

# Le Livre Blanc du MES

---

*Pilotage temps-réel et suivi des fabrications pensés comme un système intégré*

## Sommaire

Introduction .....	3
Le rôle décisif du standard ISA-95 (IEC 62264) .....	4
La genèse du MES – les 11 fonctions .....	4
L’ISA et les objectifs de l’ISA-95 .....	5
Le modèle fonctionnel de l’ISA-95 .....	5
Relation avec les niveaux du CIM.....	6
Le modèle objet d’échange.....	7
La hiérarchie des équipements.....	7
Le modèle opérationnel (modèle d’activités).....	8
Les échanges avec le niveau 4 du CIM (ERP/GPAO).....	9
La fusion ISA-95 / ISA-88 .....	9
Intégration verticale et standardisation.....	10
Typologie des logiciels de MES.....	11
L’importance des procédés ciblés .....	11
Approche spécifique (métier) .....	11
Assemblage de logiciels différents .....	12
Approche hybride.....	13
Progiciels intégrés .....	14
Intégration MES et Supervision.....	14
Le déploiement d’un MES .....	15
Un projet informatique .....	15
Technologies mises en œuvre.....	16
Compétences requises.....	16
Facteurs de succès et causes d’échec .....	16
Capitalisation et « modèle cœur » .....	17
Conclusion.....	18

## Introduction

La performance de la production industrielle est un élément clé dans la réussite des entreprises. Pour cela la production doit fabriquer les produits dans toute leur diversité en suivant des contraintes de délai, de qualité et de coût, et s'adapter en permanence à un environnement changeant : contraintes réglementaires, demande du marché et innovation.

Conçu pour aider à l'atteinte de ces objectifs, le MES est passé de l'état de concept à un marché dynamique délivrant des offres logicielles standard et abouties, permettant une réelle capitalisation des développements.

A quoi répond, le MES ? Quelques questions simples permettront de l'appréhender un peu plus concrètement.

*« De quel fournisseur provenait le colorant utilisé dans ce yaourt refusé par le contrôle qualité ? L'équipe du matin est-elle plus performante que celle de l'après-midi ? La cuve de mélange ayant servi à la préparation des produits du 13/9 avait-elle été lavée ? Combien de points de productivité perdons-nous sur la ligne 4 à cause des réglages ? Est-il possible de passer la promo avant la livraison prévue pour demain matin, ou en même temps ? Quel était le dosage exact du sucre dans le lot de produits bio ? Les opérateurs préfèrent utiliser la machine X pour ce produit, il y a moins de problèmes, mais n'est-on pas perdant au global ? ...Etc. »*

Voilà le genre de questions auxquelles un MES (Manufacturing Execution System) va tenter de répondre. Et très rapidement, les industriels ont pu s'apercevoir que ni les systèmes dont ils disposaient pour la production (contrôle-commande et supervision), ni les progiciels de gestion (en particulier les ERP – Enterprise Resource Planning) ne le permettaient. Il y avait donc bien un domaine non couvert.

Au début des années 90, une association américaine a créé le terme de Manufacturing Execution System pour désigner ce nouveau domaine, qui a d'ailleurs donné son nom à l'association elle-même (MESA). Le MESA s'est aussi fait connaître par les « 11 fonctions du MES » que nous verrons par la suite.

Néanmoins, arrivé nettement après les autres logiciels, le MES a d'abord un peu pâti de cette situation. Un peu comme dans les auberges espagnoles d'antan, où les clients apportaient eux-mêmes à manger, les industriels ont cherché à définir comme appartenant au MES tout ce qu'ils ne trouvaient pas dans leurs outils existants...

Aujourd'hui, comme nous allons le voir, grâce à des travaux importants comme ceux de l'ISA (International Standard of Automation), et aux efforts de recherche des éditeurs de logiciels, le MES a réellement pris sa place dans la palette des outils indispensables de l'entreprise pour relever ses défis.

## Le rôle décisif du standard ISA-95 (IEC 62264)

### **La genèse du MES – les 11 fonctions**

Les premiers pas du MES se sont donc traduits par l'identification par le MESA de onze fonctions qu'il était amené à couvrir (voir encadré). Et pendant longtemps ces onze fonctions ont pratiquement servi de définition au MES lui-même. Le périmètre du MES était donc clarifié ? En fait, pas tant que cela.

#### **Les 11 fonctions du MES**

- Gestion des ressources
- Ordonnancement
- Cheminement des produits et des lots
- Gestion des documents
- Collecte et acquisition de données
- Gestion du personnel
- Gestion de la Qualité
- Gestion du procédé
- Gestion de la maintenance
- Traçabilité produit et généalogie
- Analyse des performances

Le premier problème tient aux fonctions qui sont énumérées. On distingue en effet la *Gestion des ressources* et la *Gestion du personnel*. Mais du point de vue de l'informatique industrielle, le personnel n'est-il pas une ressource ? Le MES aurait pour fonction la *Gestion des documents*. Cela veut-il dire que toute la gestion des documents d'une entreprise doit être prise en charge par le MES ? Claire à première lecture, on voit que cette définition du MES prête à pas mal d'ambiguïtés.

Le second problème est plutôt sémantique. En définissant le MES par la liste des fonctionnalités qu'il est supposé assurer, on laisse penser que le tout se résume à la somme de ses parties. Or la plupart des fonctions listées n'ont pas attendu la création du terme MES pour être mises en œuvre, même partiellement. Du coup, bon nombre d'industriels se sont dits qu'ils faisaient déjà du MES sans le savoir, comme M. Jourdain faisait de la prose...

C'était passer à côté de l'essentiel, essentiel qui était dans le terme même de MES mais que les initiales tendaient à faire oublier. La seconde lettre du sigle désigne en effet *l'exécution*. Car les fonctions du MES sont en fait orchestrées autour d'un processus central qui est l'exécution des fabrications, c'est à dire *l'enchaînement des opérations qui aboutissent à la fabrication des produits*. Et la troisième lettre du sigle désigne un *système*. Or dans un système, les fonctions ne sont pas indépendantes et séparables, mais forment un tout. La figure 1 montre comment les fonctions généralement identifiées dans le MES peuvent en fait s'organiser en *fonctions opérationnelles* et en *services transversaux*, susceptibles d'être utilisés par toutes les fonctions opérationnelles.

