



Modélisation et utilisation de traces d'expérience d'utilisation de systèmes informatiques

Yannick Prié

Equipe « Cognition, Expérience et Agents situés »
LIRIS – Université Claude Bernard Lyon 1

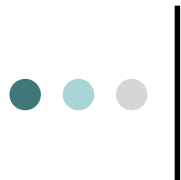
Journée CPER

25 octobre 2005, ENS-LSH, Lyon



Avertissement / plan

- Présentation construite à partir d'un cours de M2R Informatique 04-05
 - extraits + conclusion
- Plan
 - introduction
 - approche Musette
 - discussion / conclusion
 - annexes



Constat

- Pratiques et usages des outils informatiques en mutation
 - massification des usages
 - tâches de plus en plus fondées sur des outils informatiques
 - outils de plus en plus complexes et interconnectés
- Tâches et utilisation de ressources
 - accès croissant aux « ressources » informatiques via des « portails » plutôt que par des applications spécialisées
 - intégration et mobilisation de ressources variées pour des tâches faiblement spécifiées et difficiles à assister
 - « évidence » de l'effet de contexte dans les usages
- Liens conception / usages
 - conception des outils (fatalement) en décalage avec l'utilisation
 - tension entre simplicité d'utilisation et adaptabilité aux besoins dans les pratiques



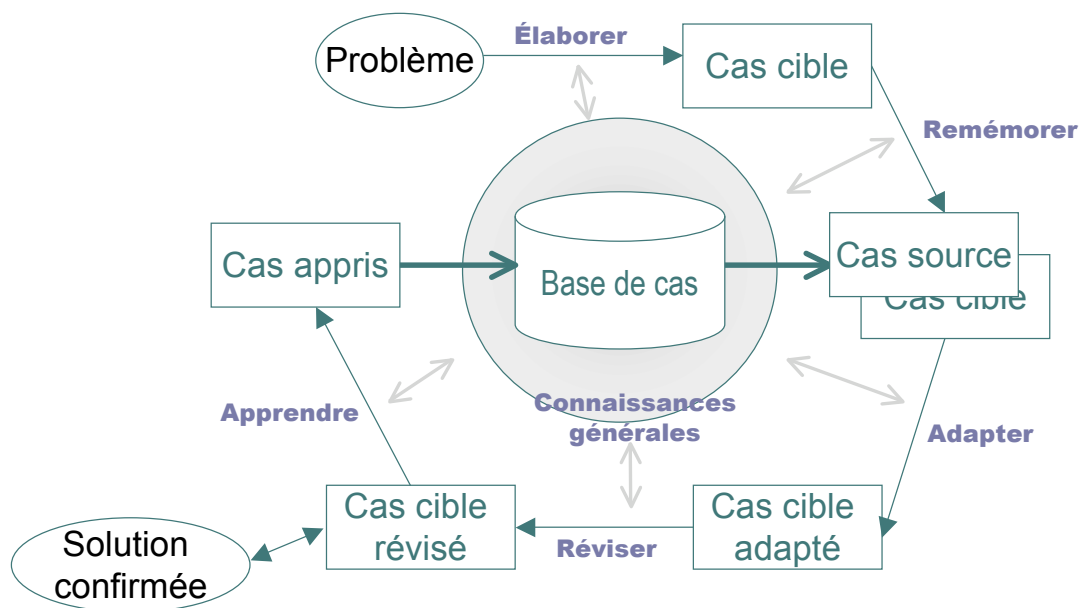
Objectif

- Objectif général recherché
 - construire des systèmes qui fonctionnent en intelligence avec leurs utilisateurs.
 - interfaces graphiques / assistants
 - comprendre et analyser les usages
 - outils conceptuels et informatiques pour l'analyse
- Approche générale
 - utiliser l' « expérience d'utilisation » des systèmes
 - pour améliorer leur fonctionnement, faciliter les tâches des utilisateurs qui les utilisent
 - pour analyser celle-ci *a posteriori* en tant que trace de l'usage
 - « expérience d'utilisation »
 - traces concrètes d'interaction entre système et utilisateur

● ● ● | Du RàPC *au* RàPE

- Equipe originale
 - construction de différents systèmes fondés sur le RàPC
- Raisonnement à partir de cas
 - un cas représente une instance concrète d'un problème et de sa solution
 - idée
 - stocker des cas sous la forme problème/solution
 - réutiliser ces cas comme solutions partielles pour de nouveaux problèmes
- Cycle du raisonnement à partir de cas
- Variantes et améliorations diverses

