



HAL
open science

Anaxagore, un exemple d'ingénierie dirigée par les modèles pour la supervision industrielle

Olga Goubali, Pascal Berruet, Alain Bignon, Patrick Girard, Laurent Guittet

► **To cite this version:**

Olga Goubali, Pascal Berruet, Alain Bignon, Patrick Girard, Laurent Guittet. Anaxagore, un exemple d'ingénierie dirigée par les modèles pour la supervision industrielle. Proc. Ergonomie et Informatique Avancée (ERGO'IA 2014), Oct 2014, Bidart, France. hal-04205794

HAL Id: hal-04205794

<https://hal.science/hal-04205794>

Submitted on 13 Sep 2023

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.

Anaxagore, un exemple d'ingénierie dirigée par les modèles pour la supervision industrielle.

Olga GOUBALI

LIAS/ENSMA & Segula
1 avenue Clément Ader, 86961
Chasseneuil, France
olga.goubali@ensma.fr

Alain Bignon

SEGULA Technologies
BP 50256, 56602 LANESTER
Cedex, France
alain.bignon@segula.fr

Pascal Berruet

Université de Bretagne Sud
Lab-STICC
BP 92116
56321, Lorient Cedex, France
pascal.berruet@univ-ubs.fr

**Patrick Girard,
Laurent Guittet**

LIAS/ENSMA
1 avenue Clément Ader, 86961
Chasseneuil, France
patrick.girard@ensma.fr
laurent.guittet@ensma.fr

RÉSUMÉ

Depuis plusieurs années, l'IDM (Ingénierie Dirigée par les Modèles) occupe une place de plus en plus importante dans l'Interaction Homme-Machine (IHM). Elle est particulièrement bien adaptée à la conception de logiciels de supervision pour les systèmes sociotechniques reconfigurables. Dans cet article, nous décrivons *Anaxagore*, un outil basé sur une démarche IDM, qui permet la génération automatique d'un logiciel interactif de supervision à partir de descriptions synoptiques métier.

ABSTRACT

For several years, MDE (Model Driven Engineering) has been increasing in Human Computer Interaction domain. It is particularly suitable for designing supervision software for sociotechnical and reconfigurable systems. In this article, we describe *Anaxagore*, an IDM based tool, which allows complete generation of interactive supervision software from business oriented synoptic descriptions.

Author keywords

MDE; HCI; industrial supervision

ACM Classification Keywords

H.5.m. Information interfaces and presentations (e.g., HCI): Interactive systems and tools.

INTRODUCTION

Dès le début du XXI^{ème} siècle, l'ingénierie logicielle s'est résolument orientée vers l'Ingénierie Dirigée par les Modèles (IDM) qui permet aujourd'hui d'exploiter les modèles de conception comme une matière de production grâce à des techniques de génération automatique.

L'utilisation de l'IDM dans le domaine de l'IHM est un sujet en plein essor [10, 23]. Ses concepts s'appliquent aussi aux interfaces de supervision industrielle qui consistent à contrôler et surveiller l'exécution d'une tâche, à visualiser en temps réel l'état et l'évolution d'une installation automatisée, à intervenir en cas de dysfonctionnement.

La complexité des systèmes de supervision conduit les concepteurs d'IHM à exprimer leurs connaissances par des modèles. Les approches basées sur les modèles visant à automatiser et à formaliser le processus de conception d'IHM sont rares. Nous proposons dans cet article une solution complète au problème de la génération automatique d'une interface de supervision et d'un programme de commande pour un système sociotechnique à partir d'un schéma structuro-fonctionnel normalisé appelé schéma synoptique.

ETAT DE L'ART

L'Object Management Group propose une première formalisation des concepts de l'IDM au travers de sa recommandation MDA¹ [20]. Cette approche vise à élever le niveau d'abstraction des activités de programmation en plaçant les modèles au tout début du processus de développement logiciel. Elle introduit notamment les notions de *modèles*, *méta-modèles* et *transformations* ainsi que les relations de *représentation*, *conformité* et « *basée sur* ». Ces termes sont détaillés dans [4].

Les recherches dans le domaine de l'IHM se basent de plus en plus sur un ensemble de modèles et de transformations pour la conception des IHM [19, 9]. L'application du paradigme de modèle au développement des IHM permet de réduire le coût de développement tout en restant indépendant des langages d'implémentation, d'obtenir des IHM de meilleure qualité, de détecter plus rapidement les erreurs de conception et d'intégrer les connaissances de l'expert métier dans les modèles exécutables [23]. CAMELEON² [7] est ainsi devenu un modèle de référence d'architecture pour la production d'IHM plastiques en utilisant les techniques de l'IDM. Il recommande quatre modèles fondamentaux (Figure 1) : le modèle de tâches et le modèle des concepts métiers, l'IHM abstraite, l'IHM concrète et l'IHM finale. Ces termes sont plus détaillés dans [8].

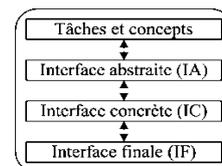


Figure 1. Modèles fondamentaux de CAMELEON [7]

¹ Model Driven Architecture

² Context Aware Modelling for Enabling and Leveraging Effective interactiON

Les systèmes interactifs de supervision présentent une vue du système qui fournit à l'opérateur une représentation graphique proche de la représentation physique et concrète du système supervisé. À travers ce type d'IHM, l'utilisateur doit avoir la possibilité d'interagir avec le système physique concret [11]. Un système de supervision doit donc permettre de piloter et de surveiller ce qui se passe sur le procédé supervisé. L'Interaction Homme-Machine dont les éléments centraux sont l'utilisateur, la tâche et le système, joue un rôle majeur dans tout système de supervision [17].