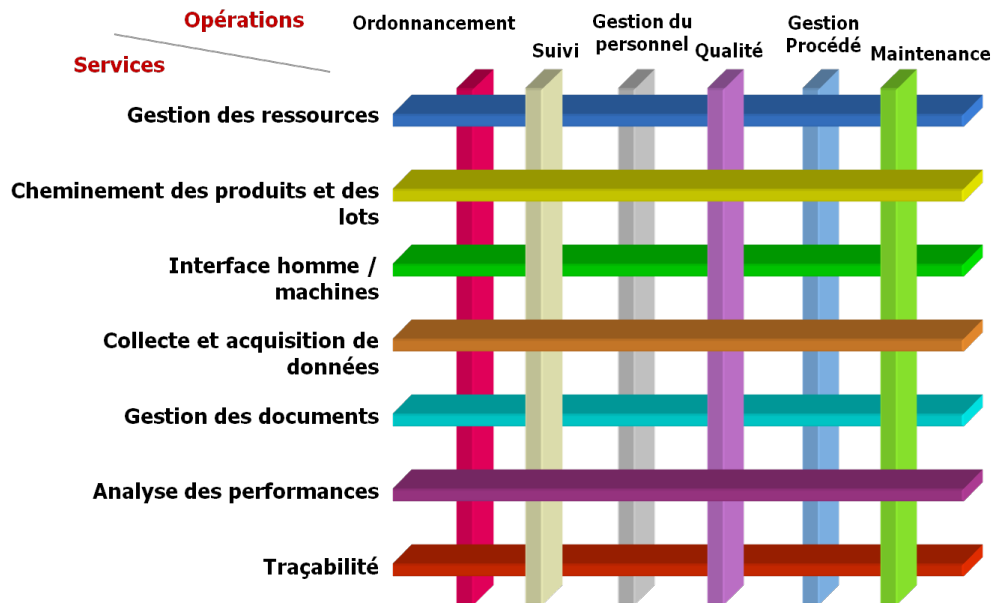


Figure 1 - Le MES est un système (source ORDINAL Software)

## L'ISA et les objectifs de l'ISA-95

L'ISA, déjà présentée en introduction, est une association américaine forte de 39 000 membres de l'industrie. A l'origine tournée vers l'instrumentation, elle a étendu le champ de ses travaux à l'ensemble de l'automatisation, et est à l'origine de nombreux standards indépendants. Un nombre conséquent de ces standards a été repris au niveau européen sous les dénominations IEC. Cela a été le cas en particulier pour le standard ISA-95, repris au niveau européen sous la dénomination IEC 62264.

Les objectifs de l'ISA-95 sont multiples. Un premier objectif, en apparence modeste, a été de faire partager aux industriels une terminologie commune pour les notions manipulées par le MES. En effet, les industriels ont l'habitude d'utiliser les termes propres à leur métier pour les désigner, ce qui contrarie une approche générique des problèmes. Un industriel va parler de rognures, un autre de déchets, un troisième de rebuts, mais il y a fort à parier qu'il s'agisse là de la même notion. Un des objectifs de l'ISA-95 a donc été de clarifier le domaine en fournissant cette terminologie commune.

Avec la progression considérable des ERP dans les entreprises, le dialogue du MES avec ces logiciels est clairement apparu comme un passage obligé – et même pour certains une raison d'être – pour le MES. Le second objectif de l'ISA-95 a été de fournir un ensemble de modèles d'échange entre le MES et les ERP.

Les travaux de l'ISA-95 se sont ensuite étendus jusqu'à fournir un modèle opérationnel pour le MES, qui peut servir de base à la structuration des logiciels du marché. Ces travaux fournissent aujourd'hui une base solide tant pour les éditeurs de logiciels pour mieux répondre aux demandes industrielles, que pour les industriels qui peuvent s'y appuyer pour la rédaction de leurs cahiers des charges.

## Le modèle fonctionnel de l'ISA-95

Le modèle fonctionnel de l'ISA-95 est dérivé des travaux de modélisation du fonctionnement d'une entreprise réalisés par l'université de *Purdue*, dans l'Indiana. L'originalité de cette modélisation est de faire apparaître le contrôle de production au centre du modèle, et donc de mettre en évidence les liens de celui-ci avec les différents services de l'entreprise (figure 2). La nappe « déployée » autour du contrôle de production représente le périmètre couvert par le MES.

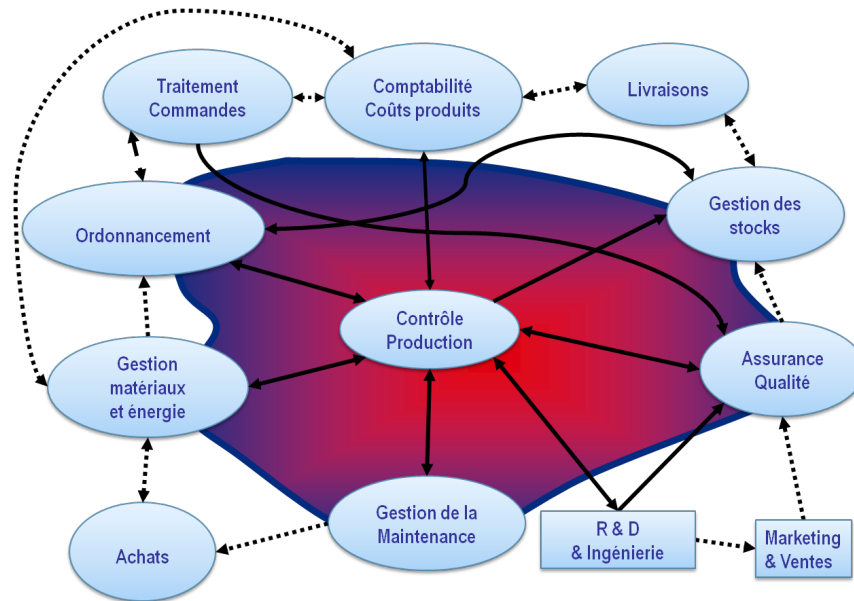


Figure 2 - Le modèle fonctionnel du MES (source ORDINAL Software, d'après l'ISA)

Un contresens est parfois fait qui consiste à penser que ce modèle « impose » que l'entreprise soit organisée de cette façon, c'est-à-dire que les services représentés existent « physiquement » dans l'entreprise. Ce n'est évidemment pas le cas. Mais cette représentation aide à représenter les flux (explicites ou implicites) entre les différentes entités.