● ● ● | Le cycle du RàPC



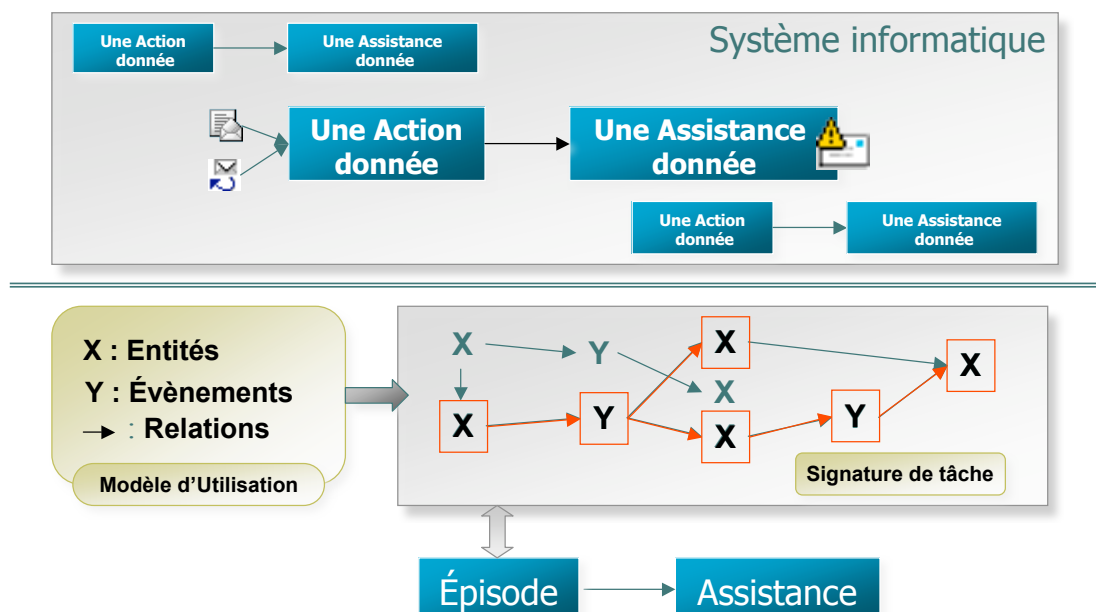


Du RàPC au RàPE

- Limitations du RàPC
 - structure des cas définie a priori, figée
 - connaissances d'explication acquises à l'avance
 - finalement
 - faibles possibilités d'adaptation au contexte, à l'utilisateur
- Idée
 - se donner la possibilité de définir des cas en fonction des besoins et des contextes d'utilisation, dont la liste n'est jamais figée
- Pour cela
 - enregistrer une trace correspondant à une tâche générique d'utilisation du système
 - pouvoir y considérer à volonté des épisodes d'utilisation



Du RàPC au RàPE





Du RàPC au RàPE

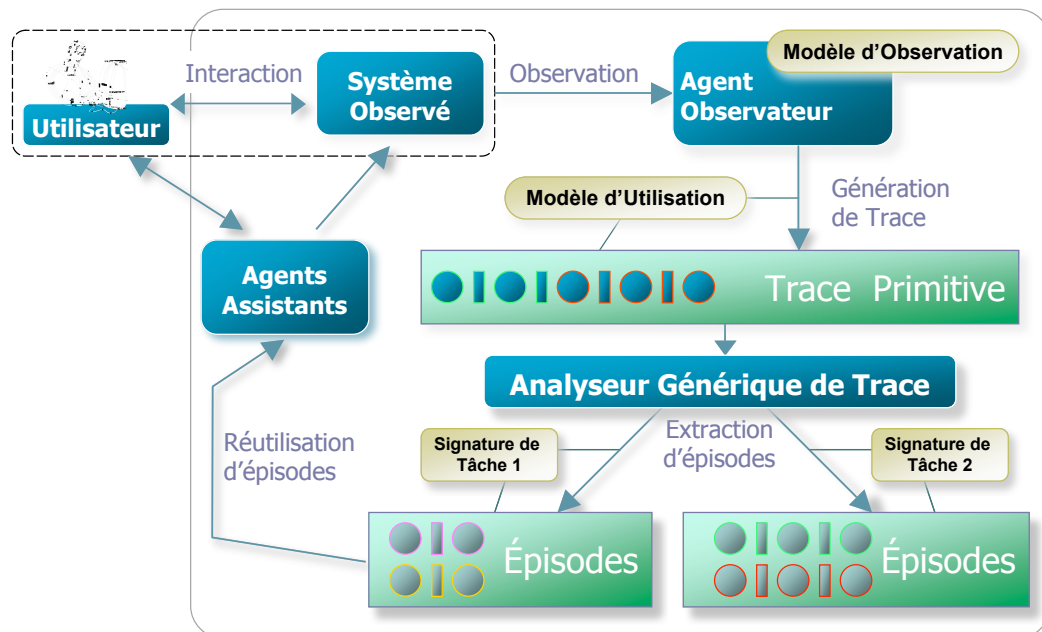
- RàPE
 - un cas est un épisode découpé dans la trace
 - un problème peut s'exprimer en fonction de ce qu'a déjà fait l'utilisateur
 - assistance : appel à l'aide, assistance fondée sur l'expérience des épisodes précédents
- Concepts
 - modèle d'utilisation
 - ensemble des descripteurs d'un système informatique
 - trace d'utilisation
 - instances des descripteurs issus de l'utilisation du système
 - épisode
 - sous-partie de la trace correspondant à la signature d'une tâche particulière



Attention à la trace en tant que telle

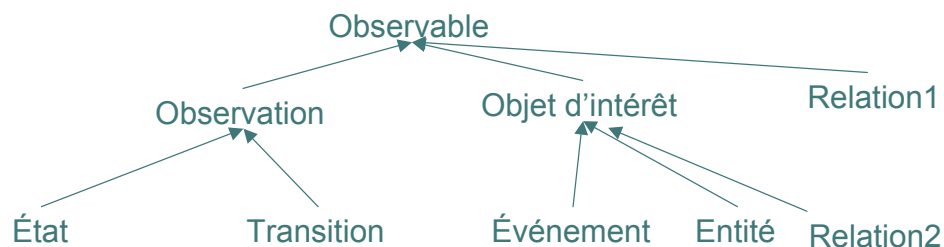
- Attention à l'observateur
 - le modèle d'utilisation n'est pas absolu, il dépend de ce que choisit l'observateur
 - une application peut avoir plusieurs modèles d'utilisation (qui diffèrent d'un éventuel modèle de conception)
 - tout dépend de ce que veut faire le modélisateur de l'expérience dans son système
- Trace primitive
 - niveau fondamental à la charnière entre le concepteur du système fondé sur l'expérience et l'utilisateur
 - attention particulière à lui porter
- La trace pour l'utilisateur
 - l'utilisateur a une pratique courante de la trace, en tant qu'inscription
 - pas forcément besoin de cas