L'approche IDM est essentiellement utilisée pour la génération des applications à partir des modèles [6]. Cependant, la modélisation des tâches et la génération automatique d'IHM à base de modèles ne sont pas généralisées à des cas d'IHM complexes [13, 12].

Afin d'éviter les erreurs humaines sur des tâches de haut niveau telles que la surveillance globale, en particulier dans le domaine du contrôle de procédés industriels, la méthode ERGO CONCEPTOR [18] a été proposée. Elle permet, à partir d'une description détaillée du procédé à surveiller, de générer les spécifications qui serviront de base à la conception de l'IHM de supervision. Lors de la conception de l'IHM, le concepteur peut choisir le mode de représentation selon les vues graphiques à concevoir. Malgré ses nombreux avantages pour le monde industriel, ERGO CONCEPTOR présente certaines limites. La première phase de la conception qui est la description du procédé par page-écran textuelle est fastidieuse, source d'interrogation et d'oublis [14]. En effet, au cours de cette phase le procédé est décrit suivant plusieurs axes dont l'axe fonctionnel. Ce dernier permet de décrire le fonctionnement des sous-systèmes du procédé et les interactions entre eux en termes de variables. Cette phase peut s'avérer difficile lorsqu'il s'agit d'un procédé complexe [14].

CONCEPTION DES SYSTEMES DE SUPERVISION

La supervision dans le domaine industriel est une technique de suivi et de pilotage de procédés automatisés [26]. Les systèmes de supervision industrielle offrent ainsi un ensemble de moyens utilisés pour gérer un procédé aussi bien en situation normale qu'anormale. Une interface de supervision est le plus souvent présentée sous forme synoptique ; elle permet d'une part, d'acquérir les données relatives aux mesures, aux alarmes et au retour d'état de fonctionnement du système réel, et d'autre part d'accéder aux paramètres de commande des processus généralement confiés à des automates programmables [1]. La présentation des IHM de supervision sous une forme graphique animée permet de faciliter la tâche à l'utilisateur pendant le processus de supervision. Plusieurs logiciels de conception d'application dénommés génériquement SCADA¹ (tels InTouch² ou

Panorama E2³) ont été conçus pour faciliter la conception des systèmes de supervision. Ces outils « temps-réel » permettent de visualiser les états physiques ou fonctionnels des équipements et de prendre en charge les fonctions avancées d'un procédé.

Pour ce projet, nous avons utilisé Panorama E2 qui est un logiciel propriétaire commercialisé par la société CODRA ; il est par exemple utilisé pour superviser le LMJ⁴ du CEA. Avec ce logiciel, l'utilisateur peut créer une application de supervision à partir d'objets de base qu'il dispose au sein d'une arborescence construite pour modéliser le procédé à superviser.

Panorama E2 utilise une architecture basée sur un découpage en trois niveaux (Figure 2).

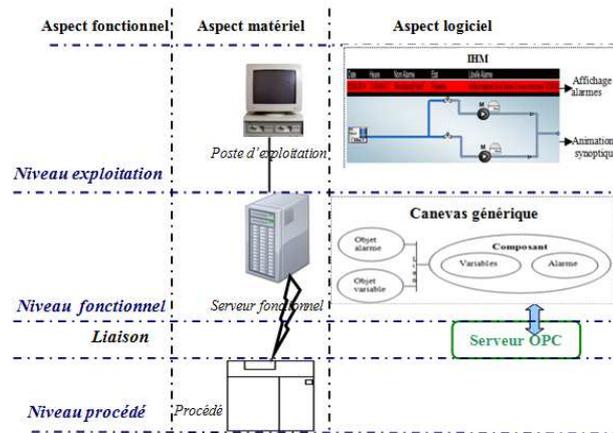


Figure 2 Schéma d'architecture de Panorama E2

Le *niveau exploitation* est constitué d'outils destinés à l'exploitation de l'installation. Son fonctionnement est lié aux activités de l'opérateur. Il représente l'IHM de l'application. Le *niveau fonctionnel* est le noyau de la supervision. Il assure les fonctions de communication avec les systèmes d'acquisition de données, avec lesquels la supervision communique pour connaître l'état du procédé. Il prend également en charge les traitements étroitement associés au fonctionnement opérationnel du procédé (gestion des alarmes, archivage des données, automatisation, ...). Enfin, le *niveau procédé* qui se situe hors de Panorama E2 correspond aux systèmes d'acquisition et de contrôle de l'installation supervisée. Les échanges entre Panorama E2 et ce niveau peuvent être mis en œuvre avec un serveur de données de type OPC⁵. Ce niveau assure plusieurs fonctions comme la gestion d'entrées/sorties, l'automatisation, la régulation, et peut être dévolu à un logiciel spécifique, généralement un automate (dans notre cas, nous utilisons le logiciel Straton³).

