



Architecture modulaire pour le pilotage des microsystemes de production. Application à la micromanipulation télé-opérée.

David Hériban, Eric Descourvières, Benoît Ballarin, Dominique Gendreau,
Philippe Lutz

► To cite this version:

David Hériban, Eric Descourvières, Benoît Ballarin, Dominique Gendreau, Philippe Lutz. Architecture modulaire pour le pilotage des microsystemes de production. Application à la micromanipulation télé-opérée.. 5ème Conférence Internationale sur l'Automatisation Industrielle, AIAI'2007., Jun 2007, Montréal, Canada. sur CD ROM - 4 p. <hal-00163841>

HAL Id: hal-00163841

<https://hal.science/hal-00163841v1>

Submitted on 18 Jul 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

ARCHITECTURE MODULAIRE POUR LE PILOTAGE DES MICROSYSTEMES DE PRODUCTION. APPLICATION A LA MICROMANIPULATION TELE-OPEREE.

David Hériban, Eric Descourvières, Benoît Ballarin, Dominique Gendreau, Philippe Lutz

Laboratoire d'Automatique de Besançon - UFC-ENSMM-CNRS UMR 6596

24, rue Alain Savary - 25000 BESANCON-France

dheriban@ens2m.fr eric.descourvieres@insa-strasbourg.fr benoit.ballarin@ens2m.fr dgendreau@ens2m.fr plutz@ens2m.fr

Résumé : La fabrication automatisée de microproduits requiert lors du prototypage une assistance technique à l'opérateur compte tenu de l'inaccessibilité directe que ce soit pour agir ou pour observer et mesurer. L'instrumentation des plateformes devient incontournable et l'interopérabilité des modules requiert une architecture adaptée du système d'information. Nous utilisons une modélisation du système d'information sous UML pour caractériser les spécificités de ces systèmes et nous avons développé une application en Programmation Orientée Objet. L'interface modulaire obtenue, nommée AP2M, a été testée puis validée sur plusieurs stations de télé-opération. Cette AP2M répond aux contraintes logicielles et matérielles des équipements périphériques de production.

Mots clés : pilotage modulaire, microsystemes d'assemblage, système d'information, Unified Modelling Language (UML), Programmation Orientée Objet (OOD : Object Oriented Design), Interface Homme-Machine (IHM)

1. INTRODUCTION

La fabrication de micro-objets, dès lors appelés microproduits dans notre contexte d'automatisation, nécessite la mise en œuvre de microsystemes de production spécifiques. La manipulation des composants de taille micrométrique soulève bon nombre de problèmes : précision de positionnement, comportement indéterministe des microcomposants, inaccessibilité directe (visuelle et tactile) du processus, etc. Ainsi, de nombreux modules d'actionnement pilotés de façon souvent très différente sont assemblés pour répondre à ces problèmes. La flexibilité nécessaire pour le micromonde doit être répercutée au système de pilotage des cellules de micromanipulation.

En l'absence actuelle de méthodes automatiques aux échelles micrométriques, les cellules sont temporairement téléopérées lors de l'expérimentation de gammes de production. Puis le système de pilotage automatique prendra le relais. L'architecture de pilotage doit alors permettre à l'opérateur d'influer directement sur le micromonde pour développer ses connaissances sur cet univers inaccessible directement. A ce titre, les travaux de recherche se situent au niveau Contrôle-Commande dans la pyramide du CIM (Computer Integrated Manufacturing). Ils s'appuient principalement sur une modélisation UML de diagrammes de cas d'utilisation et de diagrammes de classes.

Notre objectif, présenté ici, a donc été de produire une architecture de pilotage modulaire et flexible. Elle nous permet de piloter les stations de micromanipulation quelque soient les modules utilisés (micropinces, tables de micropositionnement motorisées, bras robots ...). Elle se conforme en outre à un certain standard lui permettant de partager les modules matériels et leurs ressources informatiques entre les stations. Ainsi, utilisateurs et concepteurs de commande peuvent mutualiser leurs développements grâce au standard établi. A plus long terme, l'objectif est de concevoir une « plateforme de

production de microproduits » automatisée, flexible, réorganisable et reconfigurable, pouvant accueillir des cellules automatisables constituées de modules élémentaires de différente nature (actionneurs, capteurs, outils, etc.) [1]. Nous présentons une architecture de pilotage répondant à ces objectifs. Nous définissons et caractérisons les modules et leurs entrées/sorties ainsi que leur organisation au sein d'une chaîne de commande.