MUSETTE : les grandes étapes



MUSETTE-Base

- « top level ontology » = ensemble de classes à spécialiser en un modèle d'utilisation



Contraintes

- Ordre séquence état/transition
- Etat contient entités
- Transition contient Evénements
- Relations entre objets d'intérêt

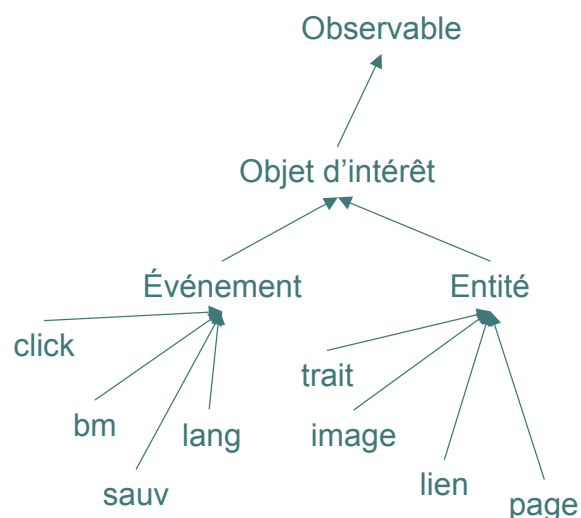
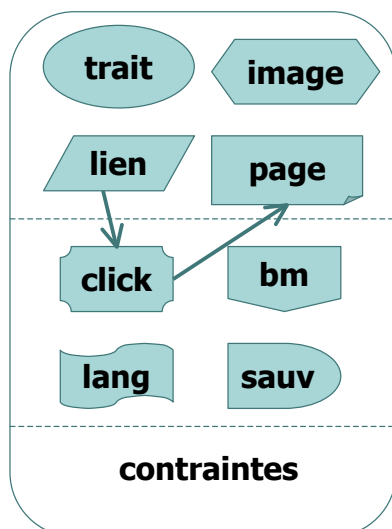


Modèle d'utilisation

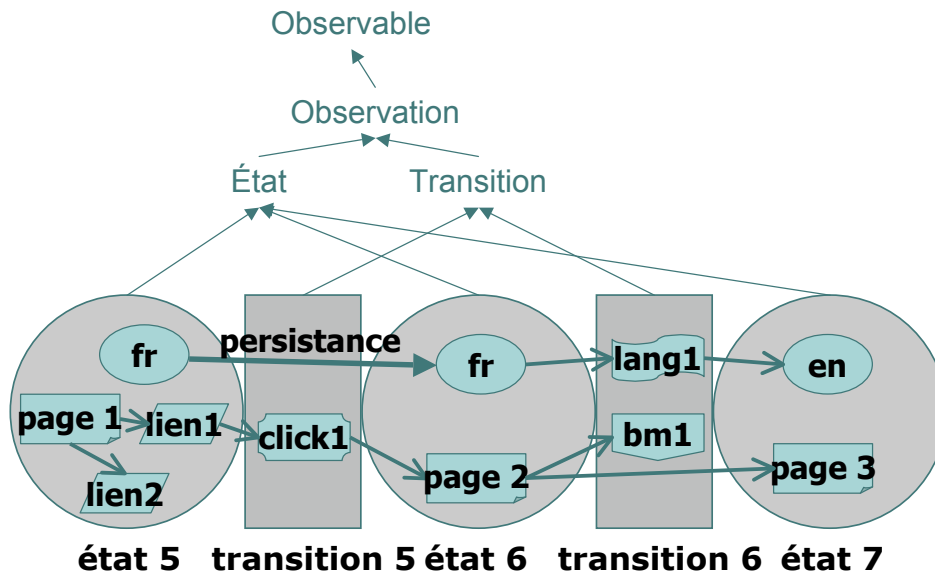
- Ensemble de types d'entités, de types d'événements et de types de relations
- Dans la mesure où le langage d'expression le permet
 - contraintes sur les types (spécialisation, exclusion mutuelle...)
 - contraintes sur les relations (domaine et co-domaine, transitivité, relations inverses, ...)
 - contraintes sur la disposition des objets d'intérêt dans les observations



Modèle d'utilisation exemple Web



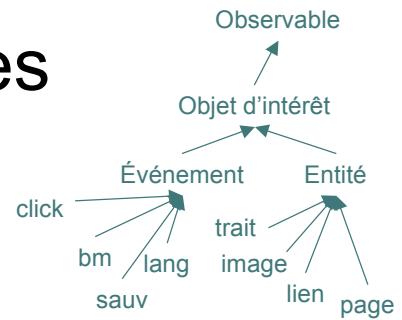
Traces : séquence états-transitions



Signatures de tâches expliquées et épisodes

- Le modèle d'utilisation permet d'inscrire l'utilisation dans une trace primitive
- La trace contient potentiellement des épisodes d'utilisation re-traçant une expérience utilisable pour l'assistance en contexte
- Les épisodes sont repérés dans la trace grâce à des signatures de tâches « expliquées »