² www.wonderware.fr

³ fr.codra.net/

⁴ Laser Mega Joule <http://www-lmj.cea.fr/>

⁵ www.copalp.com

¹ Supervisory Control and Data Acquisition

PROBLEMATIQUE DE LA CONCEPTION DES SYSTEMES DE SUPERVISION

La supervision d'un processus complexe est un domaine de croisement des disciplines telles que la mécanique, l'automatique, l'informatique, l'ergonomie, la psychologie cognitive, et la sociologie [22]. Si des outils techniques ou méthodologiques existent pour faciliter la conception de tels systèmes, ils souffrent cependant de certaines limites.

Un état de l'art industriel réalisé dans le cadre de nos travaux [5] a démontré l'usage systématique, et au plus tôt dans le projet, d'un schéma technique unique qui suivra le système tout au long de son cycle de vie. Ce schéma, que nous appelons synoptique, constitue le modèle de conception de base partagé par les experts en charge de la conception.

Nos travaux se situent dans le domaine maritime, les systèmes supervisés étant des systèmes autonomes. Plus précisément, nous considérons le système de distribution, stockage et production d'eau douce sur un navire. Ce type de système présente la particularité de posséder une sémantique particulière liée à l'écoulement de fluides. Le système présenté sur la Figure 3 permet de transférer de l'eau d'une soute (St1) vers une soute (St2) via l'un ou l'autre des groupes hydrophores (H1, H2 et H3). Un groupe hydrophore est un dispositif qui permet de maintenir la pression dans le réseau. Chaque groupe est isolé de la soute St1 par une vanne motorisée à deux voies (par exemple V2VM03, V2VM05 et V2VM07). L'eau transférée est envoyée vers la soute St2 via l'une des vannes motorisées à trois voies (V3VM01, V3VM02 et V3VM03), le module de chloration (TRCH) et la vanne V2VM02.

Afin de superviser ce système, des spécifications techniques doivent être ajoutées. Ces spécifications concernent essentiellement le chemin que doit prendre

l'eau pour aller de St1 vers St2 et conduisent à sept configurations possibles. La première configuration consiste à faire passer l'eau par V2VM03 et H1. La deuxième consiste à faire circuler l'eau par V2VM05 et H2. La troisième consiste à faire circuler l'eau par V2VM07 et H3. Les configurations restantes s'obtiennent par combinaison des trois précédentes. Les spécifications doivent être communiquées aux concepteurs de l'interface de supervision et du programme de commande.

Sur de tels projets de conception, le nombre des intervenants et la diversité des domaines d'études entraînent des problèmes de cohérence d'ensemble et d'interprétation des spécifications. De fait, les erreurs issues de ces problèmes de compréhension ne sont détectées que lors des phases d'essai du système conçu. En effet, 79% des défaillances trouvent leur origine dans les prestations d'étude et de réalisation [24], alors que 72% des défaillances ne sont détectées que lors des essais de l'exploitation [21].

L'utilisation de l'IDM permet d'envisager de s'affranchir de ces problèmes d'interprétation. En effet, l'essentiel de la sémantique du système est contenue dans le schéma PID. Les opérations possibles sur les différents éléments du système dépendent directement de leur nature (visualiser une alarme, ouvrir/fermer une vanne, acquitter un alarme, etc.). Quant aux règles d'animation (écoulement de l'eau), elles dépendent de la configuration globale du réseau.

La solution que nous proposons consiste à combiner une approche à base de composants pour prendre en compte la nature même d'un système sociotechnique, lui-même constitué de composants physiques, à une approche IDM permettant de prendre en compte et de transformer la sémantique implicite de l'agencement métier des composants.

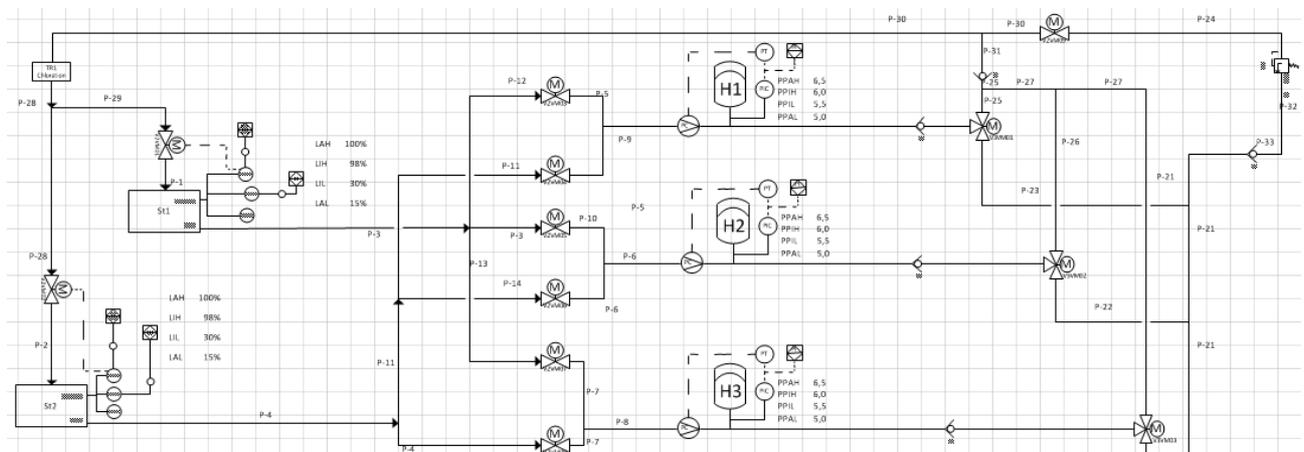


Figure 3. Schéma PID du système

DESCRIPTION DE LA SOLUTION

L'objectif du projet de recherche *Anaxagore* est de proposer des méthodes et outils de conception innovants qui garantissent la qualité des produits documentaires et logiciels d'un projet tout en réduisant le temps de conception et de re-conception. La démarche s'appuie sur une approche ascendante utilisant des éléments standardisés pour réaliser le schéma synoptique. Des passerelles entre les différents corps de métiers sont alors établies par un jeu de transformations automatiques de modèles [20] afin de faciliter les échanges entre les concepteurs.

