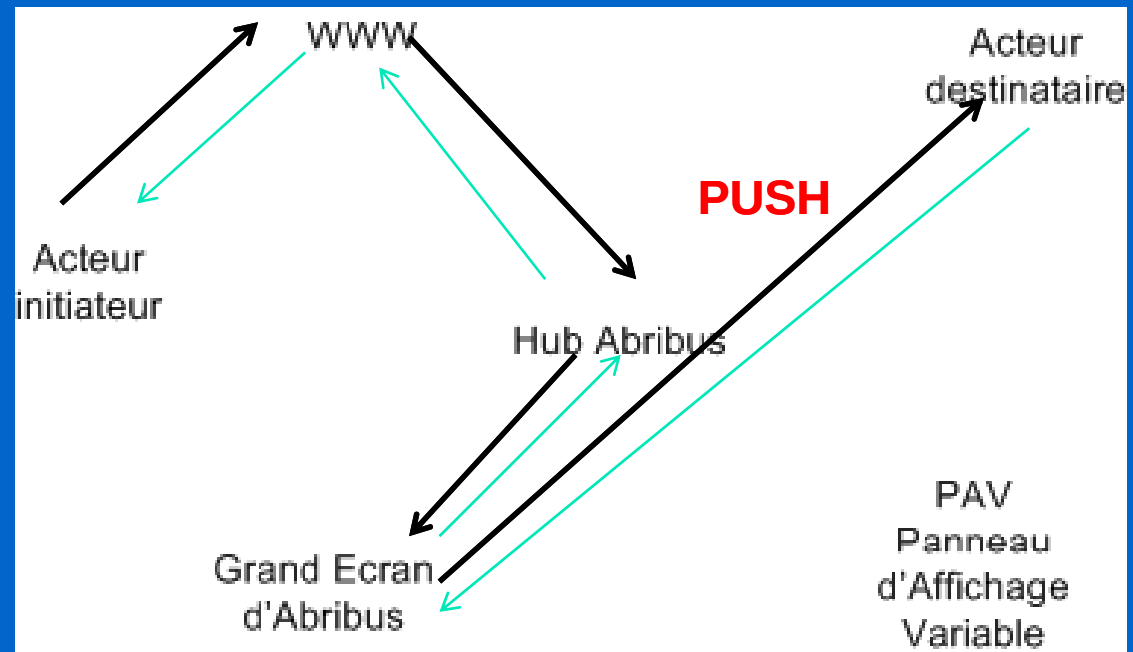
- Interaction locale avec le Hub d'Abribus et accès explicite par l'acteur destinataire



Exemple : Inscription au spectacle, communication de la destination, ... utilisables par l'acteur destinataire

Comportement n°5

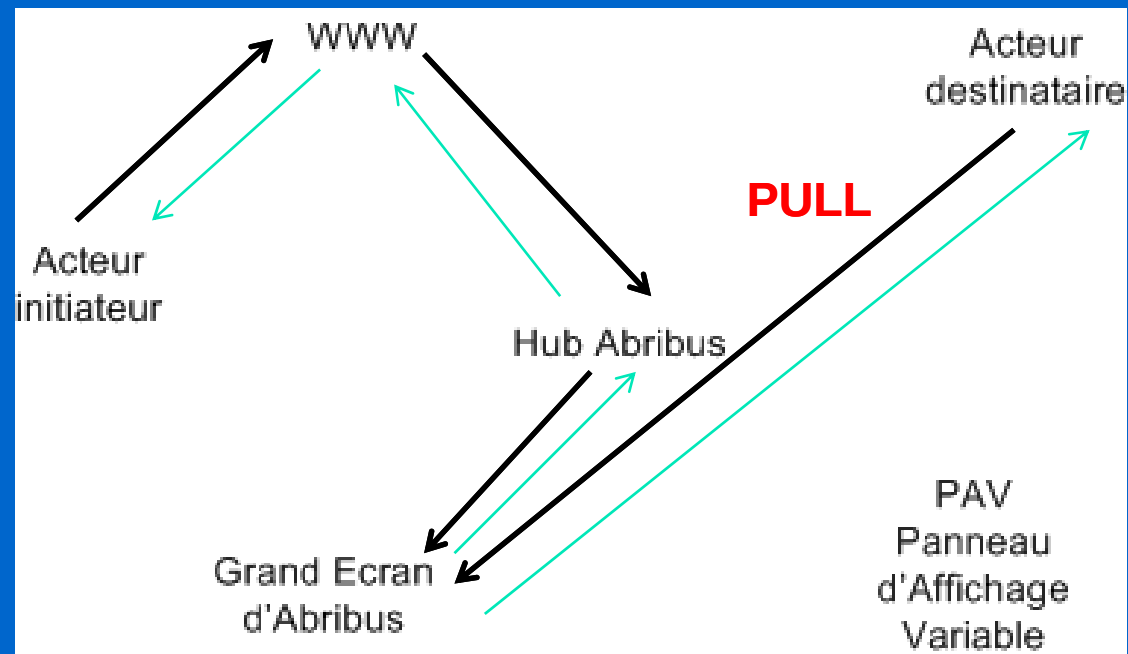
- Interaction locale avec le Grand Ecran de l'Abribus et propagation possible vers l'acteur destinataire