Signature de tâches expliquées



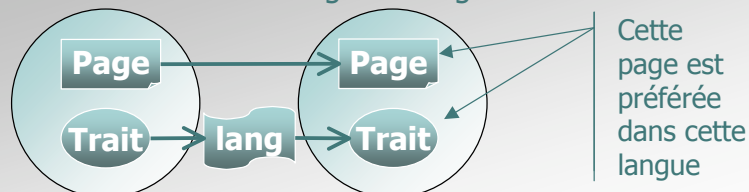
- Composition
 - motif d'objets d'intérêt (OI) dans la trace
 - contraintes sur la position relative des OI dans l'épisode
 - contraintes sur les attributs des OI
 - Annotations
- Explained task signature (EXTASI)

Signatures : exemple

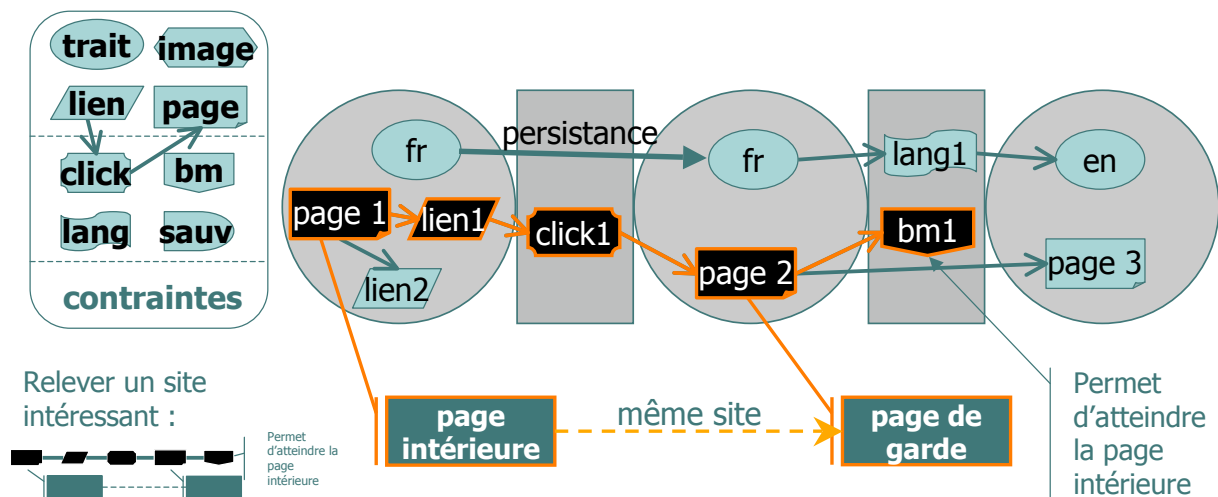
Signature de tâche : Relever un site intéressant