Étape de construction du synoptique

Cette étape concerne la définition du synoptique des éléments du système à concevoir. Ce type de schémas peut par exemple être saisi sous Microsoft Visio[®]. Cet outil offre par ailleurs la possibilité de prédéfinir les symboles du schéma afin d'en respecter la norme, et permet d'enregistrer le schéma au format XML.

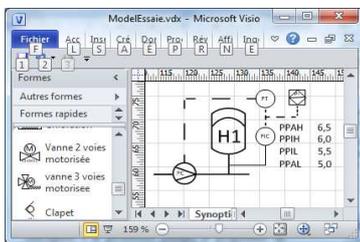


Figure 4. Vue Synoptique de l'hydrophore

La Figure 4 propose ainsi une vue de Microsoft Visio[®] tel qu'il a été adapté pour le projet. À gauche, sont représentés différents composants prédéfinis pouvant entrer dans la conception des systèmes sociotechniques. À droite est dessiné un modèle de composant (hydrophore) prédéfini, avec ses différentes caractéristiques. La réalisation du schéma (sur la Figure 3) se fait par assemblage de ces symboles prédéfinis sur la Figure 4.

Méta-modèle de la bibliothèque d'Anaxagore

Les éléments standardisés d'*Anaxagore* sont regroupés dans une bibliothèque (Figure 5). Chaque élément comporte des informations organisées en autant de « vues » qu'il y a de concepteurs aux profils différents. Un élément est rattaché à ses vues par les attributs *vSyn* (relatif au synoptique), *vSup* (relatif à la supervision) et *vCmd* (relatif à la commande) qui sont des URI¹. Il est caractérisé par les attributs *family*, *designation* et *reference*. Enfin un élément peut avoir des interfaces (Interface sur la Figure 5).

Du fait de la finalité opérationnelle, des outils spécialisés sont utilisés pour créer les vues des éléments. Ainsi la vue de supervision d'un élément est, dans notre cas, implémentée sur Panorama E2. De la même manière,

nous utilisons l'environnement de programmation d'automate Straton pour créer la vue de commande. Une vue de commande est en fait un bloc fonctionnel au sens de la norme IEC 61131 [12].

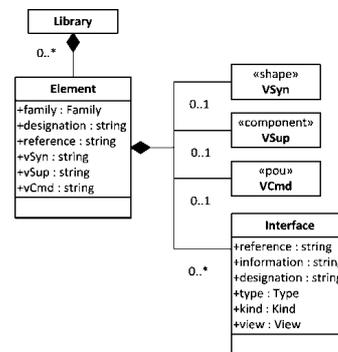


Figure 5. Méta-modèle de la bibliothèque d'Anaxagore [5]

Tous les éléments de conception d'un système ne peuvent être standardisés. Pour certains éléments spécifiques, le surcoût engendré par la standardisation rend cette opération contre-productive. Nous avons donc défini dans la bibliothèque un élément générique qui peut être adapté à un besoin spécifique. Chaque élément adapté ainsi défini est stocké dans une bibliothèque indépendante. A partir du schéma PID et des bibliothèques, des méthodes d'association d'informations et de dérivation de modèles permettent de générer progressivement les modèles opérationnels.

Démarche de génération

La Figure 6 présente le flot de conception proposé dans *Anaxagore*. Nous nous intéresserons plus particulièrement dans la section suivante à la partie concernant la génération de l'IHM de supervision (partie claire).

01. Opération de Construction. A partir des cahiers des charges et d'une bibliothèque d'éléments standard (*Bibliothèque standard*), l'expert en charge de la conception du système construit un schéma du système à concevoir (*Schéma synoptique*). *02. Opération d'Inventaire* est une opération automatique d'inventaire basée sur la bibliothèque et sur le synoptique. Cette opération permet de générer une nomenclature (*Nomenclature*). *03. Opération de spécialisation* déroule un processus d'adaptation, alors que *04 Opération d'Insertion* est l'opération qui permet l'insertion automatique des informations relatives à la supervision issues des vues des éléments dans un modèle standard d'IHM (*modèle standard d'IHM*), à partir de la nomenclature et des bibliothèques. *05 Opération d'Assemblage* réalise l'assemblage des vues de commande issues des éléments dans un programme élémentaire (*Basic Component-Based Program*).

Enfin, *06 Opération de Liaison* génère un serveur de donnée OPC (*OLE for Process Control Data Server*) permettant de gérer les échanges entre la supervision et le programme de commande.