Exemple : Dans la boulangerie voisine on vent les dernières tartelettes à moitié prix

Comportement n°6

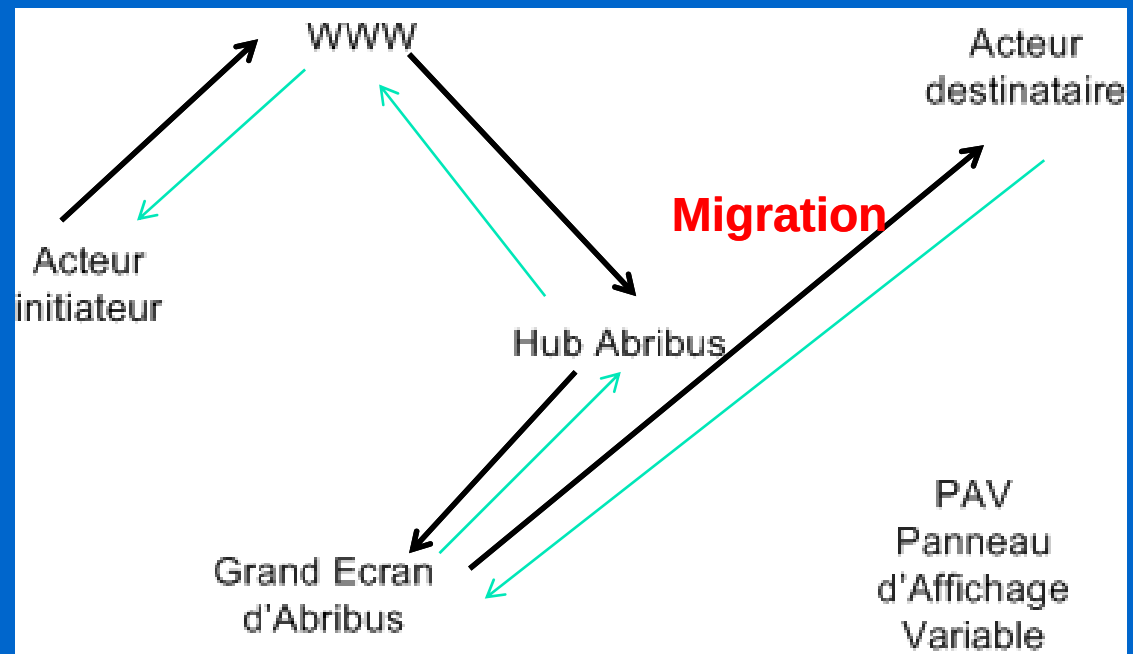
- Publicité locale envoyée au Grand Ecran de l'Abribus avec accès explicite possible par l'acteur destinataire



Exemple : Programme du cinéma du quartier

Comportement n°7

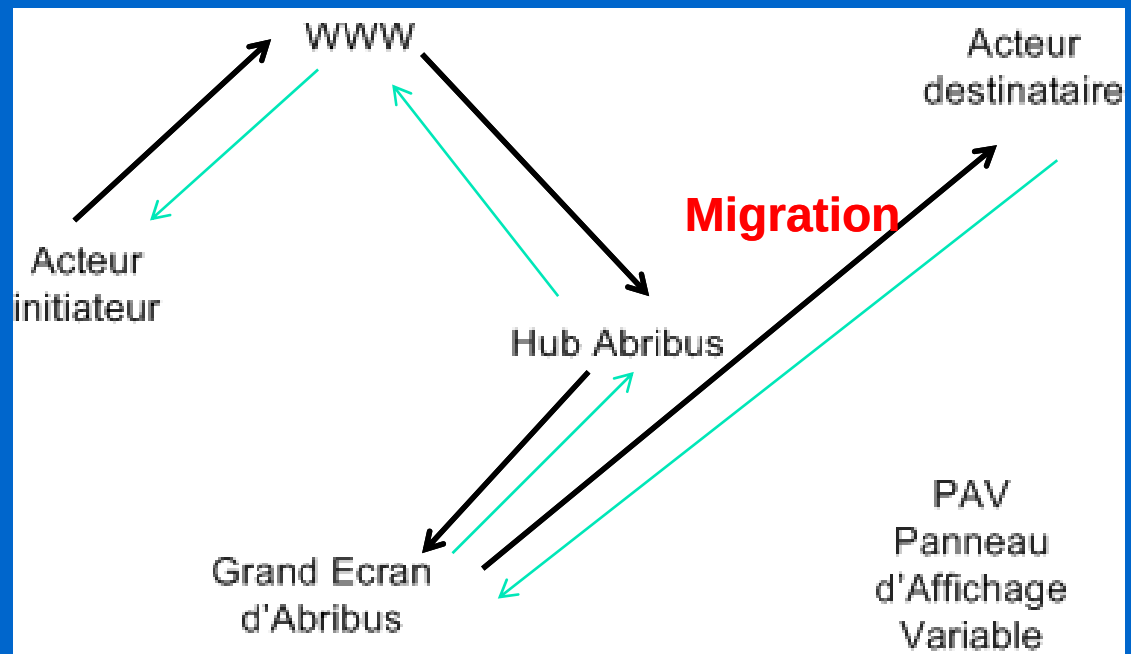
- Information semi-publique envoyée au Grand Ecran de l'Abribus avec collecte explicite possible par l'acteur destinataire



