

# Un logiciel d'exploration de schémas de circuits électriques basé sur l'API MICOLE

*Thomas Pietrzak — Isabelle Pecci — Benoît Martin*

LITA, Université Paul Verlaine — Metz  
57006 Metz Saulcy  
{thomas.pietrzak, isabelle.pecci, benoit.martin}@univ-metz.fr

## RESUME

De nouvelles technologies permettent de proposer de nouvelles interactions, notamment pour les personnes en situation de handicap. Par exemple grâce à de nouvelles technologies, des utilisateurs voyants et déficients visuels peuvent collaborer, notamment dans une salle de classe. Le but du projet européen MICOLE est de fournir une API de développement d'applications multimodales et multi-utilisateurs pour déficients visuels. L'architecture choisie s'articule autour d'un système multi-agents communiquant par un bus logiciel. L'API supporte plusieurs périphériques haptiques : PHANToM, souris VTPlayer, des périphériques d'entrée comme le clavier, la souris, ou encore la SpaceMouse, et gère le rendu sonore. Nous montrons ici une application permettant d'explorer des schémas de circuits électriques construite à partir de cette API.

**MOTS CLES :** Retour de force, affichage haptique, schémas de circuits électriques, déficients visuels, système multi-agents

## ABSTRACT

New technologies allow to propose new interactions, in particular for disabled people. For example, thanks to new technologies, sighted users can collaborate with visually impaired ones, like in classroom situation. The goal of the european project MICOLE is to provide a development API for multimodal and multi-users applications intended for visually impaired users. The choosen architecture is built around a multi-agent system that communicates with a software bus. The API supports several haptic devices : PHANToM, VTPlayer mouse, input devices like keyboard, mouse or SpaceMouse and manages sound ren-

dering. We show here an electric circuit schematic exploration software that is built with this API.

**CATEGORIES AND SUBJECT DESCRIPTORS:** H5.2 [Information interfaces and presentation] : User Interfaces. - input devices and strategies, auditory (non-speech) feedback, haptic I/O, user-centered design.

**GENERAL TERMS:** Human Factors, Design

**KEYWORDS:** Force feedback, haptique rendering, electric circuits schematic, visually impaired users, multi-agent system

## INTRODUCTION

Une enquête que nous avons menée auprès d'enfants déficients visuels et d'éducateurs nous a appris que les principaux besoins pour l'éducation des déficients visuels se situaient au niveau des schémas et des notations mathématiques. Ce sondage a été réalisé au stade préliminaire du projet européen MICOLE [1]. Le but de ce projet est de créer un environnement multimodal et collaboratif destiné aux enfants déficients visuels. Le système doit favoriser la collaboration d'enfants visuels avec des enfants voyants ou des enseignants. Il doit permettre de naviguer dans des données. Dans le cadre de ce projet nous devons développer des applications permettant de combler les lacunes mises en évidence par notre enquête, ainsi que d'une API permettant de développer rapidement ce genre d'application. Nous avons utilisé cette API pour développer un logiciel d'exploration de schémas de circuits électriques (décrit dans cet article), et d'autres partenaires du projet s'en sont servis pour écrire d'autres applications, comme une application d'exploration du système solaire.

Si on observe un système multimodal existant comme celui développé dans le projet TeDUB [3], on constate qu'il est dépendant des périphériques utilisés. Ce système permet d'explorer des schémas. Il est composé d'un interpréteur et d'un navigateur, ce qui permet à un utilisateur de parcourir des diagrammes UML, des circuits électriques, etc. L'utilisateur explore les schémas avec un clavier, un joystick ou une tablette

graphique. Le seul retour de force utilisé est celui des joysticks à retour de force. L'intérêt est le faible coût de ces périphériques. Le désavantage est le manque de précision au niveau du retour haptique. Notre approche consiste à fournir une API qui permet de développer ce type d'application rapidement, et d'avoir une flexibilité qui donne un plus grand choix de périphériques et de techniques d'interaction. Notre système est spécialisé dans l'haptique. L'API MICOLE permet de gérer des périphériques variés plus ou moins précis comme le PHANToM, la souris à picots VTPlayer ou une souris conventionnelle par exemple. Elle est destinée aux applications multimodales et à ce titre elle offre aussi des entrées/sorties sonores (2D/3D) et visuelles (2D/3D). Le son n'est pas détaillé dans cet article car notre application ne s'en sert pas.