¹ Uniform Resource Identifier, il s'agit dans notre cas d'une chaîne de texte représentant le chemin d'accès à une vue

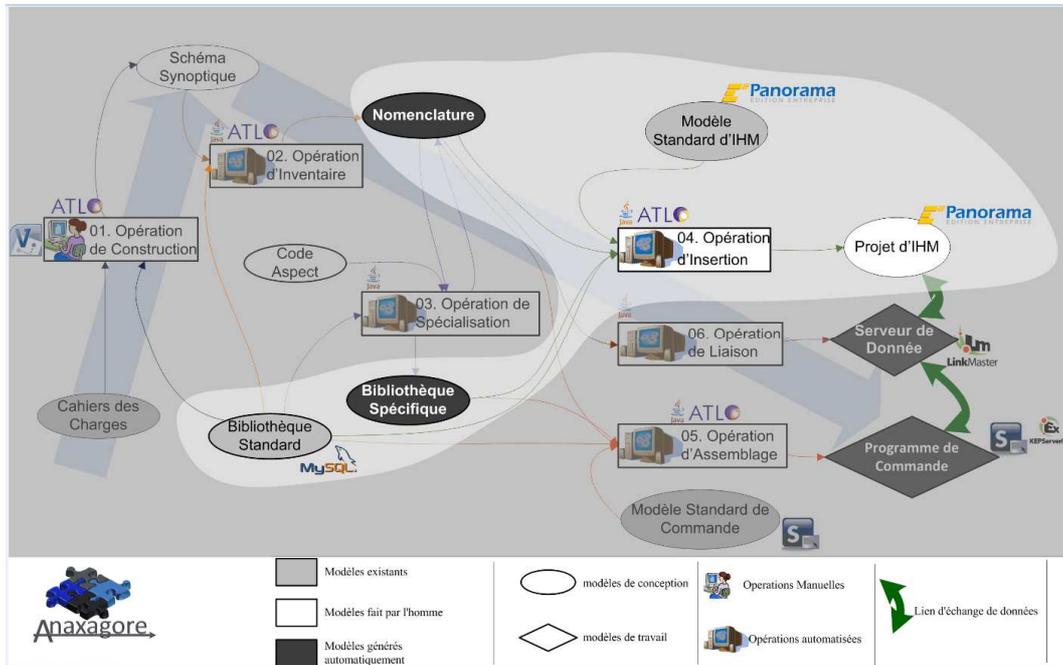


Figure 6. Flot de conception intégré.

CONSTRUCTION AUTOMATIQUE D'INTERFACE

À ce jour, le projet *Anaxagore* permet de générer complètement une interface de supervision et un programme de commande élémentaires à partir d'un schéma PID et des bibliothèques d'éléments. Le cas d'application considéré est un système de production de distribution et de stockage de l'eau douce embarquée sur un navire, noté EdS¹. La Figure 7 présente l'IHM générée à partir du schéma PID de la Figure 3, un sous-système du système EdS et présentant les dispositifs de stockage (soutes d'eau douce) et de distribution (groupes hydrophores). Cette interface permet notamment de jouer des scénarios de transfert d'eau entre les 2 soutes. Dans le logiciel Panorama E2, elle se compose d'un nombre important d'objets de base du logiciel, que l'on peut assimiler à des widgets de haut niveau généralement connectés à un programme de contrôle/commande.

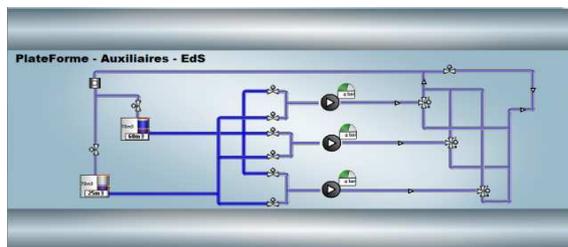


Figure 7. IHM générée

Par exemple, l'objet *Alarme* prend en charge le traitement d'une alarme de l'installation supervisée. Avec Panorama E2, on peut créer facilement des objets complexes. En effet, la notion de composant (Figure 8) permet de définir des modèles à partir desquels il est

possible de créer rapidement des objets complexes en faisant appel à des mécanismes d'instanciation.

L'objet V2VM présent sur la Figure 8, censé contrôler une vanne de distribution de fluide, est ici composé de deux objets *Alarme* (Al_StFMat et V2VM_StFCmd) qui se déclenchent lorsqu'il y a un défaut au niveau de la vanne, d'un objet *Variable* (Remplis) qui définit la circulation du fluide dans la vanne, et de deux *vues* (V2VM et V2VM_Control). La vue V2VM représente la vue de la vanne visible sur l'IHM. La vue V2VM_control (bandeau de contrôle) apparaît sur l'IHM lorsque l'utilisateur sélectionne la vanne en cliquant dessus. Elle permet à l'utilisateur d'interagir avec l'objet sélectionné.

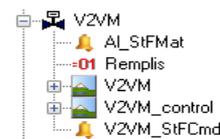


Figure 8. Modèle arborescent du composant vanne

La conception d'une telle IHM peut s'avérer très fastidieuse pour les concepteurs. Le nombre de composants nécessaires devient vite très important : par exemple, cet extrait pourtant très modeste est composé de 123 objets et 311 connexions. De plus, si le principe des widgets permet à chaque composant d'encapsuler son comportement, la dynamique de l'animation nécessaire à la mise en évidence des phénomènes physiques du système nécessite une interaction entre les composants et doit être programmée « à la main ».