Collecte : migration sur le terminal
Exemple : Communication dont le contenu se dévoile sur le terminal du destinataire

Comportement n°8

- Information privée envoyée au Grand Ecran de l'Abribus avec collecte explicite par l'acteur destinataire (besoin d'un code d'accès)

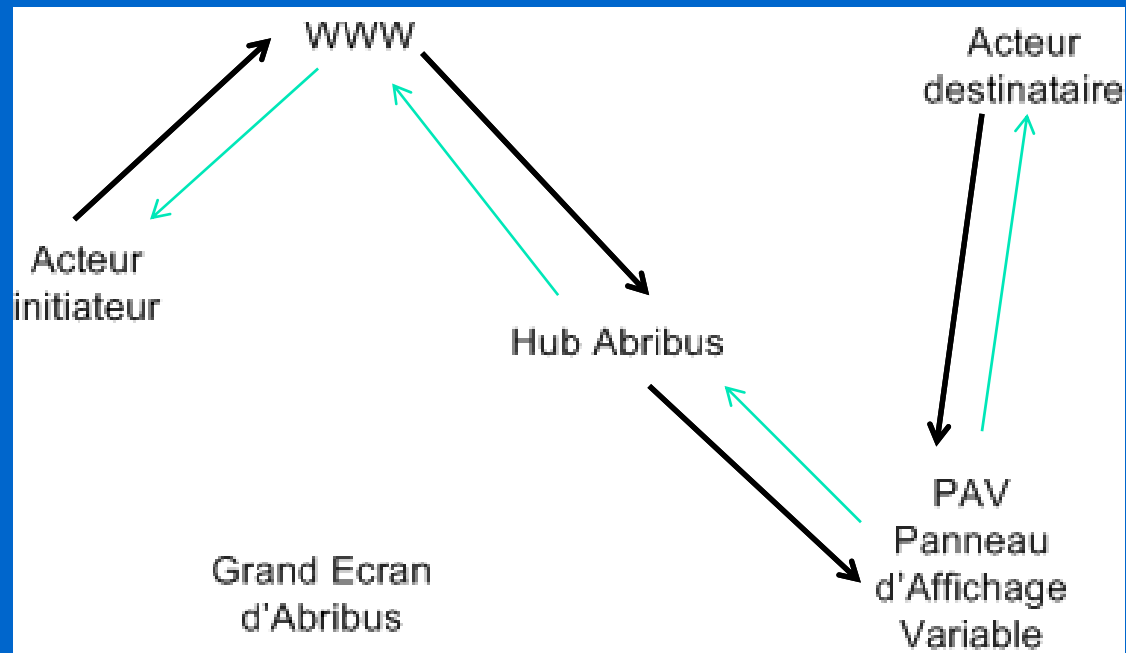


Collecte : migration sur le terminal

**Exemple : Communication secrète
et personnelle via un écran commun**

Comportement n°9

- Information envoyée au PAV (Panneau d’Affichage Variable de l’Abribus avec réaction d’un destinataire potentiel



Exemple : Gestion du co-voiturage

Démarche à appliquer

- Prendre en charge de la problématique « Abribus+ »
- Modéliser des tâches et des acteurs identifiés
- Choisir les dispositifs d'interaction de l'ordinateur porté
- Dérouter la démarche méthodologique de conception d'IHM
- Etudier des patterns à réutiliser
- Proposer des IHM appropriées par rapport aux tâches à mener et les dispositifs choisis pour les acteurs identifiés
- Evaluer des choix par la méthode QOC

Les concepts de MOCOCO :

- La **mobilité** propose d'utiliser les dispositifs portables (ou portés), miniaturisés, offrant des services appropriés au contexte et atteignables dans le contexte de communication globale.
- La **collaboration** (locale ou à distance) amène la possibilité de faire intervenir des acteurs multiples à compétences variées et non nécessairement co-localisés pour gérer la situation posée.
- La **contextualisation** propose de rendre disponible sur le lieu de l'action et plus généralement à tous les acteurs, les informations contextualisées.

Dispositif léger d'accès au système

Connectable - déconnectable -
reconnectable

Accompagnant l'utilisateur dans ses
mouvements

PDA, "*wearables computers* »,
"*handheld computers*".



- **Mobilité des acteurs :**
 - ➔ Mobilité des acteurs à une échelle de la ville ou de l'agglomération
 - ➔ Atteignabilité des acteurs
 - ➔ Atteignabilité par ceux-ci du système d'information commun
 - ➔ Communication entre les acteurs, le système d'information et les Abribus
- Dispositifs variés (miniaturisation)
- Connexion - Déconnexion
- Prise en compte de la localisation
- Support de distribution et mobilité

Prise en compte de la localisation

➤ La notion d'environnement attentif :

- ➔ Des capteurs observent l'environnement et actualisent l'état perçu.
- ➔ Le contexte est pris en compte :
 - qui : identification des objets et utilisateurs,
 - où : localisation physique des objets et utilisateurs,
 - quand : historique d'interactions.