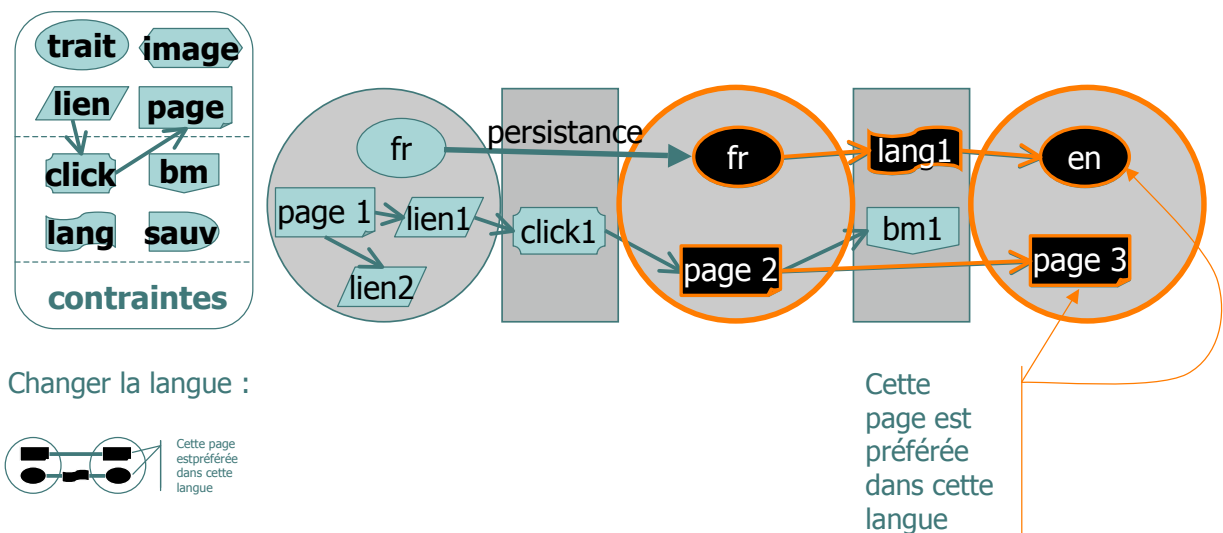
Signature de tâche : Changer la langue



Episodes : illustration 1



Episodes : illustration 2





Scénarios d'assistance

- Assistant spécifique lié à une EXTASI (cf. RàPC)
- Assistant générique
 - moteur d'assistance s'adaptant à une signature de tâche reconnue (paramétrage par une Extasi)
 - la tâche est « réifiée » par les Extasi, et guide aussi bien l'assistant que l'utilisateur
- Assistant générique
 - moteur d'assistance proposant interprétations de la tâche et assistances en fonction de ce que fait l'utilisateur et en repérant des épisodes (partiellement) instanciés
 - les requêtes sont simplement faites en agissant sur le système : moins de biais dans l'expression des besoins
- Assistant générique
 - moteur d'assistance réagissant à une signature de tâche « construite » à la volée par l'utilisateur
 - possibilité de décrire de nouvelles tâches, et de construire sa propre assistance



Applications assistance

- Facilitateurs de tâches liées à l'apprentissage humain (Pixed)
- Facilitateurs de tâches liées à la conception (Dassault)
- Facilitateurs de tâches de veille technologique (Amowebea)
- Facilitateurs de tâches collaboratives (Projets OSCAR, ISOCELE)
- ...



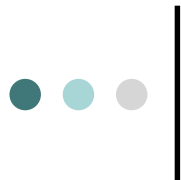
Problématiques courantes

- Questions de méthodologie
 - comment définir un modèle d'utilisation ?
- Questions d'éthique
- Modélisation, stockage et exploitation des traces
- Modèles utilisateurs et modèles d'utilisation
- Auto-confrontation à la trace
- Collecte de traces
- Musette-analyse
- Musette-analyse et vidéo
- ...



Thèses en cours à CEXAS

- Apprentissage de langages et traces
 - Arnaud Stuber
- Traces de conduite automobile
 - Olivier Georgeon
- Théorie de la trace, outils d'utilisation de traces dans les EIAH
 - Lotfi Sofiane Settouti (projet Traces-EIAH cluster Région)
- Traces et collaboration dans les espaces documentaires
 - Julien Laflaquière
- Trace et auto-confrontation
 - Magali Ollagnier-Beldame



Conclusion

- Genèse et présentation de « Musette assistance »
- A suivre
 - Musette-analyse
 - traces Musette, annotation et vidéo
 - confrontation à base de traces entre logique de conception et logique d'usage
 - SET : système d'exploitation à base de traces
 - ...
- Les transparents de la présentation (avec les annexes) sur le web :

<http://liris.cnrs.fr/yannick.prie/download/cper-25-10-2005.pdf>



Annexes

Quatre types d'assistances
Problématiques courantes



Scénario d'assistance 1

- Assistant spécifique lié à une EXTASI
- Exemple
 - notifier quand on arrive sur un site intéressant
 - basé sur l'Extasi « relever un site intéressant »
- Possibilité de RàPC
 - cas = épisode de résolution de problème, reconnu par une signature de tâche
 - assistance à base de cas standard
- Avantage
 - possibilité de partager la même base de cas (trace primitive) entre plusieurs assistants
 - possibilité d'ajouter de nouveaux assistants (nouvelles Extasis) sans changer la trace, en fonction des besoins



Scénario d'assistance 2

- Assistant générique
 - moteur d'assistance s'adaptant à une signature de tâche reconnue
 - paramétré par une Extasi
- Exemple
 - navigation dans un historique expliquée
 - recommandations automatiques
 - mettre un signet
- Avantage
 - la tâche est « réifiée » par les Extasi, et guide aussi bien l'assistant que l'utilisateur



Scénario d'assistance 3

- Assistant générique
 - moteur d'assistance proposant des interprétations de la tâche et des assistances en fonction de ce que fait l'utilisateur et en repérant des épisode (partiellement) instanciés
- Exemple
 - Supposons deux Extasis supplémentaires
 - « raffinement de requête » : deux requêtes successives à un moteur, dont l'une enrichit la première. Première requête annotée par « trop de réponse »
 - « demander de l'aide à quelqu'un » : requête + envoi d'un courriel avec les termes de la requête. Marquage de la requête comme « pas de réponse »
 - Un utilisateur demande de l'aide
 - l'assistant propose « trop de réponses », cette interprétation est rejetée
 - l'assistant propose « pas de réponse », accepté
 - l'assistant propose alors d'envoyer un courrier pour demander de l'aide
- Avantage
 - les requêtes sont simplement faites en agissant sur le système
 - moins de biais dans l'expression des besoins



Scénario d'assistance 4

- Assistant générique
 - moteur d'assistance réagissant à une signature de tâche « construite » à la volée par l'utilisateur
- Exemple
 - au lieu de demander l'aide à quelqu'un, changer de moteur
 - équivalent à décrire sa tâche en cours
- Avantage
 - la tâche est peut-être déjà inscrite dans la trace, mais non connue : la nouvelle Extasi permet alors de proposer de l'aide
 - possibilité de décrire de nouvelles tâches, et de construire sa propre assistance



Questions de méthodologie

- Comment définir un modèle d'utilisation ?
- Eléments méthodologique
 - que chaque observable soit significatif pour l'utilisateur
 - qu'il puisse comprendre la trace primitive
 - que chaque observable soit inclu dans une signature tâche prévue a priori
 - associée à une utilisation pour l'utilisateur
 - que chaque observable soit effectivement observable
 - traitements raisonnables
- Définir en même temps
 - MU
 - Exemples de traces
 - Principes d'observation
 - EXTASI validantes



