

# Utilisation de SMV pour la vérification de propriétés d'IHM multimodales

Nadjet KAMEL<sup>†\*</sup>

<sup>†</sup> LRIA/USTHB  
BP 32,  
El Alia, Bab Ezzouar, 16111, Alger, Algerie  
<sup>\*</sup> LISI/ENSMAS  
BP 40109  
86961, Futuroscope Cedex, France  
[nkamel@wissal.dz](mailto:nkamel@wissal.dz) , [kamel@ensma.fr](mailto:kamel@ensma.fr)

## RESUME

Dans cet article nous nous intéressons à l'utilisation de la technique de vérification sur modèle pour vérifier les propriétés CARE des IHM multimodales. Les systèmes de transitions ainsi que la logique temporelle CTL sont mis en œuvre pour représenter un système multimodal et les propriétés associées. Le contrôleur sur modèle SMV assure la vérification des propriétés exprimées en CTL.

**MOTS CLES** : Interactions Multimodales, Méthodes Formelles

## ABSTRACT

This paper addresses the use of the model-checking technique to check the CARE properties of input multimodal interactions. Transition systems and temporal logic are used to specify a multimodal system and its properties. The properties of the multimodal interface are specified in the CTL temporal logic and checked by the SMV model checker.

**CATEGORIES AND SUBJECT DESCRIPTORS:** D.2 [Software Engineering] : Model Checking.

**GENERAL TERMS:** Verification, Experimentation

**KEYWORDS** : Multimodal Interaction, Formal Methods

## INTRODUCTION

Avec la puissance des machines et le progrès des systèmes de reconnaissance de la parole et de synthèse vocale qui ont permis d'augmenter les capacités sensori-motrices et représentationnelles des systèmes, nous assistons au développement d'applications où les moyens de communication et d'interaction sont de plus en plus

variés. Dans les interfaces multimodales, l'utilisateur peut interagir en utilisant un ou plusieurs moyens de communication, avec l'application, tels que la parole, le geste, la manipulation directe etc. Dans ce type d'interfaces, les interactions peuvent être utilisées de manière séquentielle, parallèle, indépendante ou bien combinées de manière synergique. Dans ce domaine, les travaux de recherche ont identifié plusieurs concepts liés aux systèmes multimodaux (mode, modalité,...) sans atteindre un consensus sur les termes qui les désignent. Nous retrouvons dans [7] une synthèse et une mise au point sur la terminologie et sur les référentiels applicables à la conception d'interfaces multimédias et multimodales. Dans ce domaine, plusieurs travaux de recherche s'intéressent à différents aspects. Certains s'intéressent à l'implémentation d'applications multimodales [2], d'autres à la définition de concepts relatifs à la multimodalité [7] et d'autres aux propriétés d'interfaces multimodales [3]. Les techniques formelles basées sur les modèles mathématiques permettent l'analyse et la vérification des systèmes d'une manière rigoureuse et sans ambiguïté. L'utilisation de ces techniques dans le domaine des IHM multimodales (IHM3) reste rare et c'est dans ce contexte que se situe notre travail. Nous nous intéressons à l'utilisation des techniques formelles pour la modélisation de l'interaction homme machine en entrée. Les systèmes de transitions ainsi que la logique temporelle CTL sont mis en œuvre pour représenter un système multimodal et les propriétés associées. Le contrôleur sur modèle SMV[9] assure la vérification des propriétés exprimées en CTL. Les propriétés que nous exprimons puis vérifions sont des cas de propriétés CARE[3] identifiées pour les IHM3.

## IHM MULTIMODALES (IHM3)

L'interaction dans les IHM3 est complexe dans le sens où elle est définie par plusieurs événements provenant de plusieurs canaux d'entrée, ce qui engendre un fort degré de parallélisme. Plusieurs modalités peuvent participer à la réalisation d'une interaction élémentaire : un *énoncé*. Ces énoncés participent à la réalisation de *tâches* et *sous tâches* d'interactions utilisateurs plus complexes. Les IHM3 peuvent être classées selon plusieurs critères

combinés. Une classification, donnée dans [1], repose sur les trois critères : production des énoncés (séquentiel ou parallèle), usage des médias (exclusif ou simultané) et nombre de médias par énoncé. Cette combinaison a produit plusieurs types de multimodalité. Par exemple, dans le cas où une seule modalité est utilisée par énoncé et la production des énoncés est séquentielle on parle de type *exclusif*.

### PROPRIÉTÉS DES IHM3

Bien que les propriétés liées à la validité et à la robustesse soient utiles pour caractériser les systèmes interactifs, elles ne couvrent pas complètement les spécificités des nouvelles technologies d'interaction telles que les IHM3. Afin de palier cette insuffisance, de nouvelles catégories de propriétés ont été définies pour les IHM3. Ce sont les propriétés dites CARE [3] (Complémentarité, Assignation, Redondance, Equivalence).