Le PHANToM (figure 1 à gauche) est un bras articulé à retour de force. Il permet de capter les déplacements en 3 dimensions et les rotations en 3 dimensions, et de rendre des forces sur les 3 dimensions en translation. La VTPlayer (figure 1 à droite) est une souris munie de 4 boutons sur les côtés, ainsi que de deux cellules de picots  $4 \times 4$  sur le dessus permettant d'afficher des motifs tactiles. L'architecture permet au développeur d'application de gérer ces périphériques sans avoir à utiliser directement une API spécifique. La gestion du PHANToM, de la souris, du clavier, du rendu visuel et du rendu sonore sont gérés par l'intermédiaire de l'API Reachin [4].



Figure 1 : PHANToM Omni et VTPlayer

## TECHNIQUES D'INTERACTION

Des techniques d'interactions utilisant les périphériques décrits ci-dessus ont été créées et intégrées à l'API. La première est PICOB (Post-It haptiques par codes-barres), qui permet d'afficher des informations avec un PHANToM en utilisant des impulsions. Cette technique se décline en deux modes.

Le premier mode est le mode guidé : l'utilisateur est guidé en aller/retour sur quelques millimètres par le PHANToM. Un aller/retour fait ressentir une bosse, qui se paramètre en durée, en amplitude et en direction. Une étude préliminaire [5] a montré que des utilisateurs peuvent reconnaître 6 directions (haut, bas, gauche, droite, avant et arrière) et deux amplitudes (4mm et 16mm) avec une durée fixée à 250ms.

Le second mode est le mode semi-guidé : l'utilisateur ressent des bosses lorsqu'il parcourt une ligne sur la-

quelle il est contraint. Les paramètres sont similaires : il y a toujours la direction et l'amplitude, mais la durée est remplacée par la longueur. Il y a cependant une limitation au niveau des directions vu que l'on ne peut pas utiliser les directions de l'axe sur lequel se déplace l'utilisateur. Une autre étude préliminaire [6] a montré que sur une ligne horizontale allant de droite à gauche des utilisateurs arrivaient à reconnaître 4 directions (haut, bas, avant et arrière) et deux amplitudes (4mm et 16mm), avec une longueur de 5mm.

Une autre technique d'interaction a été développée : les icônes tactiles [7]. Elle utilise la capacité de la VTPlayer à afficher des motifs pour donner des informations. Il y a deux types d'icônes : statiques et dynamiques.

Les icônes statiques désignent des motifs de picots levés ou baissés. Le motif est affiché pendant une durée déterminée soit par le programmeur, soit par l'utilisateur si l'affichage de l'icône est conditionnée par un événement sur lequel l'utilisateur a le contrôle.

Les icônes dynamiques désignent une succession de motifs dont l'affichage est temporisé, donnant ainsi une animation tactile. L'icône peut être jouée une seule fois ou être jouée en boucle.

Un système de guidage a été intégré à l'architecture. Il permet par exemple à un enseignant de montrer à l'élève un endroit de la scène qu'il n'a pas parcouru. Pour cela l'API permet de définir des points de passage par lesquels le PHANToM sera attiré afin de lui faire suivre une trajectoire.

## L'ARCHITECTURE MULTIMODALE

L'architecture de l'API est basée sur un système multi-agents. Chaque agent est une entité autonome qui s'occupe d'une tâche. Les agents peuvent communiquer par un bus logiciel soit en envoyant un message sur ce bus, soit en exécutant du code quand des messages sélectionnés transitent sur le bus. L'envoi de message se fait simplement à l'aide d'une méthode. L'exécution du code se fait en associant une fonction à une expression régulière telle que la fonction est appelée à chaque fois qu'un message correspondant à l'expression régulière est envoyé sur le bus.

Les agents peuvent avoir leur propre boucle d'exécution en parallèle. L'utilité de cette boucle est évidente par exemple pour des agents qui ont pour tâche d'interroger un périphérique par l'intermédiaire d'un driver ou une API et d'envoyer les coordonnées sur le bus.

Le choix du système multi-agents est motivé par la volonté d'afficher la même information de plusieurs manières différentes, et ce de manière totalement autonome. L'idée est de traiter un message purement

sémantique par un ou plusieurs agents successifs jusqu'à arriver à des ordres précis à donner à un périphérique. Pour cela le concepteur de programme définit plusieurs niveaux d'abstraction de ses informations, un agent servant à faire passer l'information d'un niveau à l'autre. Par exemple si nous voulons donner à l'utilisateur une information de guidage : « vers le haut ». Le système envoie un message signifiant « haut » sur le bus. Des agents prennent le relais pour transformer ces informations et les afficher sur différents périphériques. Un agent va créer une icône tactile et l'afficher sur la VTPlayer. Un autre va créer un PICOB guidé et le jouer sur le PHANToM. Un autre peut très bien dessiner une flèche sur l'écran.