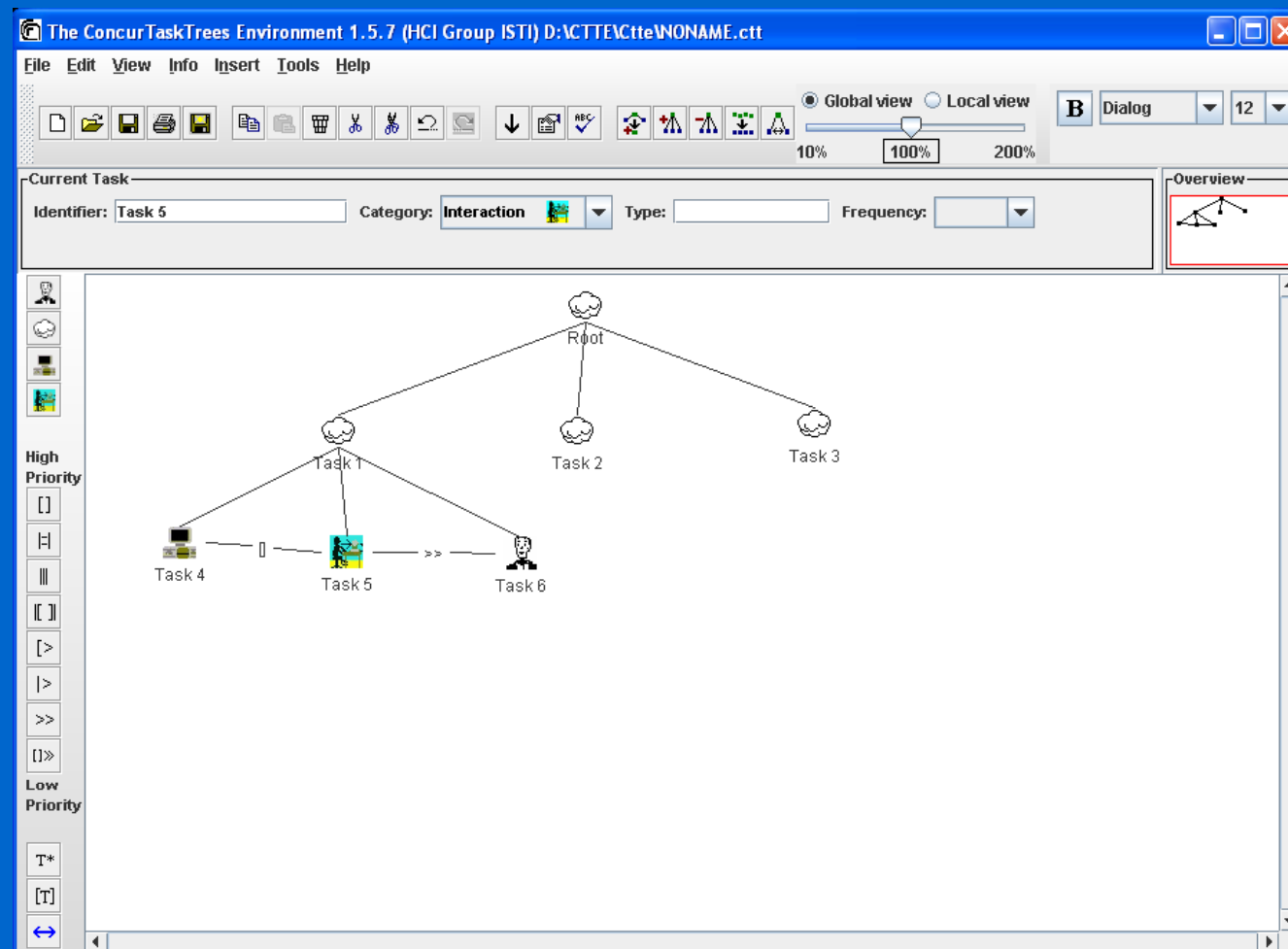
Travail à faire

Le but de ce BE est :

- ➔ d'identifier les acteurs du projet Abribus+ et s'occuper d'eux,
- ➔ d'élaborer les arbres de tâches pour les différents types acteurs
- ➔ de choisir les équipements portés appropriés (si nécessaire)
- ➔ d'élaborer, en suivant la démarche méthodologique de conception d'IHM, les différentes IHM
- ➔ d'évaluer celles-ci avec la méthode QOC

CTTE : ConcurTaskTrees Environment

This tool is realised by the [Human Computer Interaction Group - CNUCE \(Pisa\)](#). With this editor you can build a task model and generate an interactor-based architectural model (this beta version does not support it yet). It is possible build Cooperative Tasks Models or Single User Task Model.