Par exemple, sur la Figure 7 toutes les vannes sont fermées, l'eau ne passe donc qu'entre les soutes et les premières vannes. Les tuyaux en bleu foncé sont remplis

¹ EdS : Eau douce Sanitaire

d'eau et ceux en bleu clair sont vides. Pour faire passer l'eau l'utilisateur doit ouvrir une vanne. Lorsque l'utilisateur clique sur une vanne pour l'ouvrir, un bandeau de contrôle s'affiche avec les boutons lui offrant la possibilité d'agir sur le composant (Figure 8). Lorsqu'il clique sur le bouton Ouverture/Valider, la commande associée à ce bouton envoie au système physique un ordre de changement d'état. Si la vanne n'a aucun problème, la manœuvre s'effectue correctement, et l'état du capteur est renvoyé au système. La variable correspondant dans l'API change alors d'état et est transmise à la supervision pour montrer à l'utilisateur que la vanne est ouverte. Dans ce cas l'utilisateur observe les animations : la vanne passe en bleu foncé (il y a de l'eau qui traverse la vanne) ainsi que les tuyaux en aval de la vanne (Figure 9).

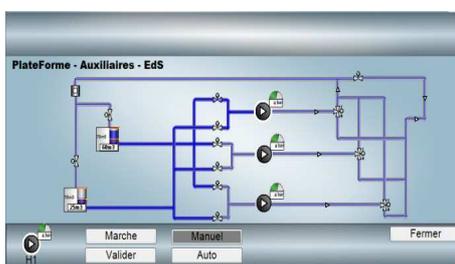


Figure 9. Animation de la propagation de l'eau

S'il y a un problème, le système réel ne réagit pas à la commande de l'automate et au bout d'un délai paramétrable, l'automate signale au superviseur qu'il y a défaut de matériel. Ceci déclenche une alarme signalant le type de défaut, la vanne passe alors en rouge.

On imagine bien la difficulté de programmer dans un logiciel comme Panorama E2 la dynamique entre les composants graphiques nécessaires à la bonne compréhension de l'opérateur.

Anaxagore offre une bonne solution à ce problème en automatisant complètement la construction de l'interface sous Panorama E2. Pour générer automatiquement l'IHM de la Figure 7, on utilise, dans un premier temps, une transformation de modèle qui permet à partir du schéma PID de la Figure 3 d'obtenir un modèle de synoptique indépendant de toute plateforme. On utilise ensuite des règles implémentées avec ATL, le langage de transformation de modèle qui fait partie du projet Eclipse M2M (Model-to-Model). Ce langage fournit des moyens de produire un ensemble de modèles cibles à partir d'un ensemble de modèles sources. Les règles de transformations utilisées pour l'épuration du fichier Visio traitent des informations relatives au document (domaine, système et sous-systèmes), à un symbole, aux connexions entre les symboles et permettent d'obtenir un modèle de synoptique correspondant à un méta-modèle simple et contenant toutes les informations dont on a besoin pour générer complètement l'IHM du système.

L'étape suivante de notre démarche est la génération du modèle de *nomenclature* (Opération d'inventaire sur la

Figure 6). La première partie de cette opération permet de référencer les interfaces des éléments de la bibliothèque représentés dans le synoptique. Pour cela, on utilise le modèle de synoptique obtenu précédemment et les bibliothèques. La deuxième partie permet de générer, en utilisant une transformation de modèle ATL, le modèle de la *nomenclature* à partir de la bibliothèque, du modèle de synoptique et des interfaces collectées lors de la première partie de cette opération. Le modèle de nomenclature généré contient toutes les informations nécessaires à la génération de l'IHM. Par exemple sur la Figure 4 toutes les informations relatives aux paramètres PPAH et PPAL de l'hydrophore sont renseignées dans la *nomenclature* (Figure 10).

```
<instance family="Process" designation="Hydrophore" reference="HP"
id="H1" subSystem="EdS">
  <interface reference="HP" id="H1" information="PPAH"
designation="Parametre Seuil tres haut" type="REAL"
kind="Var" view="VSup" subSystem="EdS"/>
  <interface reference="HP" id="H1" information="PPAL"
designation="Parametre Seuil tres bas" type="REAL"
kind="Var" view="VSup" subSystem="EdS"/>
  <coord x="195.0" y="79.86667"/>
  <ci nature="PPAH" value="6.5" type="WORD"/>
  <ci nature="PPAL" value="5" type="WORD"/>
</instance>
```

Figure 10 Extrait de la nomenclature

La génération de l'IHM correspond à l'Opération d'Insertion détaillée sur la Figure 12. Elle permet d'obtenir le « projet d'IHM » de la Figure 7 à partir du modèle de la bibliothèque Panorama, du modèle de la *Nomenclature* et du modèle standard d'IHM. Le modèle standard d'IHM dispose de six zones numérotées sur la Figure 11. La zone 1 reçoit le nom du système représenté sur l'interface. La zone 2 permet de visualiser la liste des alarmes du système. La zone 3 reçoit les commandes de haut niveau permettant de réaliser les fonctions du système. La zone 4 reçoit la représentation du système. Tous les composants de schéma PID sont représentés par leur vue de supervision. La zone 5 permet de visualiser les boutons de commande d'un élément sélectionné dans la zone 4. La zone 6 contient les menus de navigation entre les interfaces des différents sous-systèmes du navire.