Un des problèmes lors du développement d'application multimodales utilisant plusieurs périphériques est que chaque périphérique utilise son propre pilote ou sa propre API de développement. L'investissement en terme de développement devient très vite important. L'API MICOLE constitue une interface avec plusieurs autres API et pilotes permettant au développeur d'applications de ne pas avoir à se soucier de la communication avec le périphérique. Des agents d'entrée et sortie sont intégrés à l'API et permettent de créer des contraintes avec un PHANToM (sur des points, des lignes, des disques, des plans, et des sphères, des PICOB semi-guidés), de jouer des PICOB guidés, de jouer des icônes statiques et dynamiques sur la VTPlayer, de jouer du son 2D et 3D, d'utiliser des périphériques d'entrée comme un clavier, une souris ou encore un Magellan. Il est possible de gérer d'autres périphériques. Pour cela il faut développer les agents d'entrée et de sortie qui se chargeront de faire le lien entre l'API MICOLE et le pilote ou l'API de développement de ce périphérique. Un framework de développement a été défini afin d'uniformiser les méthodes de communication avec les périphériques.

Dans un premier temps nous avons utilisé le bus Ivy [2] pour les communications entre agents. Cependant la pratique nous a montré qu'Ivy est optimisé pour des communications entre plusieurs applications et non entre plusieurs agents au sein d'une même application. Nous avons donc ré-implémenté notre propre système de bus en se basant sur Ivy.

### LES CIRCUITS ÉLECTRIQUES HAPTQUES

L'application développée à Metz avec cette architecture permet d'explorer des schémas de circuits électriques. L'exploration peut se faire de deux manières, chacune offrant des possibilités de rendu différentes. La première utilise le PHANToM : le rendu est haptique et visuel, et la seconde utilise la souris et n'offre qu'un rendu visuel.

En ce qui concerne le retour haptique, tout d'abord le

stylet est contraint sur le schéma afin de l'empêcher de sortir et ainsi éviter de le perdre dans la scène.

Ensuite les composants sur le circuit sont représentés de deux manières. La première est une séquence de PICOB semi-guidés. Une bosse représente une lampe, deux bosses représentent une pile et trois bosses représentent une résistance. La deuxième manière utilise les icônes tactiles de la VTPlayer (figure 2). Une croix représente une lampe, un petit suivi d'un grand trait représentent une pile, un grand carré creux représente une résistance, une ligne verticale épaisse représente un fil, et un petit carré représente une jonction.

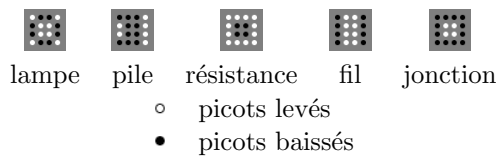


Figure 2 : Icônes représentant les composants

Les utilisateurs peuvent aussi demander les directions possibles en cliquant sur le bouton du PHANToM. Cette information est potentiellement utile aux jonctions. L'information d'orientation est elle aussi donnée de deux manières différentes simultanément. La première utilise les PICOB guidés sur le PHANToM : le système joue une séquence d'impulsions dans les directions où l'utilisateur peut se déplacer. La seconde consiste à afficher en séquence des icônes directionnelles (figure 3) représentant les directions possibles sur la VTPlayer.

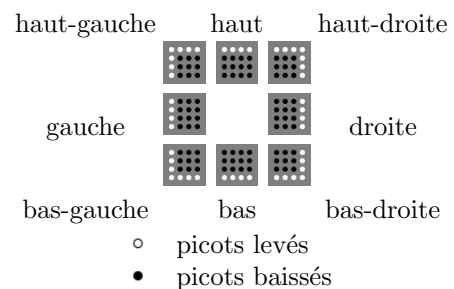
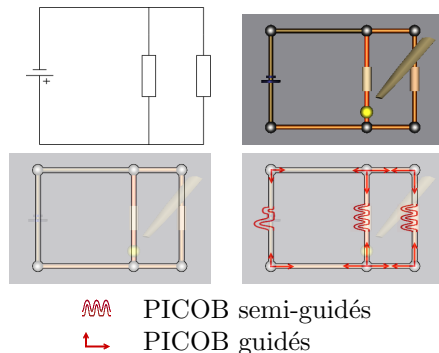


Figure 3 : Icônes représentant 8 directions

Le rendu visuel repose principalement sur une vue du schéma électrique. Les composants sont représentés de manière similaire aux symboles courants : un petit et un grand trait pour la pile, un rectangle pour la résistance et un cercle avec une croix pour la lampe. L'objet actuellement parcouru est en surbrillance et les objets déjà parcourus sont plus foncés. Un curseur jaune représente la position du PHANToM et un curseur rouge représente la position de la souris.

La figure 4 montre un exemple de schéma électrique, sa représentation dans le logiciel, les contraintes haptiques, et enfin les PICOB guidés pour les indications

de direction et semi-guidés pour les composants électriques.