### UTILISATION DES APPROCHES FORMELLES POUR LES IHM3

Peu de travaux mettant en œuvre des techniques formelles se sont intéressés à la modélisation, à la vérification et à la validation d'IHM3. Trois approches ont retenu notre attention. En premier, les travaux [8] utilisent les interacteurs pour modéliser l'interface de l'application multimodale *Matis* à partir d'un modèle de tâche défini avec UAN. Les interacteurs ont été implémentés en Lotos. Un ensemble de propriétés a été vérifié à l'aide de la logique temporelle ACTL avec le contrôleur sur modèle de l'outil Lite. Ensuite, [4] et [5] montrent comment des techniques telles que Z et CSP peuvent être utilisées pour modéliser une application multimodale. Dans ce travail, les auteurs font abstraction des modalités, et évitent la prise en charge de toutes les spécificités d'une interface multimodale. Les problèmes de fusion et de vérification des propriétés CARE ne sont pas traités. Enfin, [10] utilisent le modèle ICO pour représenter la fusion de deux modalités : le geste et la parole, mais ne traite pas de la vérification de propriétés spécifiques aux IHM3.

Notre travail propose l'utilisation des systèmes de transitions pour modéliser l'interaction en entrée dans une IHM3 selon le type de multimodalité. Les propriétés de l'interface sont modélisées en logique temporelle et la vérification est effectuée à l'aide de la technique de vérification sur modèle. Cette dernière, connue par model-checking, est une technique basée sur la construction d'un modèle fini du système et sur la vérification des propriétés sur ce modèle par énumération exhaustive de l'espace d'états.

Pour illustrer cette approche, nous avons utilisé le contrôleur sur modèle SMV. Le système de transitions représentant le système est décrit dans le langage d'entrée de SMV et les propriétés sont modélisées dans la logique temporelle CTL. Nous avons choisi de modé-

liser les interactions d'entrée et vérifier les propriétés CARE de l'application multimodale *Matis* [6]. Cette application permet d'interroger une base de données sur les programmes de vols d'une ville de départ vers une ville de destination. Plusieurs possibilités sont offertes à l'utilisateur pour formuler ses requêtes. Trois modalités sont possibles : manipulation directe avec la souris, saisie de la requête dans un formulaire ou prononcer une phrase.

### MODÉLISATION DE L'INTERACTION MULTIMODALE

La syntaxe du modèle que nous proposons est donnée par une grammaire inspirée des algèbres de processus. Les énoncés sont paramétrés par les modalités utilisées dans leur réalisation. Notons  $A_{mi}$ , l'ensemble des événements  $ei$  produits par la modalité  $mi$ . Ainsi, si on dispose

de  $n$  modalités, on pose  $A = \prod_{i=1}^n A_{mi}$  et on note  $e$  les

éléments de  $A$ .

Nous proposons de modéliser une partie de l'interaction de l'utilisateur avec l'interface de l'application multimodale *Matis*. Nous avons choisi de présenter dans cet article la modélisation de la tâche de formulation d'une requête dans le cas où l'interface serait de type *exclusif*. D'autres modèles peuvent être définis selon le type de multimodalité, où de nouveaux opérateurs seront définis tel que l'opérateur du parallèle. La syntaxe est donnée par la grammaire suivante :

$$T ::= T [] T \mid T \gg T \mid E_{Amk} \text{ avec } Amk \subseteq A$$

$$E_{Ami} ::= ei ; E_{Ami} \mid \delta \text{ avec } ei \in A_{mi}$$

Les opérateurs  $[]$  et  $\gg$  définissent respectivement le choix et la séquence entre tâches. L'opérateur  $;$  définit le préfixage d'un énoncé par un évènement. La sémantique formelle des différents opérateurs est donnée par la technique formelle sous jacente, dans notre cas les systèmes de transitions.

**Application à l'étude de cas.** Nous définissons l'ensemble des évènements générés pour chacune des modalités *parole* et *manipulation\_directe*.

$$A_{parole} = \{ \langle \text{show me flights} \rangle, \langle \text{from this City} \rangle, \langle \text{to This City} \rangle, \langle \text{From Boston} \rangle, \langle \text{To Oslo} \rangle \}$$

$$A_{manipulation\_directe} = \{ \text{Click\_Boston}, \text{Click\_Oslo}, \text{Click\_From}, \text{Click\_To}, \text{Click\_n\_requête} \}$$

Avec:

1. *Click\_Boston, Click\_Oslo* : click de la souris, respectivement, sur la ville *Boston* et sur la ville *Oslo* dans une liste de villes ;
2. *Click\_From, Click\_To* : click de la souris sur le champ texte respectivement, *From* et *To* du formulaire de saisie d'une requête ;
3. *Click\_n\_requête* : click de la souris sur le bouton attribué à la création d'une nouvelle requête. Le click sur ce bouton engendre la création d'une nou-

velle fenêtre de formulaire de saisie d'une nouvelle requête.