Deux niveaux de tâches

- Dégagés des modélisations effectuées
- Utilisation de l'outil
 - bas-niveau
 - manipulation de l'interface
 - « envoie requête »
 - proches des événements système
- Tâches plus générales
 - plus haut-niveau
 - « classer son courrier »
 - nécessite une interprétation dans l'observation
- Musette permet d'articuler les deux dans une trace
 - éclairer la manipulation si besoin
 - inscrire la tâche de haut-niveau dans la manipulation



Questions d'éthique

- Pour chaque application
 - que l'utilisateur sache que des traces sont enregistrées
 - qu'il en maîtrise l'utilisation
 - possibilité de les voir
 - possibilité d'enlever des observables ou des parties (ex. attributs)



Modélisation, stockage et exploitation des traces

- Pour chaque application
 - appliquer les principes généraux Musette
 - décider d'un format pour exprimer
 - le modèle d'utilisation
 - les traces
 - stockage
 - les signatures de tâches
 - requêtes
 - compromis
 - expressivité
 - possibilités de recherche
 - a priori, l'instanciation de signatures doit être rapide
 - possibilité de transformations vers des formats plus adaptés à certains types de requêtes



Modèles utilisateurs et modèles d'utilisation

- Beaucoup de champs de recherche définissent des modèles utilisateurs
 - ensemble des propriétés associées à un utilisateur
 - utilisées pour personnaliser
 - Recherche
 - Documents (cf. Documents Virtuel Personnalisables)
- Limites des modèles utilisateurs
 - infiniment affinés pour répondre au besoins
 - en arriver à des modèles psycho-sociologiques ?
 - les besoins varient pour un même utilisateur
- Approche Musette
 - les modèles d'utilisations sont au moins aussi pertinents que les modèles utilisateurs
 - se concentrent sur la tâche en cours (réelle)
 - Interaction utilisateur / système



Auto-confrontation à la trace

- Traces
 - possibilité de lire et relire son histoire récente
 - ce n'est pas nouveau
 - historiques
 - relire son histoire de navigation
 - plus généralement,
 - toute tâche laisse des inscriptions matérielles plus ou moins volontaires qui seront à leur tour utilisées pour coordonner l'action (poursuite de la tâche en cours, « synchronisation » d'une tâche avec les tâches passées)
- Musette
 - outils fondés sur les traces en tant que telles
 - se donner les moyens de l'interprétation
 - signatures pré-définies ou définies au besoin
- Thème de recherche en soi
 - comment présenter ?
 - comment piloter la présentation ?
 - raconter une histoire vs percevoir des configurations
 - etc.



Collecte des traces

- Les traces primitives
 - sont construites par observation de l'interaction système / utilisateur
 - selon un modèle d'observation
 - par un agent observateur
- Qu'y a-t-il dans un agent observateur ?
 - en machine
 - on part des événements du programme
 - au final
 - des entités et des événements, organisés dans des états et des transitions, mis en relations
- Comment générer les traces ?
 - agents ad-hoc
 - agents paramétrables



Collecter des traces

- Deux agents à considérer au minimum
 - agent en ligne
 - trace l'utilisation pendant l'interaction
 - récupère des événements systèmes
 - enregistre la trace brute
 - événements / entités
 - doit récupérer *toute* l'information nécessaire
 - ce qui est en mémoire à un instant donné pourra disparaître rapidement
 - agent hors-ligne
 - traduction de la trace brute en trace primitive pour construire cette dernière
 - plus de temps pour les calculs
 - crée entités, événements, états et transitions
 - ajoute les relations



Collecter des traces (2)

- Agent en ligne
 - « embedded » (comme les journalistes)
 - nécessite d'intervenir dans le code de l'application
 - géré par le programmeur de/des l'application
 - hors-projet
 - lequel veut bien enregistrer des événements, des attributs, mais pas plus...
 - dans le projet
 - on peut lui demander plus
 - extérieur
 - agent de traçage qui tourne en même temps que le système dont on étudie l'interaction avec l'utilisateur
 - nécessite d'accéder au système
 - applications
 - système d'exploitation
 - impossible sans API d'interrogation
- On peut avoir les deux en parallèle



Collecter des traces (2)

- Agent hors-ligne
 - différents degrés de complexité
 - réécriture simple
 - traitements plus complexes
 - la question du paramétrage
 - possibilité d'un agent générique ?
 - paramétrage par un fichier de configuration qui spécifie ce qui est intéressant ou non, les réécritures et transformation nécessaires
 - on n'en est pas encore là
 - agents *ad hoc*



Collecter des traces (3)

- Modèle d'observation en conception
 - associé à la spécification du MU
 - ensemble de règles décrivant comment fabriquer la trace primitive
 - i.e. quel(s) événement(s) machine conduiront à quelles entités, événements, changement d'état, relations dans la trace primitive
 - description textuelle
- Modèle d'observation réel
 - dans l'agent en ligne
 - dans l'agent hors ligne
 - dans d'éventuels fichiers de configuration
- Evidemment
 - nécessité de tout documenter pour pouvoir faire évoluer



Musette et l'analyse

- Trace d'utilisation
 - assistance à l'utilisateur fondée sur les traces
 - mais aussi outil d'analyse pour le chercheur
 - pourquoi ?
 - prise en compte du niveau de l'observateur
 - autonomisation de la trace
 - remarque
 - dans la lignée de l'accent mis sur les études d'usage au niveau national (laboratoires et plateformes d'études des usages / appels à projets)
- Musette-analyse
 - approche Musette pour l'analyse de traces, donc de tâches et d'activités
 - analyse seule
 - analyse en vue de / avec assistance



Musette-analyse : méthodologie d'observation ?