L'ISA-95 va jusqu'à un haut niveau de détail dans la description de ces flux. Par exemple, le service d'Assurance Qualité va fournir au Contrôle de Production les standards et les exigences Clients ; ce dernier fournira en retour au service d'Assurance Qualité les résultats des contrôles qualité. L'Ordonnancement fournira au Contrôle de Production le plan de production, lequel fournira en retour la capacité de production restante, et ainsi de suite. Par contre, il n'y a pas d'échange direct entre le Contrôle de Production et le service Comptabilité.

### Relation avec les niveaux du CIM

Les niveaux du CIM (Computer Integrated Manufacturing), bien que de définition assez ancienne, servent souvent de référence pour situer le périmètre de travail des matériels et des logiciels dans le secteur industriel. La figure 3 permet de situer le domaine de MES au sein des niveaux du CIM.

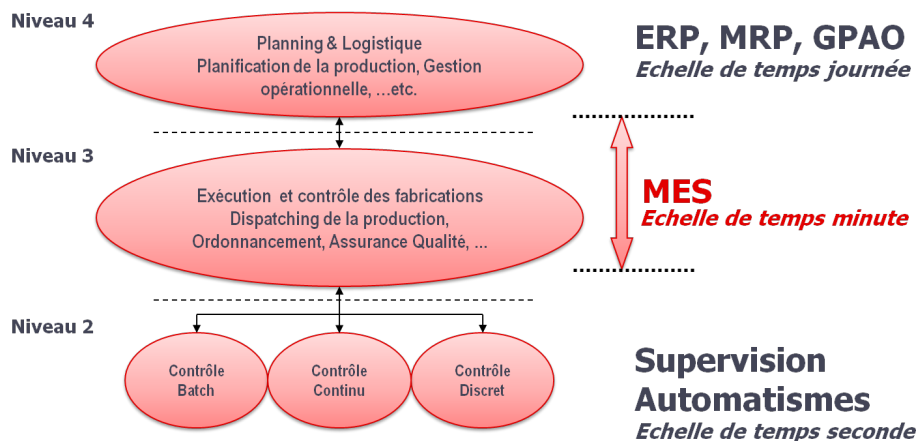


Figure 3 - Le MES et les niveaux du CIM (source ORDINAL Software, d'après l'ISA)

Le MES ne situe au niveau 3 du CIM, entre les Progiciels de Gestion Intégrée ou ERP (SAP, Oracle Applications, JD Edwards, PeopleSoft, ..) du niveau 4, et les logiciels de supervision et d'automatismes du niveau 2 (Rockwell, Schneider, Siemens, ...). Un point intéressant à noter est que les différents niveaux ne traitent pas l'information à la même échelle de temps : alors que les ERP au niveau 3 traitent des informations à l'échelle de la journée, la supervision et les automatismes à l'échelle de la seconde ou même moins, le MES au niveau 2 se situe plutôt à l'échelle de la minute. En effet, au niveau de l'exécution des fabrications, il faut réagir en moins de quelques minutes ou dizaines de minutes si une ligne ne produit pas comme attendu, ou que les contrôles qualité révèlent un problème de fabrication. Le MES réalise également le lien indispensable entre les usines et la gestion de l'entreprise, même s'il serait trop réducteur de limiter son rôle à ce lien. Dans ce contexte, l'ISA-95 s'est donné comme première tâche de définir les échanges entre le niveau 3 et le niveau 4, et pour cela définir un modèle objet d'échange.

### ***Le modèle objet d'échange***

Le modèle objet d'échange entre le MES et le niveau 4 constitue une partie majeure des travaux de l'ISA-95. Pour la première fois en effet, un modèle de données très complet (près de 200 types d'objets sont définis) permet d'aborder l'exécution des fabrications suivant trois axes de ressources (matières, équipements, personnel) et en mettant en œuvre des objets de haut niveau tels que le segment de procédé, le segment produit, la définition produit, ...etc.

Défini sur la base d'un langage standard légèrement simplifié (UML : Unified Modeling Language), ce modèle objet, même s'il n'a pas vocation à être implémenté tel que décrit, constitue néanmoins une base sérieuse pour la réalisation d'un logiciel de MES générique, ce qui est la vocation des logiciels standards. Il est même possible de l'utiliser comme modèle de référence interne (ce que fait le logiciel COOX d'ORDINAL). Il a essentiellement été conçu pour assurer les échanges bilatéraux entre le logiciel de MES et l'ERP.

### ***La hiérarchie des équipements***

La hiérarchie des équipements vise à décrire l'installation physique sous une approche générique, applicable à tout type d'entreprise vue sous l'angle de ses unités de fabrication. C'est aujourd'hui l'une des parties les plus utilisées du standard ISA-95, qui permet de partager une vision simple d'une installation de production. La figure 4 montre cette modélisation.

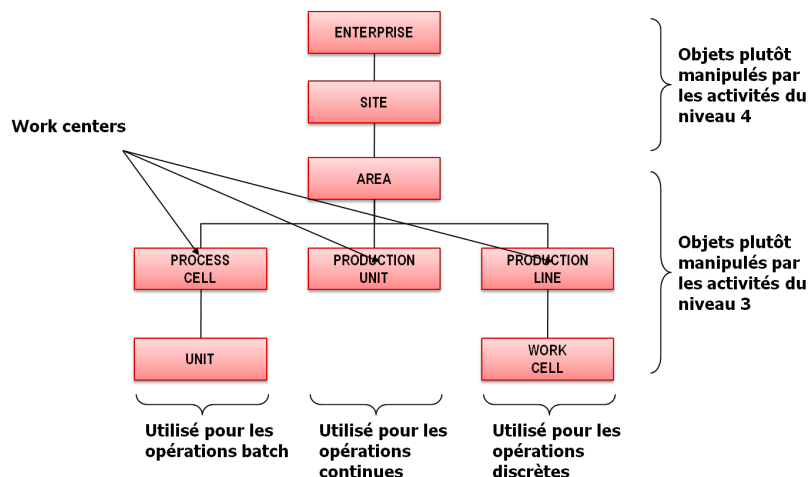


Figure 4 - Hiérarchie des équipements (source ORDINAL Software, d'après l'ISA)

Dans cette schématisation, une entreprise pourra comprendre un ou plusieurs sites, chacun d'eux comprenant une ou plusieurs *area* (bâtiments par exemple), elles-mêmes divisées en plusieurs sous-ensembles. On note bien qu'à ce niveau, c'est-à-dire dans la partie basse de la hiérarchie des équipements, la terminologie change suivant le type de procédé concerné : batch, continu ou discret. Ces types de procédés sont généralement considérés comme forts différents, et la hiérarchie ISA-95 reflète cette différenciation. Toutefois, l'ISA-95 propose d'adopter la terminologie unifiée de work center (que l'on pourrait traduire par atelier) pour les sous-ensembles de l'*area*.