La conception de cette architecture nous a permis de développer une application modulaire répondant aux contraintes logicielles et matérielles des équipements de production et en particulier les cellules de micromanipulation. Le Laboratoire d'Automatique de Besançon, spécialisé dans ce domaine, a permis le déploiement de l'application sur des cellules de micromanipulation de différents types. Une première version du logiciel de pilotage, baptisé AP2M (Application pour le Pilotage de MicroManipulations), a été développée et implémentée afin d'offrir une application de commande télé-opérée de stations modulaires pour la micromanipulation. Regroupant de multiples commandes pour des actionneurs de nature différente, cette version acquiert également les mesures de positionnement en provenance du micromonde et offre ainsi de nombreuses informations à l'opérateur. L'interface Homme-machine ainsi développée dispose de nombreux paramétrages et possibilités pour que l'opérateur réalise facilement et avec ergonomie ses essais de manipulation dans le micromonde. La suite de l'article s'attache à décrire plus précisément l'AP2M, application validant l'architecture modulaire.

2. SPECIFICITE DES MICROSYSTEMES D'ASSEMBLAGE

On appelle « microsysteme d'assemblage » tout système employant des techniques et des technologies spécifiques à l'assemblage de microproduits et dont le volume n'excède pas un mètre cube. Un « microproduit » est constitué par

l'assemblage de plusieurs microcomposants, dont le volume n'excède pas un millimètre cube.

Trois différences essentielles caractérisent les systèmes de production micrométriques par rapport aux systèmes mesométriques (relatifs à l'échelle humaine) : l'indéterminisme du processus réalisé par le système, la difficulté d'intervention d'un opérateur dans le micromonde et l'absence de savoir-faire « métier » dans la conception de systèmes de production.

Ces trois éléments nous permettent de justifier la mise en place d'une architecture modulaire et la réalisation d'une interface de programmation adaptée à la micro-manipulation télé-opérée.

2.1. Indéterminisme du processus mis en œuvre dans les microsystèmes de production

Tout système de production requiert un certain nombre de paramètres. Les paramètres et leur influence sont clairement identifiés et bien modélisés dans les systèmes mesométriques depuis de nombreuses années. On a ainsi la possibilité d'appliquer des stratégies pour négliger, réduire, maîtriser leur influence, et ainsi assurer un fonctionnement garantissant un niveau de qualité, de délai et de coût satisfaisant. Les microsystèmes de production dérogent à cette règle au sens où le processus comporte le plus souvent une part importante d'indéterminisme. En effet, la conjugaison de nombreux phénomènes physiques du micromonde, bien qu'il soit développé des modèles les représentant [2] fait qu'il est quasiment impossible de prévoir de façon satisfaisante le comportement du processus. Cet univers est ultra sensible à tous les paramètres d'environnement dont on ne peut avoir qu'une modélisation très imprécise.

2.2. Niveau d'intervention d'un opérateur dans les microsystèmes de production

En l'absence de dangerosité, il est généralement possible de faire intervenir un opérateur dans un système dimensionné à son échelle ; les systèmes micrométriques sont quant à eux inaccessibles à celui-ci en raison des capacités sensorielles et des précisions motrices de l'être humain. Les limites visuelles et tactiles conjuguées avec les mouvements parasites d'une amplitude de l'ordre de la centaine de micromètres rendent impossible l'intervention manuelle dans les microsystèmes. La perception des phénomènes micrométriques ainsi que les interventions sur le microsystème passent par conséquent obligatoirement par des interfaces adaptées et sécurisées pour ce système.