- Il ne s'agit plus d'assister :
 - quelle validation pour le MU ?
- Réutilisation possible du critère de compréhension utilisateur
 - les observables de l'analyste doivent être compréhensibles par l'utilisateur pour être pertinents a priori dans son analyse
 - encore plus valide si on espère mettre en place aussi une assistance
- Construction d'autres critères
 - spécifiques à chaque domaine de recherche
 - souvent déjà en place → les adapter à des observations instrumentées par Musette
- Validation a priori par participation à une signature liée à une hypothèse
 - possibilité de validation de celle-ci



Musette-analyse : outils d'analyse de traces

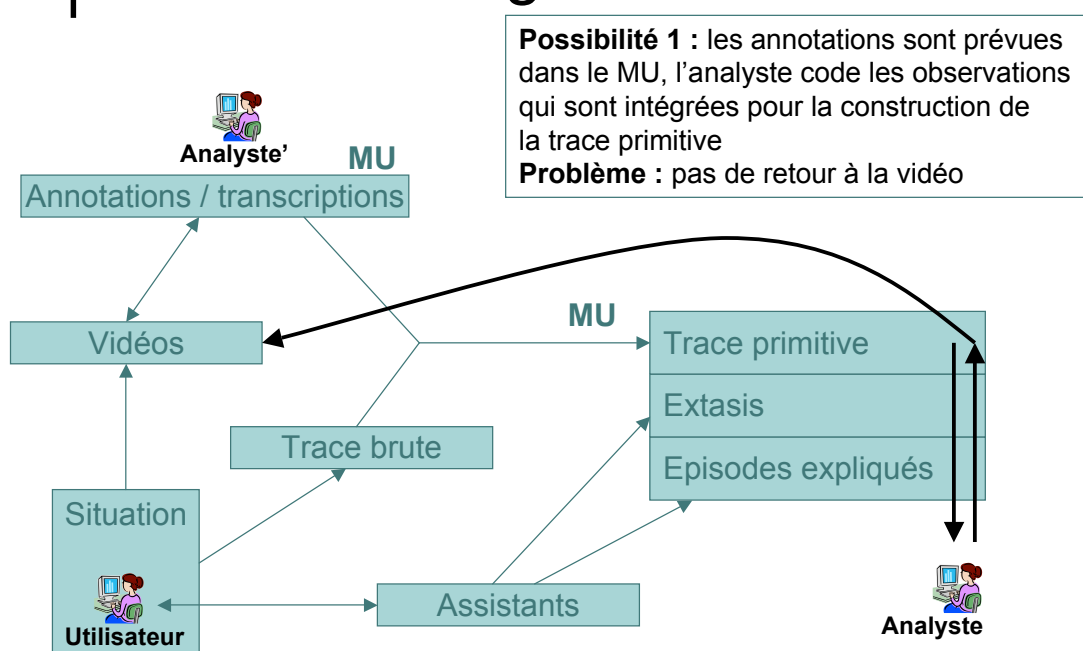
- Analyse directe de la trace
 - présentation / navigation
- La signature de tâche comme requête de l'analyste
 - vérifier les hypothèses initiales ayant présidé à la construction de la trace
 - construire et valider des hypothèses nouvelles
 - se rapproche des travaux orientés analyse assistée par ordinateur
 - linguistique de corpus particulièrement
- Autres possibilités
 - fouille de traces
 - recherche de motifs fréquents / non fréquents
 - avantage par rapport à fouille de données classique : la trace est déjà signifiante en soi
 - remettre en cause le passage de la trace brute à la trace primitive

Musette-analyse : et la vidéo dans tout ça ?