CTT Notation

- **Temporal Relationship Priority:**
- **Following it is reported the priority ordering of the temporal relationship listed from major to minor:**
- **unary operator**
- **$[]$ $|=$ $||$ $|||$ $|||$ $[>$ $|>$ $>>$ $[]>$**
- **Temporal Relationship Combining**
- **There is the possibility to combine the unary operators with the others, but in this case some expression can be result wrong.**
- **The tool offer the possibility to check those inconsistency. Following we present the possible wrong expression that it is possible build.**
- **The expression $A*>>B$ is wrong , B is never reachable**
- **Left and right side of the operator $|>$, $[>$ and $[]$ can not be optional.**

NAME	SYNTAX
Choice	$T1 [] T2$
Order Independency	$T1 = T2$
Interleaving	$T1 T2$
Synchronization	$T1 T2$
Sequential composition (enabling)	$T1 >> T2$
Sequential composition with information passing	$T1 []>> T2$
Disabling	$T1 [> T2$
Infinite Iteration (unary operator)	$T1^*$
Optional Execution (unary operator)	$[T1]$
Suspend/Resume	$T1 > T2$

1. How to Build a new single-user task model

- **The first screenshot**
When you start CTTE, the "Root" node of a new single-user task model is shown, and its category is set by default to "abstraction" (a cloud-shaped icon is used). At that moment the root is the only node in the task model: within the editor panel it is highlighted as the "*current*" task (see the black frame around its icon).
- **Focus on the "*current*" task**
Apart from very few exceptions, at any time while you use CTTE only one task will be highlighted as the "*current*" task, meaning that every action you do will be referred to this task. Changing the current task is easy: select the icon of the concerned task. Some information about the current task is shown in the main window to allow you quick access and editing.
- **Adding new tasks: the insertion mode**
A new task can be added according two possible *insertion modes* (the current insertion mode is represented by a two-state button included in the CTTE global toolbar):
 - as the *farthest right child* of the currently selected task. This insertion mode is selected by default and in this case the two-state button displays a downward arrow.
 - as the *left sibling* of the currently selected task. In this case the two-state button displays a leftward arrow.
- In order to change the insertion mode you can use the two-state button displayed within the toolbar, or alternatively use the menubar and select Insert |Below(child) or Insert |To the left(sibling) depending on the desired insertion mode.

1. How to Build a new single-user task model, cont.

- If you want to add a new task as the *farthest right child* of a task *T* do the following 1.-3. steps:
 - ➔ make sure that the appropriate insertion mode is selected within the tool (namely, the two-state button displays a *downward* arrow);
 - ➔ select task *T* (namely, click on its icon);
 - ➔ select the icon associated with the appropriate task category within the related toolbar
- If you want to add a new task as the *left sibling* of task *T*: make sure that the appropriate insertion mode is selected within the tool (namely, the two-state button displays a *leftward* arrow), then follow the previous 2.-3. steps.
- Add operator between tasks
You can add a temporal operator on the right side of the currently selected task. Thus, if you have two sibling tasks, *Task1* and *Task2* and you want to specify *Task1 op Task2* (where *op* is a CTT operator) you have to click on the icon associated with *Task1*, then select (within the operator toolbar) the button associated with the desired operator .
- Change properties of a task
To change the properties of a task, double-click on its icon. A new window will be displayed making you able to change the properties you are interested in. A subset of relevant properties about the *current* task is also displayed within the main window in order to give you quick access to them.

2. How to Build a new cooperative task model

- **Enable "Cooperative mode" option**
When you start CTTE, by default you are supposed to build a single-user task model. If you want to create a cooperative task model you have to enable the View|Cooperative mode option. As a result, the root of the Cooperative task model ("Cooperative root") will be displayed in the editor panel and the "Cooperative" label is displayed in the tabbed panel within the main window, meaning that the task of the "Cooperative" task model is currently shown.
- **Add new roles**
Before building the cooperative task model, you should specify the task model of the involved "roles", so you have to add new roles to the cooperative task model. In order to do it select the "New Role" button in the bottom-right hand corner of the window displayed when the tabbed panel selects the "Cooperative" task model. As a result, the tabbed panel is automatically switched to the brand-new role task model allowing you to edit the task model of a single-user role.
- **Focus on the "*connection tasks*"**
If you are building a cooperative task model, you are supposed to have specified -within the task model of each role- some "*connection tasks*". Such tasks are displayed -for each role- when the Cooperative task model is selected in the right-hand part of the main window and they are useful to build easily the cooperative task model.
- **Building the cooperative task model: set connection tasks**
All the leaves of the cooperative task model have to be connection tasks previously defined in the task model of the involved role. If you want to add a (connection) task in the cooperative task model you have to add the task in the cooperative task model as usual (that is by selecting the appropriate icon in the left-hand part of the window). Then, make this task the *current* task and select the appropriate role/task in the lists displayed in the right-hand part of the window, then click "Set Connection" button.