### Le modèle opérationnel (modèle d'activités)

Le modèle opérationnel de l'ISA-95 plonge plus profondément dans la description interne d'un système MES. Il a été particulièrement développé dans la partie 3 du standard ISA-95. Sans être un guide de conception pour la réalisation d'un logiciel de MES, il est une base de réflexion utile, en même temps qu'élégante et novatrice, sur les fonctions qui composent un MES. Dans la partie 3 du standard ISA-95, ce modèle est décrit comme modèle d'activité, il s'agit du modèle décrit figure 5.

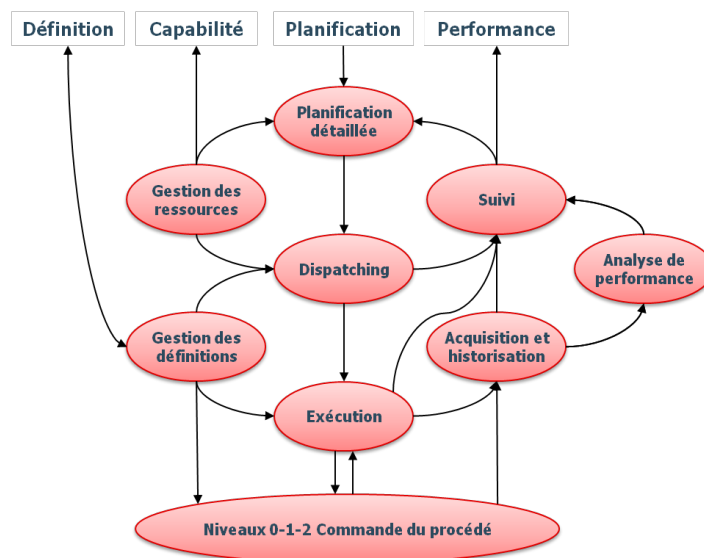


Figure 5 - Modèle opérationnel d'un système MES (source ORDINAL Software, d'après l'ISA)

Les objets de *Définition*, *Capabilité*, *Planification*, *Performance* sont les objets d'échange avec le niveau 4 (ERP/GPAO), sur lesquels nous reviendrons dans le paragraphe suivant. Dans le principe, un ordre de fabrication issu de l'ERP va faire l'objet d'une *Planification détaillée*, puis d'un *Dispatching* en tâches sur la base de la *Gestion des ressources*. Les tâches planifiées vont passer séquentiellement ou parallèlement dans les opérations d'*Exécution*, lesquelles prendront en compte la *Gestion des définitions* (recettes, procédures, gammes) des produits à fabriquer. Ces dernières opérations interagissent directement avec la *Commande du procédé*, que celle-ci soit manuelle ou automatisée (supervision, contrôle-commande).

Dans la partie droite du modèle s'effectuent les opérations liées à la traçabilité et à l'optimisation du procédé. Elles commencent (en partant du bas) par l'*Acquisition et historisation*, qui va permettre la collecte de données nécessaire aux autres opérations. Suivant l'objectif d'utilisation de ces données, on s'orientera vers l'*Analyse de performance* ou vers le *Suivi* (dossiers de lots, suivi des flux matières, généalogie des produits), qui vont fournir au niveau 4 des informations sur la performance de production. Certaines opérations d'analyse permettent d'ajuster les opérations de planification (le suivi de production, par exemple, permettra de modifier la planification détaillée).



Outre son élégance et sa facilité de compréhension, l'une des originalités de ce modèle est qu'il s'agit en fait d'un *méta-modèle* applicable à plusieurs domaines. En effet, si nous venons de l'appliquer naturellement à la Production, on peut tout aussi bien l'appliquer à la Maintenance (planification des ordres de travaux, dispatching, exécution, suivi, ..), à la Qualité, ou la Gestion des transferts de stocks. Adopter un même modèle pour ces différents domaines est susceptible de faciliter la compréhension du système par les exploitants, d'améliorer la courbe d'apprentissage des logiciels et dans certains cas, de mettre en commun des composants métier de logiciels MES.

### ***Les échanges avec le niveau 4 du CIM (ERP/GPAO)***

C'est la partie interface du modèle opérationnel de l'ISA-95, essentiellement décrit dans la partie 2 du standard ISA-95.

Suivant le cas, les échanges se feront essentiellement depuis l'ERP ou niveau 4 vers le MES, depuis le MES vers l'ERP, ou dans les deux sens.

La partie **Définition** comprend les recettes, gammes et procédures de fabrications des produits. Généralement transmises par l'ERP vers le MES, ces informations demandent en tout état de cause à être précisées au niveau de ce dernier, car il est rare que les informations précises de production (réglages machines, détail des étapes de la gamme, conditions d'exécution, ..) soient présentes au niveau de l'ERP. De ce fait, beaucoup d'industriels font le choix de redéfinir complètement les produits dans le MES, ou dans un logiciel dédié à cette fonction (gestion de gammes).

La partie **Capabilité** est transmise par le MES vers l'ERP ou niveau 4. En théorie, elle fournit à ce dernier les informations d'occupation de la capacité de production de l'unité de fabrication. Ces données sont susceptibles d'améliorer la planification faite par l'ERP. Dans la pratique, les ERP capables d'exploiter ce type d'information de manière automatisée et les systèmes MES capables de fournir ces informations sous une forme simple à exploiter sont rares. La conséquence est que la planification s'effectue plus en fonction d'une expérience antérieure codifiée sous forme de règles que d'une exploitation de la disponibilité effective de l'unité de production.

La partie **Planification** est transmise de l'ERP vers le MES. Elle est indispensable et c'est de loin le flux de données le plus utilisé. Aujourd'hui, la plupart des industriels exigent que la planification « descende » de l'ERP et « interdisent » que des ordres de fabrication soient créés directement dans le MES. Dans la pratique, cette possibilité est néanmoins souvent prévue en cas d'indisponibilité de l'ERP (maintenance informatique par exemple). On procède alors à une régulation a posteriori.

La partie **Performance** est transmise du MES à l'ERP ou niveau 4. Il s'agit de la performance au sens anglo-saxon du terme, c'est-à-dire à la fois ce qui est produit et la performance de production (au sens français). Les bilans de production peuvent être transmis journalièrement, par demi-journée, ou à la fin d'exécution de chaque ordre de fabrication. Parfois, il n'y aura pas de véritable transmission informatique et le MES éditera directement un état imprimé ou électronique qui sera joint aux documents de fabrication.