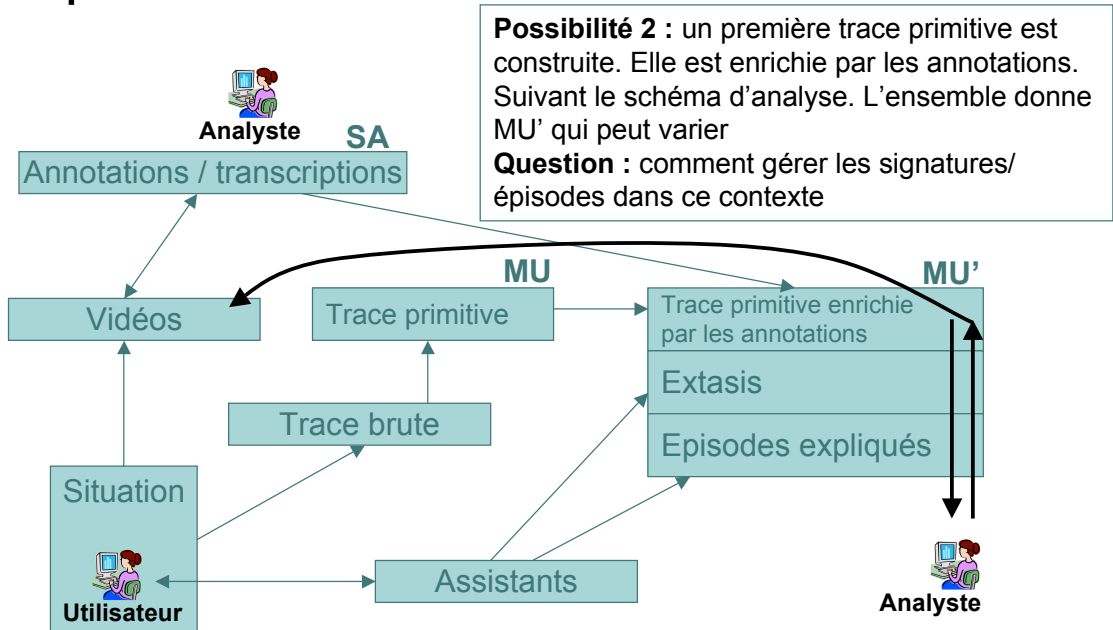
- Un type d'observation que peut faire un analyste
 - utilisation d'un système, seul, en binôme
- Résulte déjà de choix, en fonction d'une problématique
 - choix de la situation
 - acteurs, moment, ...
 - placement de la (des) caméra(s)
 - cadrage (qui laisse le reste hors-champs)
 - analyse
 - transcription, annotation, construction d'une problématique
 - des similitudes avec Musette
 - mais une trace reste formelle
 - beaucoup moins riche que le magma d'une vidéo
 - moins riche
- La question
 - comment articuler l'utilisation de vidéo pour l'analyse avec une approche Musette
 - en tenant compte des particularités irréductibles de la vidéo

L'analyste' est une machine !

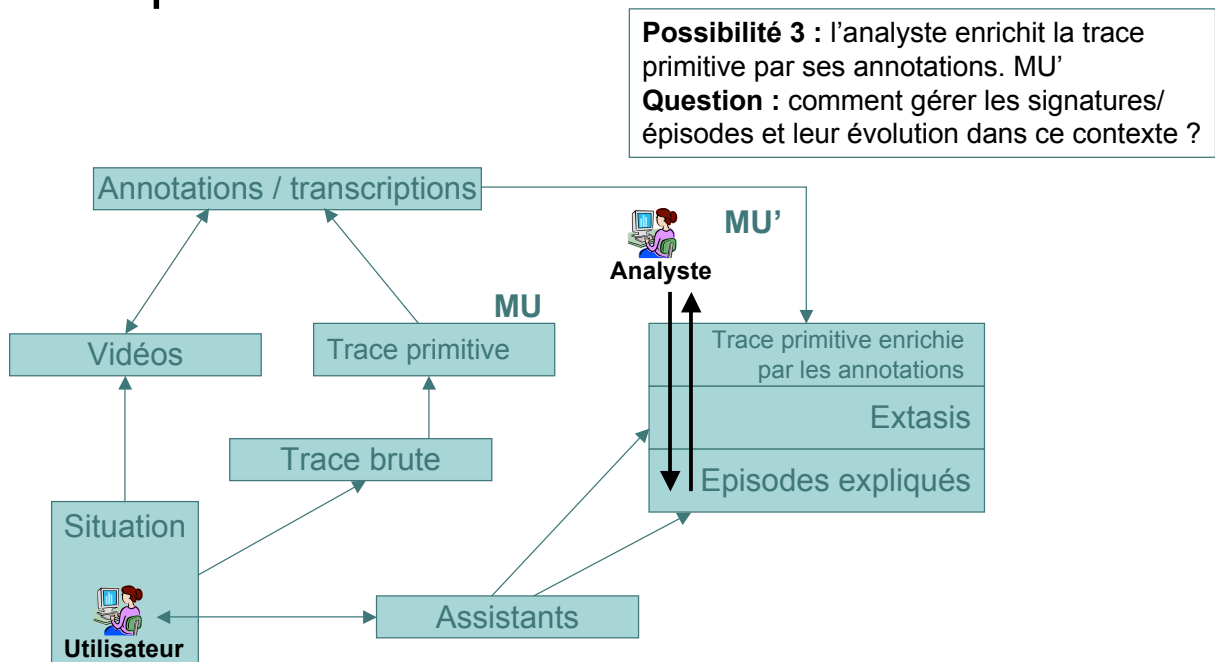
Musette-analyse : vers une vue globale



Musette-analyse : vers une vue globale



Musette-analyse : vers une vue globale





Musette-analyse : conclusion partielle

- Il reste du travail
- Vidéo et audio même combat
- A suivre...



Traces, expériences, connaissances

- L'utilisateur mobilise des connaissances dans le cadre de la réalisation de sa tâche
- Ces connaissances sont « inscrites » dans les supports qu'il manipule entre autres via un système informatique
 - elles sont ainsi inscrites en contexte
 - au moins en tant qu'indices
- Les traces de manipulation des supports d'inscription de connaissances « signent » leur contexte d'usage
- Les traces d'utilisation sont des conteneurs d'expériences de mise en évidence de connaissances



Assister la manipulation de la connaissance / expérience ?

- Dans le cas de tâches faiblement ou pas modélisées a priori
- En évoluant dynamiquement selon l'expérience d'utilisation captée
- En facilitant la réutilisation de l'expérience pour manipuler efficacement les inscriptions de la connaissance
- En facilitant le partage et la formation mutuelle sur la base des épisodes de manipulation de connaissance ainsi « concrétisée ».
- Emergence du sens dans la trace