2.3. Conception des microsystèmes de production

La conception d'un atelier de production passe par le choix d'équipements réalisés par des constructeurs. Chaque procédé de fabrication nécessite un savoir-faire « métier » qui permet l'étude et la réalisation de machines adaptées et performantes. A l'opposé, pour des applications micrométriques, l'offre dans la fourniture de microcapteurs

ou de micro-actionneurs est sommaire et surtout n'est pas facilement intégrable dans une cellule de production de petite taille. En tout état de cause, il n'existe aucune cellule disponible sur catalogue. Le savoir-faire d'intégrateur d'équipements de ce genre se trouve disséminé dans quelques laboratoires de recherche ou auprès de quelques industriels qui développent des applications très spécifiques. Ainsi, il convient de considérer des systèmes plus autonomes, qui intègrent des solutions matérielles adaptées, des modèles qui définissent leurs comportements ainsi que des commandes élaborées et spécifiques.

3. ARCHITECTURE DE DONNEES

La couche contrôle-commande (voir fig. 1) de la pyramide du CIM [3] sert de cadre à notre étude architecturale. Les contraintes du micromonde nous amènent à définir une architecture de données permettant le pilotage de stations de micromanipulation. Cette architecture présente une structure modulaire où tous les modules sont caractérisés et leurs interactions définies. La représentation est effectuée en UML qui associe un cadre standard et une lisibilité reconnue.

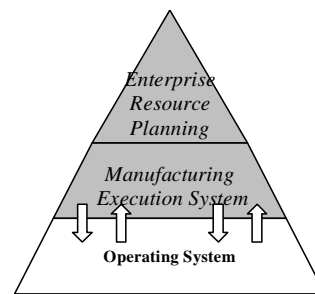


Fig. 1 : situation dans la pyramide CIM

3.1 Environnement modulaire

De par ses caractéristiques totalement différentes du monde dans lequel nous évoluons, le micromonde ne peut être accédé par l'homme qu'avec des actionneurs et des capteurs spécialement adaptés, extrêmement hétéroclites dans leur façon de fonctionner et de communiquer. La diversité des commandes nécessaires dans une station de micromanipulation est donc très importante et nécessite une architecture de données *ad hoc*.

Entre l'homme et le micromonde, et plus particulièrement entre l'homme et les outils d'interactions avec le micro, se situe l'Interface Homme-Machine (IHM). Cette interface regroupe divers éléments communiquant entre eux comme les périphériques (joystick, clavier, écran), les commandes de haut niveau (script, modèles géométriques), les commandes de bas niveau (commande d'actionneurs, gestion capteurs).

Finalement cette diversité des éléments interagissant selon leur nature, leurs fonctions et leurs buts, nous incite à encapsuler ces éléments en modules génériques afin de faciliter leur conception, leur développement et leur gestion [4].

3.2 Contrainte structurelle principale

La modularité est un élément essentiel de l'architecture de pilotage. En effet, la station de micromanipulation est aussi physiquement constituée de modules (joystick, bras robot, micropince, vidéo-microscopes, support de manipulation). L'objectif principal de l'architecture que nous présentons est de permettre le remplacement et la reconfiguration de ces modules physiques, et donc de leurs pendants logiciels, c'est-à-dire les commandes de bas niveaux et les modèles géométriques.

Ce remplacement doit induire le minimum de modifications dans l'IHM et aucune modification dans l'architecture de données, même si l'élément remplacé est totalement différent de son remplaçant. Il est envisageable par exemple de substituer une buse d'aspiration à une micropince électrostatique pour la saisie des objets. Ainsi le remplacement de ces deux actionneurs et de leur commande totalement disparate doit impacter l'IHM le moins possible.

3.3 Définition des modules structurels de l'IHM

Les modules structurels de l'IHM sont les modules génériques de l'architecture de données. Lors du pilotage d'une station de micromanipulation par un opérateur, trois grandes couches sont traversées par les informations de pilotage. L'opérateur envoie ses directives via les périphériques d'entrées et reçoit en retour des informations. Puis les périphériques envoient ces données aux commandes de haut niveau qui vont les traiter et les traduire. Ces informations vont finalement être envoyées aux commandes de bas niveau qui communiquent directement avec les modules physiques d'interaction avec le micromonde.

De ces trois couches, nous pouvons extraire trois modules génériques qui sont dérivés en modules spécifiques suivant les tâches à réaliser. Ainsi les périphériques sont représentés par un module *Peripheral* pouvant être dérivé en module spécifique *Keyboard* ou *Joystick*. Concernant les différentes commandes, nous avons choisi de nous orienter vers la commande d'axes de déplacement en translation et en rotation. Les entrées et sorties de ces modules de commande sont donc des consignes en position (μm et rad) et en vitesse ($\mu\text{m/s}$ et rad/s). Les commandes de haut niveau ont été encapsulées dans un module générique *AxisControl* et les commandes de bas niveau dans un module générique *Mover*. Les éléments peuplant nos trois couches sont donc définis mais leurs interactions restent à étudier.