### ***La fusion ISA-95 / ISA-88***

Antérieur au standard ISA-95, le standard ISA-88 a un scope qui recouvre en partie celui de l'ISA-95, mais est limité aux procédés semi-continus ou batch. Il vise essentiellement à modéliser ce type de procédés, dans le but d'augmenter la souplesse d'exécution et la capitalisation des investissements en automation. Cet objectif est recherché par une répartition innovante des traitements entre des parties

d'automatismes peu modifiables et une partie informatique de pilotage supportant un langage simple de construction des séquences d'opérations (langage Procedure Function Chart ou PFC). Le rapprochement du standard ISA-88 du standard ISA-95 présente au moins deux intérêts :

- Du fait qu'il se limite aux procédés dits batch, l'ISA-88 offre une modélisation plus précise des niveaux inférieurs de la hiérarchie des équipements, à savoir les équipements des niveaux Process Cell et Unit. Pour ce qui est de la définition des produits, les structures proposées par l'ISA-88 sont plus précises, et offrent un langage de description (PFC), que ne propose pas l'ISA-95 de manière générale. Etendre cette approche aux autres procédés (continus et discrets) n'est pas hors de portée, et plusieurs experts de l'ISA-88 considèrent cette approche comme prometteuse.
- En conséquence, l'ISA-88 fournit une modélisation plus détaillée, et plus proche de l'implémentation effective, de l'exécution d'un procédé. On peut considérer qu'il s'agit là d'une contrainte. Mais à l'inverse c'est aussi une piste sérieuse pour une standardisation qui garantit par exemple une meilleure interopérabilité entre les logiciels de MES et les automatismes.

Par ailleurs, plusieurs membres du comité de rédaction de l'ISA-88 ont également participé au comité de l'ISA-95. De ce fait, les concepts et même les objets issus de ces travaux ne sont pas totalement étrangers, loin de là. Finalement, un groupe de travail s'est constitué en vue de rapprocher les deux standards. Le fruit de ce travail a été en particulier une définition compatible entre les deux standards de la hiérarchie des équipements.

### ***Intégration verticale et standardisation***

Nous l'avons vu, l'intégration verticale, tant avec les ERP vers le niveau 4, qu'avec la supervision et les automates vers le niveau 2, est essentielle. Dès son origine en fait, l'ISA-95 s'est en effet concentrée sur la définition des objets d'échange entre le niveau 3 et le niveau 4. Plus récemment, des travaux significatifs de standardisation ont également été réalisés en ce sens, favorisés par l'avènement des technologies XML, beaucoup plus aptes que les anciennes méthodes (échange de fichiers ASCII essentiellement) à l'échange d'objets.

Pour ce qui est des échanges entre niveau 3 et niveau 4, ces travaux ont débouché sur la définition du langage B2MML (pour Business to Manufacturing Markup Language) de plus en plus employé, et supporté par des ténors de l'ERP comme SAP.

Pour ce qui est des échanges entre le niveau 3 et le niveau 2, la situation est un peu moins claire, mais paradoxalement plus standardisée en termes d'implémentation. En effet, le standard OPC, initialement promu par les constructeurs d'automates et Microsoft pour l'interface des outils de supervision, est aujourd'hui un standard ultra dominant pour les échanges de ce type. La communication entre niveau 3 et niveau 2 utilise donc tout naturellement ce standard, même si dans la pratique, les « objets » échangés restent de très bas niveau (booléens, entiers, flottants) avec simplement des informations complémentaires d'horodatage et de qualité. Dans les évolutions récentes du standard OPC (OPC-UA), on note une prise en compte du langage XML, qui ouvre la voie à une approche à la fois plus « objet » et moins dépendante de la plateforme Windows (OPC signifiait à l'origine OLE for Process Control, et est resté dépendant des technologies COM-DCOM de Microsoft).

## Typologie des logiciels de MES

### ***L'importance des procédés ciblés***

Comme nous l'avons vu précédemment, et comme cela se traduit même au sein du standard ISA-88, les procédés ciblés jouent un rôle très important. En effet, autant une entreprise admettra facilement que sa comptabilité, et même sa gestion financière relèvent de bonnes pratiques très générales, et permettent l'application de solutions logicielles communes à pratiquement toutes les entreprises, autant ces mêmes entreprises considèrent généralement leurs procédés de fabrication comme uniques, et nécessitant une analyse et des solutions qui leur soient propres.

Il est d'ailleurs vrai, comme le montrent les travaux de l'ISA-95, que les trois grands types de procédés vont induire des contraintes très différentes en termes d'optimisation de la production. Pour un procédé continu, la maîtrise des paramètres du procédé est essentielle, on cherchera à identifier la moindre dérive et les outils mathématiques de suivi des étapes du process seront souvent très poussés. Par contre, les contraintes de reconfiguration des installations, de fréquence de changement des réglages, sont faibles. Pour un procédé discret, ce sont les cadences et les arrêts de production qui vont occuper toute l'attention, ainsi que les contraintes d'occupation des machines et du personnel, et l'identification des goulots d'étranglement. Suivant leurs types, les procédés semi-continus ou batch vont cumuler les préoccupations des deux types de procédés précédents à un degré plus ou moins important. Rappelons qu'un procédé semi-continu décompose une production en une série de lots élémentaires ou « batch » correspondant généralement à une des capacités limitantes de l'installation (cuve de réaction ou de mélange par exemple), d'où leur nom. Ces procédés disposent toutefois d'un avantage important, qui est que l'analyse peut se faire sur une unité de traitement, correspondant pour le procédé à la capacité d'un « batch ». Cette restriction de l'analyse autorise des stratégies d'optimisation puissantes, grâce à l'utilisation d'un « moteur d'exécution » (cf. la recherche de ce que l'on appelle le « golden batch »).