3. How to Edit an existing single-user/cooperative task model

- General commonly-used editing operations are available within the CTTE
 - a) Cut/copy either the current selected task (Edit|Cut Selection) or the whole task tree having the current task as its root (Edit|Cut Subtree); Paste the task currently saved in the clipboard (Edit | Paste)
 - b) Undo the last editing action performed (Edit | Undo)
 - c) Insert a task whose specification is stored in a separate file (Insert|Subtree from File)
 - d) Make a new task father of a number of tasks: with the "^Control" key pressed, select such tasks, then Insert | New Level
- Some other features are available in order to support users while editing their task models
 - a) Find a task (Edit | Find...)
 - b) Hide some parts of a task model (Edit | Hide Subtree), justify a task tree, line up different tasks at the same level,
 - c) Drag a task (select the task with the left button of the mouse and drag it) or a whole task tree (select the root of the concerned task tree with the right button of the mouse and drag)
- Obtain some information about the task models
 - a) Get general information about the number/type of tasks (Info| Task Model Statistics...)
 - b) Compare two task models (Info| Compare...) to get their differences.

4. How to Save a single-user/cooperative task model in different formats

- There are three main file formats in which you can save a task model (or a part of it):
 - ".cctt"/".ctt" formats
The extension of a CTT specification is ".ctt" for single-user task models and ".cctt" for cooperative task models. By default you save the entire task model, but you can also save only a part of it (by selecting the root of the concerned subtree: File | Save Subtree as)
 - ".jpg" format
The CTTE give the users the possibility to save their task models in ".jpg" format (File| Save Tree as JPG to save a whole tree, ile| Save Subree as JPG to save the subtree whose root is the *current* task). This capability is very handy to manage task model specifications.
 - ".xml" format
A CTT specification can be saved also in XML format to enhance the interchange with other commonly-used notations and formats (File | Save CTT as XML).

5. How to Check the single-user/cooperative task model structure

- In order to facilitate the users to build correct task model specifications, the CTTE offers the possibility to perform some automatic checks on the specification itself, regarding all the possible errors which might make the specification incomplete or ambiguous (i.e.: a task with only one child, lack of temporal operators between two children, a connection task which does not appear in the cooperative task model, ...).
- To enable such functionality select **Tools | Check Model Structure**. A window appears with the list of errors (if any) together with the possibility of having quick access to the task referred by a specific error (by double-click on the error itself)

6. How to Obtain a task model starting from an informal description of the system

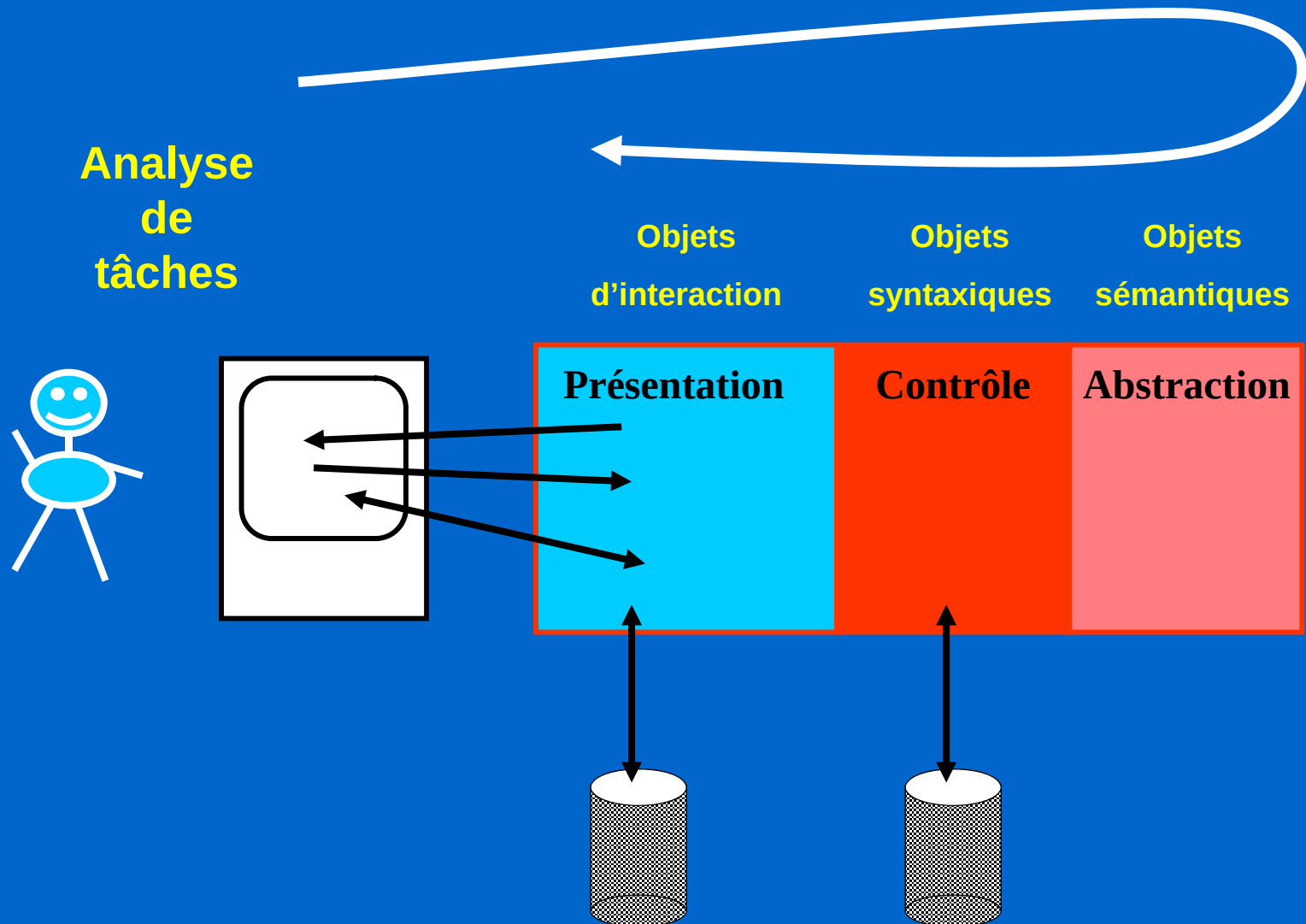
- Sometimes users might have available an informal description of a system, or some sample scenarios describing typical behaviour of the system itself.
- The CTT Environment offers the useful capability of supporting users while they extract relevant information from such scenarios in order to build a more formal specification (Tools | Informal to Formal Description).