Figure 11 Modèle standard d'IHM

Le modèle standard d'IHM et le « projet d'IHM » généré sont conformes au méta-modèle de Panorama E2. Le

projet d'IHM comporte une bibliothèque Panorama regroupant toutes les vues de supervision des composants prédéfinis et deux fichiers de configuration par interface : le premier définit l'architecture de l'interface, et le second définit la structure de l'affichage de l'interface. Les transformations réalisées dans l'opération d'insertion se font en trois phases (Figure 12): la phase de préparation, la phase de création de l'architecture et la phase de création de la structure.

Pendant la phase de préparation, les composants disposant d'une vue de supervision, associés à leurs instances dans la nomenclature, sont, dans un premier temps, insérés dans la bibliothèque de Panorama. Puis, l'utilisation d'une applet Java permet d'identifier les dimensions des symboles afin de les positionner sur l'IHM conformément au schéma PID.

Lors de la deuxième phase, les composants de l'IHM avec leurs dimensions sont ensuite instanciés conformément aux instances listées dans la nomenclature. Cette phase a été réalisée avec une transformation ATL.

Enfin, la dernière phase permet de créer concrètement l'IHM en détaillant la position des symboles instanciés.

Le projet d'IHM obtenu après la réalisation de cette opération est cohérent avec le schéma PID de la Figure 3 construit par l'expert. L'IHM ainsi visible par l'utilisateur prend en compte ses commandes de bas-niveau (consigne exprimée sur un élément du procédé représenté sur l'IHM) et lui fait un retour d'information sur l'état du composant sur lequel il vient d'interagir.

L'animation de l'interface est gérée par Panorama E2 au travers d'équations générées par *Anaxagore*. À chaque nouvel état de l'interface, le moteur de Panorama E2 recalcule les équations, et adapte la visualisation en fonction de leurs résultats. L'utilisateur peut donc commander et surveiller chacun des composants du système présent sur l'IHM. À partir de ces actions sur l'IHM, les changements d'état des variables sont communiqués, au travers d'un serveur OPC, à un API (Automate Programmable Industriel) qui exécute le programme de commande élémentaire généré conjointement avec l'IHM. Le retour d'information à la supervision est réalisé à travers ce même serveur. Ce retour permet à l'utilisateur d'observer la réaction du système réel par rapport à son action sur celui-ci.

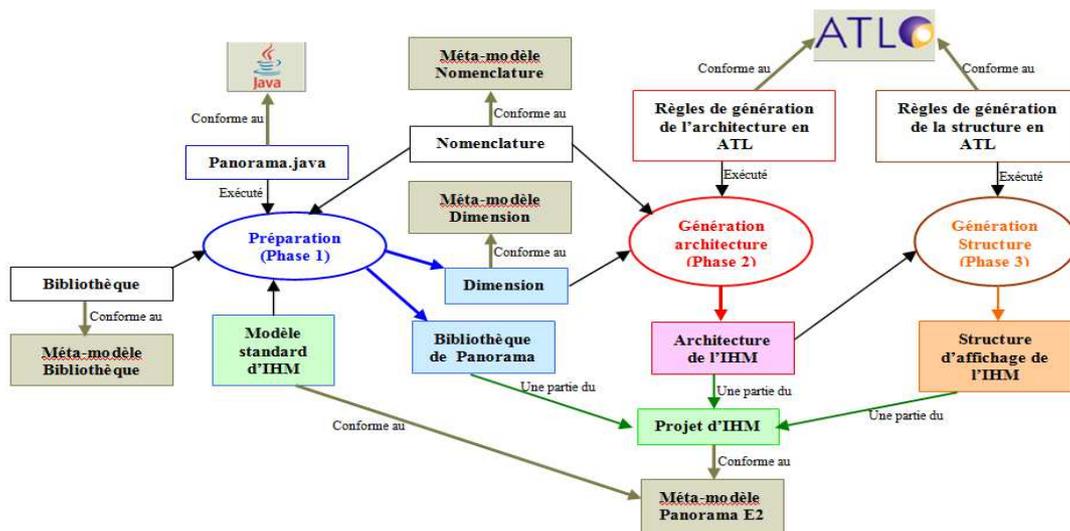


Figure 12 Description de l'opération d'insertion (Génération de l'IHM)

CONCLUSION

Anaxagore permet de générer automatiquement un système interactif de supervision directement à partir d'un schéma PID décrivant le système sociotechnique lui-même. Les connaissances métiers indispensables à la bonne prise en compte de la sémantique du système sont gérées par *Anaxagore* pour produire toutes les données permettant à un logiciel spécialisé de type SCADA de piloter la supervision.

Cependant, les tâches d'un opérateur de supervision sont plus complexes que celles que nous avons présentées. Un système de type EdS réalise à bord du navire plusieurs fonctions de haut-niveau, telles que la production et la distribution d'eau douce,

l'embarquement et le débarquement de l'eau. La réalisation de ces fonctions nécessite d'interagir avec plusieurs composants selon un processus spécifique, ce qui peut s'avérer fastidieux.