### ***Approche spécifique (métier)***

Compte tenu de l'attente des industriels quant à la connaissance précise de leur procédé, il est logique que l'approche la plus courante pour la réalisation d'un *Manufacturing Execution System* ait été initialement une approche spécifique, généralement confiée à un prestataire intégrateur connaissant parfaitement le métier de l'industriel. Généralement, l'industriel rédige donc un cahier des charges exprimant ce qu'il attend d'un logiciel de MES et demande au prestataire d'en chiffrer la réalisation. Cette approche, la plus largement répandue dans les premières années de mise en place de solutions MES, est aujourd'hui en recul très net car elle présente de nombreux inconvénients :

- Il est difficile pour l'industriel de hiérarchiser ses besoins. Idéalement, les fonctions demandées devraient être hiérarchisées en fonction d'une part des enjeux principaux de l'industriel, d'autre part du retour sur investissement attendu de la mise en place de ces fonctions. Malheureusement ce dernier est rarement connu avant la mise en place du MES, et le chiffrage analytique du coût de chaque fonction est rarement réalisé. De ce fait, l'industriel risque fort d'avoir de lourdes charges de réalisation pour des fonctions qui en définitive ne lui apporteront que peu.
- Si l'industriel peut connaître ses besoins actuels, il lui est difficile d'anticiper ses besoins futurs. Or tous les besoins qu'il n'aura pas exprimés dans son cahier des charges se traduiront dans l'avenir,

dans le meilleur des cas comme des surcoûts significatifs pour son projet, dans le pire des cas par une remise en cause des développements déjà réalisés.

- Une solution spécifique est difficile à pérenniser, particulièrement en regard de l'évolution de l'informatique en général. Il est impossible pour le prestataire d'intégrer cette pérennisation à son offre car elle reviendrait à doubler ou à tripler son montant. Pourtant l'informatique évolue et l'industriel se verra un jour ou l'autre confronté à une barrière technologique majeure : fin de la disponibilité ou du support du système d'exploitation requis, disponibilité et support des environnements de développement retenus, ...etc. Sans parler des coûts entraînés si l'industriel souhaite bénéficier de technologies plus récentes (intranet, solutions mobiles, ...etc.) qui n'auraient pas été prévues dès le départ.
- L'industriel est très fortement dépendant du prestataire ayant réalisé le développement. Il devra avoir recours à lui pour toute extension. Sans envisager des situations critiques (disparition du prestataire, mésentente commerciale), l'indisponibilité temporaire du prestataire peut être un handicap pour le développement optimal de l'industriel.

### ***Assemblage de logiciels différents***

La couverture du MES est large, et il n'est pas facile de trouver au sein d'une même offre la réponse à l'ensemble des besoins exprimés par l'industriel. Plutôt que d'adopter une approche d'ensemble du système, certains logiciels se sont spécialisés dans une ou plusieurs des fonctions du MES. Cet état de fait s'explique mieux si l'on considère que les fonctions couvertes par le MES existaient auparavant de manière isolée. Il est finalement assez courant de concevoir la solution MES comme un assemblage de logiciels plus spécialisés, comme on pourra le voir par exemple dans la figure 6.

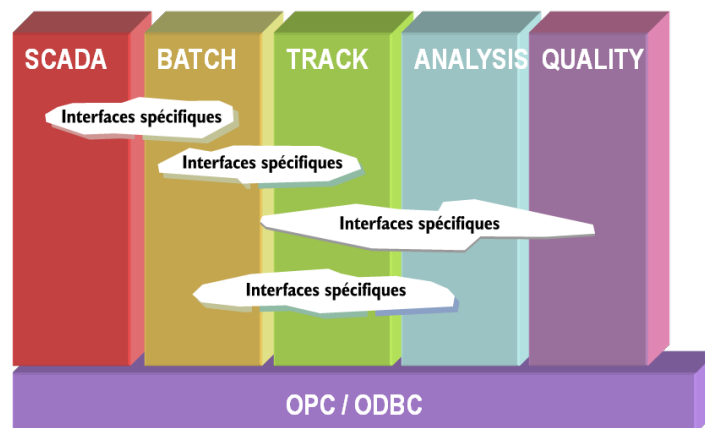


Figure 6 - Constitution d'une solution MES par assemblage (source ORDINAL Software)

Par rapport à une solution entièrement spécifique, cette solution apporte des bénéfices :

- Expertise des éditeurs de logiciel des briques constitutives permettant généralement de réduire le coût total de mise en œuvre pour les besoins actuels et permettant d'anticiper les besoins futurs
- Dans le cas où les différents logiciels sont fournis par un même éditeur, une meilleure pérennité de la solution en regard de l'évolution des technologies

Toutefois, même dans ce dernier cas, plusieurs inconvénients subsistent :

- Cette approche est en contradiction avec la notion d'un véritable système, où les différentes fonctions interagissent fortement entre elles, comme cela a été développé durant tout le début de cet exposé. L'interaction est limitée à quelques interfaces spécifiques de dialogue, et à des couches basses sans apport métier.
- Vu de l'exploitant, la solution n'est pas homogène. Les interfaces des différents modules ont en effet été conçues par des équipes logicielles différentes. Cela induit des courbes d'apprentissage de la solution plus longues. L'accès à l'outil peut aussi être inégal suivant les modules (intranet, mobiles, ...etc.)
- Conçus séparément, les différents modules ont chacun leur propre modèle de représentation. Outre que cela occasionne une multiplicité des saisies pour la définition du système, les différents modèles ne sont pas nécessairement cohérents sur le plan analytique, ce qui est source de difficultés encore plus importantes.
- Les interfaces spécifiques constituent un point de faiblesse en termes de pérennité et d'évolutivité des applications. Même fournis par un même éditeur, les différents modules évoluent en effet de manière indépendante et obligent à revoir les interfaces lors des mises à jour, quand certaines ne sont pas à reprendre complètement.
- La réponse « en standard » aux besoins exprimés par l'industriel n'est jamais complète, et cet assemblage n'offre pas d'autre issue pour les couvrir que des développements complémentaires entièrement spécifiques.

### ***Approche hybride***

Ce dernier constat ouvre en fait la voie à une autre approche, assez courante également, qui consiste à utiliser quelques briques logicielles existantes, et de les intégrer dans un développement logiciel spécifique. Cette approche est essentiellement guidée par le souci légitime de « ne pas redévelopper ce qui existe déjà », mais finalement n'en retire que peu d'avantages, tout en cumulant les inconvénients des approches précédentes.

D'une part en effet, les évolutions des briques standards entraîneront inévitablement des modifications du développement spécifique, sur lequel le (ou les) fournisseurs des briques ne pourront apporter leur concours. Et le développement spécifique, de son côté, a une trop grande importance dans la réalisation pour que sa maintenance ne pose pas de problèmes à terme.