7. How to Simulate the behaviour of the specification within the CTT Environment

- Especially with large specifications it is very useful to have the possibility to generate and follow some specific execution sequences or paths that are possible within a task model.
- Once you have specified the behaviour of a specific system, the Ctt Environment allow you to "simulate" some behaviour by means of the Task Model Simulator (Tools | Start Task Model Simulator).
- The main feature of the simulator is to enable from time to time only the tasks which are possible to execute according to the model specification and depending on the previously selected tasks.

- Résumé : Conception descendante avec les étapes suivantes :
 - ➔ Analyse des tâches
 - ➔ Synthèse conceptuelle
 - ➔ Conception sémantique
 - ➔ Conception syntaxique
 - ➔ Conception articulatoire (lexicale)
 - ➔ Conception de l'environnement utilisateur
 - ➔ Vérification de la conception
 - ➔ Mise en œuvre

Méthodologie pour la conception d'IHM



Patterns pour la Conception

- En 1970 C. Alexander a introduit des patterns pour l'architecture
 - ➔ Principe: "Chaque pattern est une règle à trois champs, qui expriment une relation entre un certain contexte, un problème et une solution"
 - Gamma et al. ont introduit les patterns pour le génie logiciel en 1995
 - Les patterns d'IHM pour la conception de systèmes interactifs existent depuis 1998
 - ➔ Coram and Lee, Tidwell, Welie, Borchers
- ➔ <http://www.welie.com/patterns/index.php>
 - ➔ <http://hillside.net/patterns/onlinepatterncatalog.htm>
 - ➔ <http://kettle.cs.berkeley.edu/ubicomp/2>

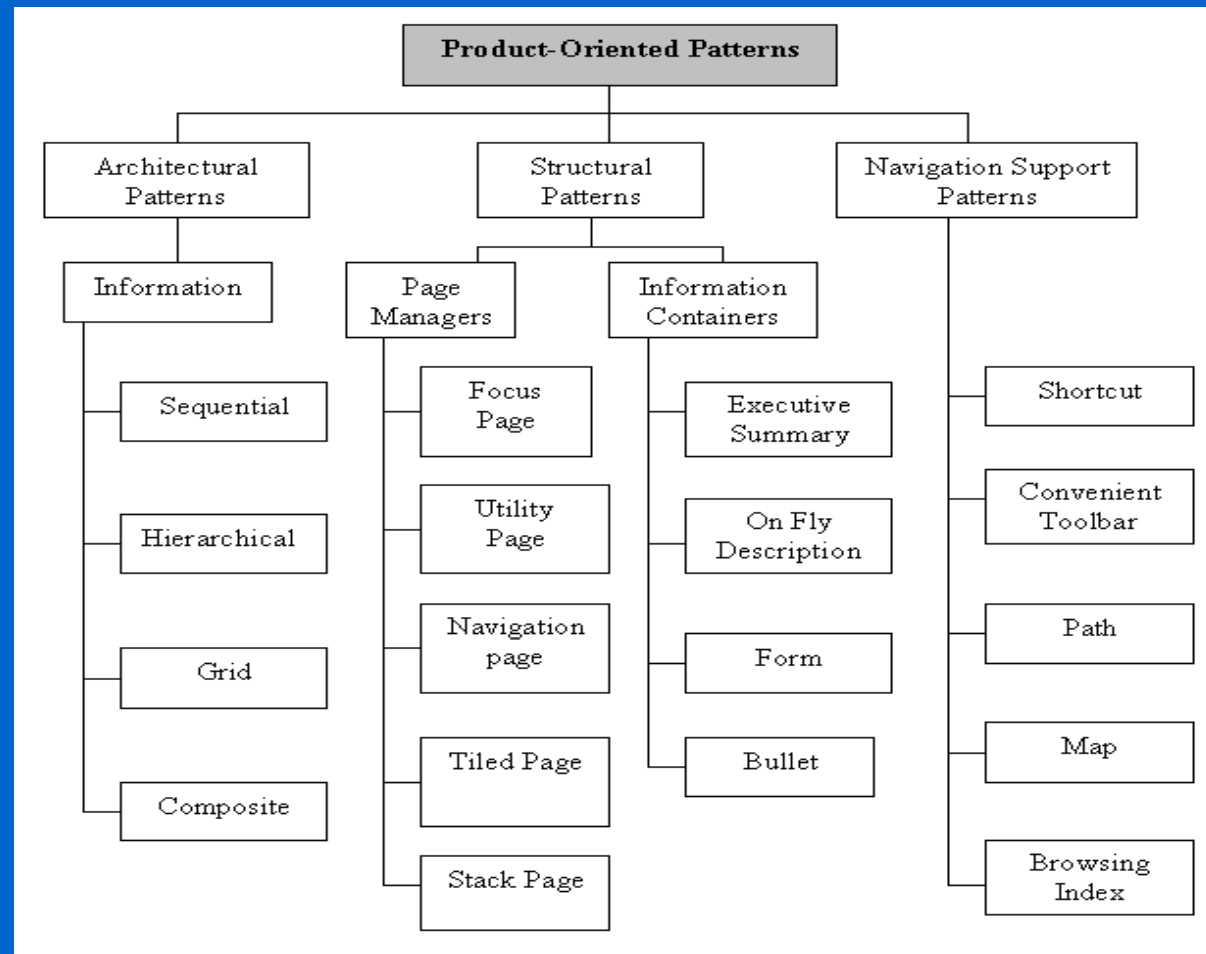
- Fournir la connaissance de conception relative aux systèmes interactifs et leurs utilisateurs
 - ➔ Propose une solution pour une Interface Utilisateur valable pour un problème spécifique d'utilisabilité dans un contexte particulier d'utilisation
 - ➔ Capture user l'expérience des utilisateurs et des meilleures pratiques de conception
- Bonne alternative aux traditionnelles règles de conception (guidelines)
 - ➔ Concrets et orientés problèmes
 - ➔ Avec un format de description spécifique