**Figure 4 :** Du circuit visuel au circuit haptique

Il est possible d'ajouter d'autres types de rendu (sonore par exemple) en ajoutant des agents gérant ce rendu. La flexibilité de l'API fait en sorte que le reste du programme ne nécessite que des changements mineurs et très localisés.

Un système de guidage utilisant les fonctionnalités offertes par l'API a été mis en place pour permettre de montrer des endroits grâce au retour de force. L'idée est de tracter l'utilisateur le long du circuit, pour lui faire parcourir par exemple des parties du circuit qu'il n'a pas exploré. Deux modes ont été développés. Le premier consiste à enregistrer un parcours fait à l'aide du PHANToM et de le rejouer. Dans ce cas l'enseignant enregistre un parcours typique et l'élève le rejoue autant de fois qu'il le désire. Le second mode consiste à guider le PHANToM à l'aide de la souris. Cette fois-ci le guidage est interactif. Le parcours n'est pas fixé, mais il ne peut pas être rejoué.

L'application utilise les agents fournis par l'API qui gèrent les entrées du PHANToM et de la souris afin de maintenir une projection de ces deux périphériques sur le circuit. Nous avons créé ensuite des agents pour chaque périphérique de sortie afin de transformer des informations comme les composants ou les directions en forces et en motifs tactiles en utilisant les techniques d'interactions ci-dessus. Le rendu est fait par les agents de sortie contenus dans l'API.

## CONCLUSION

Nous avons montré une nouvelle API de développement d'applications multimodales et multi-utilisateurs pour enfants déficients visuels. L'architecture logicielle est basée sur un système multi-agents communiquant grâce à un bus logiciel. L'API permet de piloter des périphériques haptiques comme le PHANToM ou la souris VTPlayer, de jouer des sons 3D, et de traiter des informations venant de périphériques d'entrée comme une souris, un clavier ou encore un Magellan. L'utilisation de l'API permet de s'affranchir des soucis de communication avec plusieurs bibliothèques de dé-

veloppement spécifiques à chaque périphérique, cette tâche étant reléguée au concepteur de l'API.

Plusieurs techniques d'interactions sont implémentées dans cette API, parmi lesquelles les PICOB et les icônes tactiles, ainsi que des techniques de guidage.

L'exemple exposé est un logiciel d'exploration de schémas de circuits électriques utilisant un PHANToM, une souris et une VTPlayer. Elle utilise les techniques d'interaction précédemment citées pour afficher le composant actuellement parcouru, ainsi que pour donner des informations de guidage afin d'aider l'utilisateur à se déplacer dans le circuit.

## REMERCIEMENTS

Ce travail est financé par le projet européen MICOLE (IST-2003-511592), la CA2M (Communauté d'Agglomération de Metz Métropole) par le projet PICOB et la région Lorraine.

## BIBLIOGRAPHIE

1. *European project MICOLE : Multimodal Collaboration Environment for Inclusion of Visually Impaired Children - IST-2003-511592*. <http://micole.cs.uta.fi>.
2. Buisson, M., Bustico, A., Chatty, S., Colin, F.-R., Jestin, Y., Maury, S., Mertz, C., and Truillet, P. Ivy : Un bus logiciel au service du développement de prototypes de systèmes interactifs. In *IHM 2002 : Proceedings of the 14th French-speaking conference of human-computer interaction*, Poitiers, France, Nov. 2002. ACM Press.
3. Horstmann, M., Hagen, C., King, A., Dijkstra, S., Crombie, D., Evans, G., Ioannidis, G. T., Blenkhorn, P., Herzog, O., and Schlieder, C. TeDUB : Automatic interpretation and presentation of technical diagrams for blind people. In *Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment - CVHI'2004, EURO-ASSIST-VHI-2 : Accessibility, Mobility and Social Integration*, Granada, Spain, 2004.
4. <http://www.reachin.se>. Reachin API 4, 2005.
5. Pietrzak, T., Martin, B., and Pecci, I. Affichage d'informations par des impulsions haptiques. In *IHM 2005 : Proceedings of the 17th French-speaking conference of human-computer interaction*, pages 223–226, Toulouse, France, Sept. 2005. ACM Press.
6. Pietrzak, T., Martin, B., and Pecci, I. Information display by dragged haptic bumps. In *Enactive /05*, Genova, Italy, 2005. CD-ROM proceedings.
7. Pietrzak, T., Pecci, I., and Martin, B. Static and dynamic tactile directional cues experiments with vtplayer mouse. In *Proceeding of the 6th International Conference EuroHaptics 2006*, pages 63–68, Paris, France, July 2006.