Assez paradoxalement, la brique « supervision » (ou SCADA) sera très souvent retenue dans les briques standard constitutives de ce type de solutions. Cela peut sembler paradoxal, car on ne peut pas dire que la supervision soit au cœur du MES. Bien qu'elle joue un certain rôle dans l'exécution, la supervision est même souvent considérée comme de niveau 2. On peut voir une double explication pratique à cet apparent paradoxe. En premier lieu, les superviseurs ont longtemps été les seuls outils permettant la réalisation assez simple d'interfaces utilisateur, en particulier d'interfaces permettant des animations en temps réel ; en second lieu, les logiciels de supervision sont le plus souvent dotés d'un langage de script (souvent Visual Basic) simple d'accès, qui permet d'engager des développements d'extension progressifs sans vraiment prendre conscience que l'on s'engage dans le développement spécifique. Le bilan de ce choix peut toutefois s'avérer lourd : l'application de MES finale est un développement spécifique conséquent, difficile à maintenir, et entièrement dépendant d'une brique logicielle qui n'est pourtant pas majeure sur le plan fonctionnel.

## **Progiciels intégrés**

Les progiciels intégrés constituent la réponse la plus logique et la plus pérenne pour la réalisation d'une application de MES. D'une certaine manière, cette réponse est similaire au type de réponse apportée par les ERP au niveau de la gestion de l'entreprise. Le progiciel intégré de MES n'est autre que l'ERP de l'usine. Citons quelques avantages des progiciels intégrés de MES, à comparer aux limitations des autres approches :

- Offre proposant un véritable système, où les différentes fonctions du MES interagissent fortement entre elles, et partagent un même modèle de définition
- Synergie sur les développements, permettant à l'industriel de bénéficier d'analyses et de fonctionnalités déjà mises en œuvre pour son métier, ou communes à plusieurs métiers. Les fonctionnalités offertes permettent également de mieux anticiper les besoins futurs.
- Homogénéité des interfaces utilisateur proposées aux exploitants, et des types d'accès offerts (Intranet, mobiles, ...etc.)
- Garantie par l'éditeur du logiciel de l'interopérabilité entre les différentes fonctions sans charges récurrentes d'adaptation des interfaces
- Prise en charge par l'éditeur des évolutions technologiques, en particulier celles liées aux systèmes d'exploitation, ou aux bases de données utilisées.
- Réduction de la dépendance de l'intégrateur par l'emploi d'un progiciel standard supporté par plusieurs prestataires

L'intégration suppose une bonne connaissance des problématiques auxquelles l'industriel est confronté. Il n'est donc pas étonnant que les progiciels intégrés soient souvent spécialisés sur un métier, ou sur un domaine (comme l'agroalimentaire, l'automobile, ou l'électronique). Toutefois la référence à des standards comme l'ISA-95 permet une applicabilité assez large de ces logiciels.

La crainte la plus souvent exprimée par les industriels vis-à-vis des progiciels intégrés est qu'une partie, parfois réduite mais essentielle pour eux, des fonctionnalités qui leur sont nécessaires ne soit pas couverte par le progiciel, et que leur développement ne s'avère impossible ou prohibitif. Cette crainte est légitime, car il serait présomptueux de penser que le domaine du MES est suffisamment mature pour que tous les besoins soient couverts. De ce point de vue, les progiciels intégrés bénéficiant d'une structure de *plateforme logicielle*, et permettant de réaliser des extensions logicielles bénéficiant des capacités de la plateforme (accès au modèle d'objets, réalisation d'IHM, capacités de déploiement, ...) permettent de répondre à leurs attentes dans les meilleures conditions.

## **Intégration MES et Supervision**

Comme nous l'avons vu plus haut, la supervision se trouve en quelque sorte à la périphérie du MES, et il est courant de la considérer comme appartenant au niveau 2. Pour la plupart des industries de fabrication, ce rattachement au niveau 2 (automatismes) plutôt qu'au MES n'est pas vraiment justifié, car l'exécution est essentiellement pilotée par le superviseur, et il n'est pas rare que les notions de lancement et même d'ordre de fabrication y soit gérées. Tout simplement, le besoin aidant, le scope de la supervision s'est étendu grâce aux langages de script pour couvrir de plus en plus de fonctions du MES.

Si les outils de supervision et de MES restent séparés, cela débouche inmanquablement sur une double définition des données qu'ils manipulent tous les deux, comme les recettes, les équipements ou le personnel. D'autant que supervision et MES utilisent le plus souvent les mêmes données d'entrée en provenance des automates.

L'intégration de la supervision et du MES favorise assurément une mise en œuvre plus rapide du MES, dont l'installation peut être envisagée comme conjointe au pilotage direct de l'installation. Elle économise de plus de nombreuses charges d'interface, et permet une capitalisation plus forte des développements, en particulier dans un contexte multi-site : des modèles comme les modèles d'équipements peuvent en effet intégrer visuels de contrôle-commande, visuels d'exécution et de pilotage, fonctions de suivi des flux matières, et d'analyse de performance.

Cette intégration répond aussi au souhait de simplicité et d'intuitivité exprimé par la plupart des exploitants, dont le manque peut aussi être un frein à la mise en place d'un MES.

## Le déploiement d'un MES

### *Un projet informatique*

Cela peut bien sûr paraître une lapalissade, mais un projet MES met en œuvre pour l'essentiel le développement et/ou la configuration d'un ou plusieurs logiciels sur une architecture adaptée. C'est donc un projet informatique. Cela vaut toutefois la peine de le rappeler. En effet, dans les usines, le développement logiciel n'a pas toujours été considéré avec la même rigueur que dans les départements « IT » des entreprises. Deux raisons existent à cela. La première est que les budgets alloués pour le développement logiciel en usine, quand ils existent, sont souvent faibles. La seconde est plus pernicieuse. En effet, la convivialité apportée dans le paramétrage des logiciels de supervision, alliée à des langages de script intuitifs, a conduit à confier des tâches de développement de plus en plus importantes à des automaticiens maîtrisant simplement le langage du superviseur, sans réelle mise en place de structure de projet, avec pour résultat des milliers de lignes de programmation sans commentaires, documents d'analyse ou de conception.

Nous prendrons comme exemple pour le déploiement d'un MES un projet s'appuyant sur la plateforme intégrée et modulaire COOX d'ORDINAL Software. S'agissant d'un progiciel intégré de MES, il évitera (ou limitera considérablement le périmètre exigeant) la mise en place des méthodes les plus lourdes de développement logiciel spécifique (cycle en V complet).