Nous donnons :

1. les expressions de l'ensemble des *énoncés* en fonction des *événements* de base :

```
Créer1_req = « show me flights » ; δ
Créer2_req = Click_n_requête ; δ
Remplir1_From = « from Boston » ; δ
Remplir2_From = Click_From ; Click_Boston ; δ
Remplir1_To = « to Oslo » ; δ
Remplir2_To = Click_To ; Click_Oslo ; δ
```

2. l'expression de la tâche *Formuler\_requête* en fonction des *énoncés* :

```
(Créer1_req [!] Créer2_req)
>> (Remplir1_From [!] Remplir2_From)
>> (Remplir1_To [!] Remplir2_To)
```

Le système de transitions de la tâche *Formuler\_requête* est obtenu par la composition des systèmes de transitions des énoncés qui la composent. Le système résultant est décrit en figure 1.

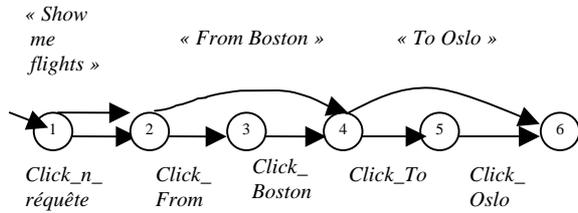


Figure 1 : Système de Transitions Etiquetées de la tâche *Formuler\_requête*

Pour vérifier les propriétés CARE, il faut raisonner sur les modalités des actions effectuées. Pour cela, il est utile de remplacer les étiquettes des actions, dans les systèmes de transitions, par leurs modalités respectives. Le système de transitions obtenu est une abstraction correcte du système d'origine (une sorte de typage). Il est codé dans le langage d'entrée du contrôleur sur modèle SMV dans les clauses *VAR*, *ASSIGN* et *TRANS* (figure 2). Deux variables d'état caractérisent ce système. La variable *etat* représente l'état global du système. La variable *Mod* représente la modalité utilisée pour réaliser les actions et transiter d'un état à un autre dans le système de transitions. L'opérateur *next* appliqué à une variable définit sa valeur suivante dans le système. La valeur de la variable *etat* est définie comme sur le système de la figure 1. On remarque le non déterminisme dans le cas où la valeur de la variable *etat* est égale à 2 ou 4. La relation *TRANS* définit la relation de transition qui relie les variables *etat* et *mod*. Par exemple, dans le cas où la valeur suivante de la variable *etat* est égale à 3 la valeur suivante possible de la variable *mod* est la modalité manipulation.

```
MODULE main
-- Déclaration et typage des variables du système

VAR
etat : {1,2,3,4,5,6}
mod : {parole, manipulation}

-- Définition du système de transitions

ASSIGN

-- Initialisation
init(etat) := 1;
init(mod) := {parole, manipulation};

-- Définition de l'état suivant.
next(etat) :=
case
etat=1 : 2; etat=2 : {3, 4}; etat=3 : 4; etat=4 : {5,6}; etat=5 : 6; 1 : etat;
esac;
TRANS next(etat) = 2 -> ((next(mod)=parole) or (next(mod)=manipulation))
TRANS next(etat) = 3 -> (next(mod)=manipulation)
TRANS next(etat)= 4 -> ((next(mod)= parole) or (next(mod)=manipulation))
TRANS next(etat)=5 -> (next(mod)=manipulation)

-- Définition de propriétés
SPEC
EG (((etat=1) and (mod=parole))
=> E ((mod=parole) U (etat=6)))
⇔
EG (((etat=1) and (mod=manipulation))
=> E((mod=manipulation) U (etat=6)))
```

Figure 2 : Code SMV du Système de Transitions Etiquetées de la tâche *Formuler\_requête*

### PROPRIÉTÉS

Les propriétés vérifiées par le contrôleur sur modèle SMV sont exprimées dans la logique temporelle CTL dans la clause *SPEC*. Les formules de CTL sont interprétées sur les chemins d'exécution du système de transitions. En plus des opérateurs de la logique du premier ordre, d'autres opérateurs temporels sont définis. Les opérateurs utilisés dans cet article : *EG*, *U* et *E* signifient de manière informelle, respectivement, « il existe toujours un chemin où », « jusqu'à » et « il existe un chemin où ».

**Application à l'étude de cas.** Nous avons modélisé et vérifié deux propriétés CARE : la propriété d'*équivalence* et la non *complémentarité* des deux modalités *parole* et *manipulation directe*.