Le nœud du problème se situe dans les échanges d'informations entre les commandes de haut et bas niveaux. La cohabitation possible de commande télé-opérée, semi-automatique et automatique va contraindre le système en terme de priorité d'accès à la commande de bas niveau, pour piloter les actionneurs de la station. De plus les fréquences d'envoi de données ne sont pas les mêmes pour toutes les commandes. La connexion directe de ces éléments étant compromise, il apparaît nécessaire de créer

un module tampon faisant le lien entre les deux couches de commande, gérant les priorités lors des chevauchements de commande et garantissant la flexibilité du système en cas de modification de la station. Ce module tampon, réunissant les consignes en position et vitesse, ainsi que l'état actuel de l'axe physique (donné par les capteurs) a été appelé « Axe Virtuel », *Virtual Axis* dans l'architecture.



Fig. 2 : représentation UML de l'architecture de pilotage

3.4 Construction de l'architecture

Au cours de la définition des modules, différents choix ont été fait pour encourager la modularité et la flexibilité du système. Le premier choix est de rendre l'axe virtuel « passif » dans la communication des informations consigne/état. Seuls les modules dérivés de la classe générique *AxisAccess*, dont dérivent les modules de commande de haut et bas niveau, *AxisControl* et *Mover*, ont un rôle actif dans l'émission et la récupération des informations. Ce choix permet ainsi de retirer un module de commande sans que cela affecte le comportement des autres modules.

L'axe virtuel n'a pour activité que la gestion des priorités d'accès à ses données via une classe *AxisInput*, matérialisant le lien entre un axe virtuel et sa commande ou son capteur. Cette passivité lui permet d'être simple, générique et de gêner le moins possible la communication et l'activité des modules l'entourant.

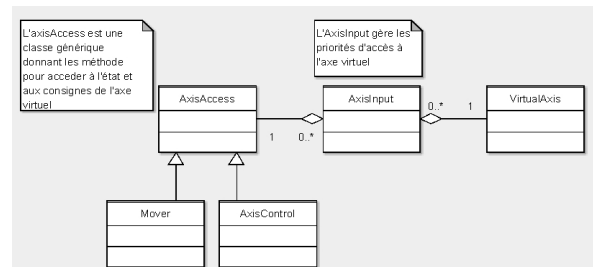


Fig.3 : Interactions axes virtuels/modules gestion des accès

4. APPLICATION DEVELOPPEE

L'architecture développée pour piloter la station de micromanipulation modulaires a fait l'objet d'une représentation UML. Les diagrammes ainsi obtenus nous ont permis de construire avec beaucoup de facilité les classes C++ correspondantes intégrées dans une application développée en Programmation Orientée Objet sous Borland C++ Builder 6.0. Ce projet a été nommé AP2M, « Application pour le Pilotage de MicroManipulation ».

4.1 Station de micromanipulation.

La station présentée figure 4 a été la première à accueillir l'AP2M. Elle est composée principalement d'une

micropince piézo-électrique portée par un bras mobile verticalement. Le support de micromanipulation est motorisé sur trois translations et une rotation. Il est entouré de deux vidéo-microscopes (vertical et horizontal) qui offrent une vue détaillée de la scène de manipulation. Disposant donc de cinq degrés de liberté pour les translation/rotations et quatre degrés pour la micropince, le pilotage de la station nécessite des outils spécifiques pour être harmonieux, aisé et précis pour l'utilisateur humain.