L'IHM actuelle ne prend pas en compte l'activité de l'utilisateur pour la description des tâches de haut niveau. Nous projetons d'intégrer à cette démarche une application permettant à l'utilisateur final, expert métier, mais pas forcément informaticien d'enrichir l'IHM générée par ces activités de haut-niveau. Nous pensons apporter une aide à l'utilisateur en lui donnant la possibilité de programmer directement dans l'IHM utilisant pour cela les techniques de programmation par démonstration.

BIBLIOGRAPHIE

1. Allegre W. Flot de conception dirigé par les modèles pour la commande et la supervision de systèmes domotiques d'assistance. *Thèse de doctorat, Université de Bretagne-Sud* (2012).
2. ANSI/ISA-5.1-1984 (R1992). Instrumentation Symbols and identification. *A. N. STANDARD.* (1992).
3. Barmes K. A., Hornaes A., Jacobsen O., Storkas R., (1992). PICASSO : A User Interface Management System for Real-Time Applications. Workshop User Interface for Expert Systems, London, (1992).
4. Bézivin, J (2004). Sur les principes de base de l'ingénierie des modèles. *RSTI-L'Objet*, 10(4), p.145-157.
5. Bignon, A., A. Rossi et P. Berruet (2013). An integrated design flow for the joint generation of control and interfaces from a business model. *Computers in industry* 64. p. 634-649.
6. Céret E., Calvary G., Dupuy-Chessa S. Flexibility in MDE for scaling up simple applications to real case studies: illustration on a nuclear power plant. *IHM13* (2013), p. 33-42.
7. Calvary, G., Coutaz, J., Bouillon, L., Florins, M., Limbourg, Q., Marucci, L., Paternò, F., Santoro, C., Souchon, N., Thevenin, D., Vanderdonckt, J.: The CAMELEON reference framework. CAMELEON Project.Deliverable 1.1 (2002), <http://giove.isti.cnr.it/projects/cameleon/pdf/CAMELEON%20D1.1RefFramework.pdf>
8. Calvary G., Dellot T., Sèdes F., Tigli J-Y. Système interactifs et adaptation centre utilisateur: la plasticité des Interfaces Homme-Machine. *Informatique et intelligence ambiante : des capteurs aux applications.* HERMES. (2012). p.175-221.
9. Elwert, T. and Schlunbaum, E. Modelling and Generation of Graphical User Interfaces in the TADEUS Approach. *DSV-IS*, (1995), p. 193-208.
10. Gabillon Y., Calvary G., Fiorino H. L'IDM passerelle entre IHM et planification pour la composition dynamique de systèmes interactifs. *IDM'08.* (2008), p. 51-56.
11. Hoc, J.M. Supervision et contrôle de processus, la cognition en situation dynamique. *PUG*, (1996). 207p.
12. IEC 61131 (2003).
13. Jambon F., Chappellet K, Calvary G., Plasticité des Interfaces par Perception de l'Interaction Homme-Machine : illustration en oculométrie. *IHM13* (2013), p.1-2.
14. Kolski C. Exemple de développement d'un outil d'ingénierie pour la conception d'interface: le système ERGO CONCEPTOR - *Ingénierie des interfaces homme-machine conception et évaluation.* Paris: HERMES, 1993, p 289- 334. ISBN 2-86601-377-8
15. Kolski C. et Ezzedine H. Conception et évaluation des IHM de supervision : éléments méthodologiques. *Revue Génie Logiciel* (2003), 65, pp.2-11.
16. Microsoft. (2011). "MSDN Library." from <http://msdn.microsoft.com/enus/library/ms123401.aspx>
17. Moray, N. Human factors in process control. In *Handbook of human factors and ergonomics*, G. Salvendy (Ed.), John Wiley & Sons, INC., (1997) p.1944-1971.
18. Moussa F., Kolski C., Riahia M. A model based approach to semi-automated user interface generation for process control interactive applications. *Interacting with Computers* 12. (2000). p. 245-279
19. Nichols J., Myers B. A., Higgins M., et al. Generating remote control interfaces for complex appliances. *UIST*, (2002), p.161-170.
20. OMG (2003). MDA Guide version 1.0.1. OMG.
21. Pham H. System Software Reality (Spring Series in Reality Engineering), *Springer-Verlag*, New York (2005).
22. Skaf A. Etude d'un système de supervision et de commande d'un procédé complexe comme un élément de base d'une organisation distribuée comprenant des machines et des hommes. *Thèse en automatique et Productique, Université Joseph Fourier.* (2001).
23. Sottet J.-S., Calvary G. & Favre J. M. Ingénierie de l'Interaction Homme-Machine Dirigée par les Modèles. *IDM'05.* (2005).
24. Sourisse, C et Boudillon, L. La sécurité des machines automatisées : Techniques et moyens de prévention opératifs, systèmes de commandes, utilisation des machines. Institut Schneider Formation, (1997).
25. Szekely, P. Retrospective and Challenges for Model-Based Interface Development. *Proceedings Design, Specification and Verification of Interactive Systems '96, DSV-IS 96, Vienne, Autriche*, (1996), pp. 1-21.
26. Zamaï E., Rigaud F., Pétin J.F., Berruet P. et Toguyeni A. Architectures de pilotage de procédés industriels. *Techniques de l'Ingénieur*, (2007), AG 3 510 p.1-19. <http://www.techniques-ingenieur.fr/res/pdf/encyclopedia/42602210-ag3510.pdf>