1. l'*équivalence* des deux modalités est définie par la formule de logique temporelle CTL suivante :

```
EG (((etat=1) and (mod=parole))
=> E ((mod=parole) U (etat=6)))
⇔
EG (((etat=1) and (mod=manipulation))
=> E((mod=manipulation) U (etat=6)))
```

Elle exprime formellement que

- l'existence d'une exécution qui démarre de l'état 1 et qui se termine à l'état 6 et dont les transitions sont toutes effectuées par la modalité *parole* est équivalente à
- l'existence d'une exécution qui démarre de l'état 1 et qui se termine à l'état 6 et dont les transitions sont toutes effectuées par la modalité *manipulation\_directe* notée ici *manipulation*.

Cette propriété est vérifiée par le système de la figure 1.

2. la *complémentarité* des deux modalités est définie par la formule de logique temporelle CTL suivante :

```
EG (
  ((etat = 1) and ((mod = parole) or (mod = manipulation)))
=> E (((mod=parole) or(mod = manipulation)) U (etat=6))
)
and
not EG ((etat = 1) and (mod=parole))
      => E ((mod = parole) U (etat =6))
)
and
not EG (((etat=1) and (mod = manipulation))
      => E ((mod= manipulation) U (etat=6))
)
```

Elle exprime formellement

- l'existence d'une exécution qui démarre de l'état 1 et se termine à l'état 6 et dont les transitions sont effectuées soit par la modalité *parole* soit par la modalité *manipulation\_directe* ;
- l'absence d'une exécution dont les transitions sont toutes effectuées par la modalité *parole* ;
- l'absence d'une exécution dont les transitions sont toutes effectuées par la modalité *manipulation\_directe*.

Cette propriété n'est pas vérifiée par le système de la figure 1. Un contre exemple est généré par SMV.

## CONCLUSION

Dans cet article, nous avons présenté l'utilisation de la technique de la vérification sur modèle pour la vérification des propriétés CARE. Un modèle formel est défini et utilisé pour la description de l'interaction multimodale. Des propriétés CARE sont spécifiées dans la logique temporelle CTL et vérifiées par le vérificateur sur modèle SMV.

Dans un travail futur, nous nous intéressons à l'utilisation de la technique de preuve pour la vérification de propriétés CARE en utilisant une approche fondée sur la preuve comme B. Ensuite, à l'image de la démarche proposée pour la modélisation de l'interaction avec les systèmes de transitions, nous nous efforcerons de modéliser formellement la description des propriétés CARE. Enfin la généralisation de la description des IHM3 en utilisant une démarche générale de description fondée sur les systèmes de transitions pour représenter le système multimodal ainsi qu'une partie des propriétés et une logique pour représenter les propriétés CARE seront étudiées. Dans ce cadre, nous nous intéresserons également à l'apport des techniques fondées sur la preuve

pour réduire l'énumération exhaustive des états du contrôleur sur modèle.

## BIBLIOGRAPHIE

- 1 Bellik Y. « Interfaces multimodales : concepts, modèles et architectures » thèse de doctorat d'université 1995.
2. Cohen P., Johnson M., McGee D., Oviatt S., Pittman J., Smith I., Chen L. & Clow J., " *Quickset: Multimodal Interaction for Distributed Application* ". Proceedings of the fifth ACM Int. conference on Multimedia, ACM Press, 1997.
3. Coutaz J., Nigay L., Salber D., Blandford A., May J. and Young R.M. « *Four easy pieces for assessing the usability of multimodal interaction: the CARE properties* ». In K. Nordby, P.H. Helmersen, D.J. Gilmore and S.A. Arnesen (eds) *Human Computer Interaction: Interact '95*. Chapman and Hall: London. pp. 115-120, 1995.
4. Duke, D. et Harisson, M. D. « *mapping User requirements to implementations* » . In Software Engineering Journal, Vol 10(1), p54-75, 1997.
5. MacColl et Carrington, D. " *testing MATIS : a case study on specification-based testing of interactive systems* ", FAHCI (1998) , pp.57-69, ISBN 0-86339-7948
6. Nigay L. « *Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales* », PhD dissertation, Grenoble University, 1994, 315 pages
7. Nigay L., Coutaz J., « *Espaces conceptuels pour l'interaction multimédia et multimodale* », TSI, spéciale Multimédia et collectif, AFCET & HERMES Publ, Vol 15(9), 1996, p. 1195-1225
8. Fabio Paterno, Menica Mazzanotte. « *Analysing Matis by Interactors and ACTL* ». Amodeus Project Document : system Modelling/WP36, 1994
9. McMillan K. L., " *The SMV system* ", Technical report, Carnegie Mellon University, 1992.
10. P. Palanque, A. Schyn. « *A model-based for Engineering multimodal interactive systems* ». Proceeding of the ninth IFIP TC13 international Conference on Human-Computer Interaction (Interact'2003) sep 1-5, 2003. Zürich, Switzerland.