Exemple de Pattern IHM

Pattern_Name: Convenient Toolbar Pattern Type: Navigation Support	
Context_Use:	User: Novice or Expert Task: Assist the user to reach convenient and key web pages at anytime
Workplace:	Web applications
Usability_Problem:	The user can easily find useful and "safe" pages regardless of the current state of the artefact. The user can reach these pages promptly.
Usability_Factor:	Factor: Efficiency, Safety Criteria: Consistency, Minimal Action, Minimal Memory, User Guidance, Helpfulness
Example:	
Design_Solution:	Group the most convenient action links, such as home, site map, help and etc. Use meaningful metaphors and accurate phrases as labels. Place them consistently throughout the web site.
Other Language Attribute:	Design_Principle Related Usability_Patterns Reading

Langages de Patterns

- Rendre disponible la connaissance contenue dans les patterns
- Description de patterns
- Relations entre patterns



La méthode QOC (Question - Option - Critère)

- Il s'agit d'explorer le plus systématique possible l'espace de solutions correspondant au problème posé.
 - Pour chaque question, qui apparaît lors de la conception du système, les options possibles constituent l'espace de solutions qu'il faut explorer et documenter.
 - *A titre d'exemple, les questions possibles concernant la gestion des salles peuvent être les suivantes : Comment repérer les salles ? Comment les adresser ? Quelles plages horaires à mettre en place, comment les représenter et les adresser ? Par quoi remplir les plages ? Les options correspondent aux solutions proposées.*
 - Les critères qui satisfont ou non à chacune des options sont indiqués.
 - *A titre d'exemple, les critères à prendre en compte peuvent être la lisibilité, la facilité d'écriture, l'encombrement, ...*

QOC Représentation graphique

- Trois types de nœuds sont à représenter : nœud-question, nœud-option, nœud-critère
- Deux types de liens : positifs et négatifs entre les options et les critères sont à mettre en place :
 - Un trait plein (lien positif) indique que le critère est satisfait par l'option considérée.
 - Un trait pointillé indique que le critère s'oppose à l'option choisie.
 - Les épaisseurs des liens peuvent varier selon les poids respectifs des critères sur l'option.
 - L'absence de lien entre options et critères (liens neutres) symbolise l'absence d'influence du critère sur l'option.
- Il est possible d'étudier des questions emboîtées (hiérarchiques).
- Les décisions prises qui correspondent aux options retenues seront entourées.

Dossier contiendra

- Présentation rapide de la problématique Gestion dynamique des voiries
- Présentation des acteurs à prendre en compte et à équiper
- Modélisation des tâches identifiées avec CTTE pour ces acteurs
- Conception d'IHM pour ces acteurs
- Etudier et justifier les choix de patterns à prendre en compte
- Evaluation des choix par la méthode QOC

- Date limite de rendu : la semaine du 14 décembre 2010
- Présentation et discussion du travail au cours de l'entretien programmé dans le courant de la 2^e semaine de janvier 2011