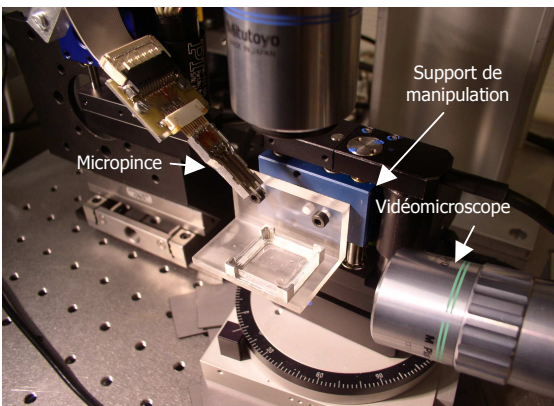


Fig. 4 : vue d'ensemble d'une station et sa zone de travail

4.2 IHM Développée

La conception de l'AP2M a fait l'objet d'une attention particulière concernant la modularité, la facilité d'utilisation et la maximisation des possibilités offertes à l'utilisateur.

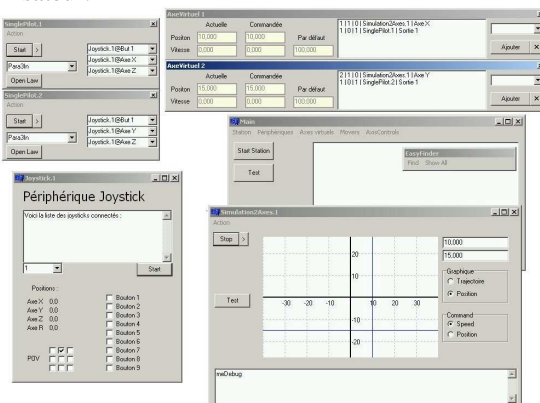


Fig. 5 : interface graphique d'utilisation

La fenêtre principale permet de configurer la station, en créant les modules spécifiques issus de l'architecture (fig. 5). Puis chaque module, représenté par une fenêtre, peut être configuré (liaison avec les autres modules, caractéristiques propres, etc.). Finalement, toute cette

station virtuelle peut être sauvegardée et modifiée à volonté suivant les besoins.

Les modules (périphériques et commandes) sont encapsulés au moment de leur conception dans des bibliothèques dynamiques Windows (dll). Ainsi, l'ajout de modules dans l'application se fait en chargeant la dll correspondante. En plus d'une optimisation de la mémoire utilisée par l'application, ce système offre l'avantage d'externaliser la conception de modules de la conception de l'application principale.

4.3 Points forts de l'architecture

L'architecture développée en Programmation Orientée Objet offre de nombreux avantages. Premièrement, l'organisation architecturale des données présentées en deuxième partie d'article offre un cadre standard pour le développement des différents modules de l'AP2M. Deuxièmement, la flexibilité du système modulaire permet d'utiliser l'application sur de nombreuses stations de micromanipulation. Enfin, comme plusieurs utilisateurs utiliseront le même logiciel, avec des modules souvent différents mais quelquefois communs, ils pourront mutualiser leurs développements et ainsi faire profiter tous les utilisateurs de leurs travaux. Au sein du Laboratoire d'Automatique de Besançon, deux stations sont déjà pilotées par l'AP2M et trois autres sont en cours de remplacement de leurs logiciels spécifiques par notre application.

5. CONCLUSION

Ces travaux ont permis de développer une architecture de pilotage standard, assurant la flexibilité maximale nécessaire au micromonde. Représentée en UML, elle offre au pilotage un cadre robuste et standard et nous a permis de développer une application, l'AP2M, offrant de nombreuses possibilités. Enfin, les travaux effectués sont intégralement transposables pour toute cellule de micromanipulation et les développements générés par les utilisateurs sont directement intégrables et disponibles pour la communauté.

REFERENCES

- [1] Descourvières E., Gendreau D., Lutz P., *Data representation for the control of full-automated microfactories*, IWMF 2006, Besancon, France.
- [2] Gauthier M., Régnier S., Rougeot P., Chaillet N., *Forces analysis for micromanipulation in dry and liquid medium*, Journal of Micromechatronics (2006), vol. 3, n°3-4, pp. 389-413.
- [3] Publisher: Springer; 2 edition, *CIMOSA: Open System Architecture for CIM (Paperback)*, book ISBN: 3540562567, Jul 9 1993.
- [4] Gendreau D., Gauthier M., Hériban D., Lutz P., *Contribution à la mise en place d'une architecture modulaire pour la conception des microsystèmes de production*, 7e CIGI, 5-8 juin 2007, Trois-Rivières, Québec (CANADA)