Pour un projet MES de ce type, il faudra toutefois s'assurer que les conditions, étapes suivantes et documentations soient bien présentes :

- Chef de projet
- Cahier des charges
- Planning prévisionnel et outils de suivi
- Document de spécification
- Documents d'analyse (analyse fonctionnelle, modélisation,...etc.)
- Documents de conception (pour des fonctions non délivrées en standard)
- Documents de recette

## **Technologies mises en œuvre**

Les technologies mises en œuvre peuvent bien sûr être très diverses suivant le progiciel choisi. La plateforme COOX est une plateforme intranet native, s'appuyant sur les technologies Java. Elle ne présente donc quasiment aucune limitation en termes de machines pour le déploiement. Les serveurs seront généralement des processeurs PC multi-cœurs, sous Windows Server ou sous Linux. Il est possible également de mettre en œuvre des architectures virtualisées. Les clients seront eux aussi des PC, parfois des PC industriels, mais parfois aussi des équipements mobiles (tablette PC en wifi par exemple).

La plateforme COOX supporte une architecture de services distribués (SOA) permettant de répartir les services MES sur plusieurs machines. Une architecture COOX pourra par exemple comprendre, outre le noyau de services ISA-95 intégré à la plateforme, un ou plusieurs services d'accès à des bases de données (gestion de la réplication), le service web, le service de gestion centralisée de la sécurité, un service OPC (éventuellement redondé), des services de communication spécifiques, ...etc.

Dès lors que la machine client supporte la technologie Java, aucune installation supplémentaire n'est nécessaire. Toutes les technologies Intranet et Internet étant présentes de manière native, il sera possible par exemple, aux exploitants d'un site d'avoir accès aux informations d'un autre site (s'ils y sont autorisés bien sûr).

Le paramétrage du logiciel s'effectue à l'aide d'un environnement de développement d'applications appelé *Builder*. Bien que son abord reste simple, et en grande partie compréhensible par des non-informaticiens, ses capacités sont très étendues et couvrent également le développement d'extensions en JavaScript et en Java. Des développements plus ambitieux utilisant les ressources de la plateforme (modèle objet compatible ISA-95, services déployés, ...etc.) sont également possibles en utilisant l'environnement de développement informatique open source *Eclipse*.

## **Compétences requises**

Les compétences requises pour le paramétrage et le développement varient au fur et à mesure que l'on s'écarte des fonctions standard proposées par le logiciel. Le paramétrage des fonctions standard ne requiert en effet aucune compétence de développement informatique et peut être entièrement maîtrisé au travers d'une formation (modulaire suivant les modules effectivement mis en œuvre). La formation permet de maîtriser l'utilisation du *Builder* déjà cité.

Des extensions et des développements plus conséquents vont s'adresser à des profils plus informatiques, tout en sachant que les langages de script comme le JavaScript restent très accessibles. La connaissance du Java sera requise pour des développements de nouvelles fonctions, ce que permet la plateforme.

## **Facteurs de succès et causes d'échec**

Nous nous limiterons aux facteurs de succès, les causes d'échec n'étant, le plus souvent... que l'absence d'un au moins des facteurs de succès.

En premier lieu de ces facteurs de succès, nous citerons la présence d'un chef de projet **dédié** au projet MES chez l'industriel utilisateur. Trop souvent en effet, la direction industrielle nomme « pour la forme » un chef de projet MES qui doit remplir une toute autre fonction opérationnelle (responsable d'exploitation, responsable maintenance). Or le rôle de Chef de projet MES est, au moins dans les



premières étapes, et souvent jusqu'à la fin de la première mise en place, un rôle à temps plein, même si la réalisation et la mise en place du MES est confiée à un prestataire. En effet, le recueil des besoins, leur hiérarchisation, la validation des spécifications et le suivi du déroulement du projet vont exiger une énergie considérable, en même temps que la conduite du changement auprès des équipes de production concernées.

En second lieu vient la **hiérarchisation** des besoins. C'est une clé des plus déterminantes. Sans hiérarchisation des besoins, la plupart des projets de MES entraîneront un budget prohibitif et ne verront jamais le jour. La collecte des besoins auprès des utilisateurs débouche sur la fameuse « lettre au Père Noël » sans que l'utilité réelle de chacune des suggestions (et parfois même leur contradiction mutuelle) soit évaluée. On pourra sans doute éliminer les moins critiques. Il en restera sans doute encore beaucoup, et une hiérarchisation des besoins permettra d'envisager une réalisation par étapes. Au-delà d'un étalement du budget, toujours plus facile pour une entreprise, il est possible que les gains engendrés par la mise en place de la première étape puissent financer les suivantes. Un conseil : choisir une première étape où la couverture en standard du progiciel de MES est forte, comme l'exécution des fabrications avec dossier de lots, ou l'analyse de performance des machines. Un tel choix peut amener un retour sur investissement rapide.

Si cette hiérarchisation n'est pas suffisante, ou si la direction n'est pas suffisamment convaincue pour donner son accord définitif à un projet global, une bonne pratique est la **réalisation d'un pilote**. Le pilote permettra, sur une partie de l'installation, de valider les fonctionnalités clés et leur retour sur investissement. C'est également une bonne méthode pour la **conduite du changement** auprès du personnel, qui est elle aussi une démarche indispensable. Des personnalisations judicieuses au niveau du pilote, et le temps d'adaptation qu'il amène, vont favoriser l'adhésion du personnel au système.

Enfin un point majeur est la formation du prestataire (si possible validée par un niveau de certification de l'éditeur) et son **accompagnement par l'éditeur**. Un éditeur disponible et réactif, aidant le prestataire à identifier les « points durs » du projet et les solutions qui peuvent être apportées, est un atout sérieux pour la maîtrise des risques du projet de MES.

### ***Capitalisation et « modèle cœur »***

Beaucoup de domaines font actuellement l'objet de consolidations importantes d'entreprises. Des synergies en sont attendues, en particulier dans le domaine de la production et donc du MES. Idéalement, plusieurs sites fabriquant les mêmes produits ou des produits similaires devraient non seulement pouvoir utiliser le même progiciel de MES, mais avoir des temps de développement et de mise en place de leur système considérablement réduits, dans la mesure où un modèle de référence Core Model ou « modèle cœur » a déjà été validé.

Dans la pratique, cette vision idéale a souvent été mise en défaut. Pour qu'elle soit possible, il faut en effet que le progiciel de MES ait réellement la possibilité de capitaliser le savoir-faire de l'industriel au sein de ses modèles. La plateforme COOX permet cette capitalisation grâce à deux choix technologiques majeurs : l'adoption au sein même de son architecture du modèle objet et du modèle opérationnel de l'ISA-95, et une technologie de modèles personnalisables permettant de définir les caractéristiques comportementales et les interactions avec les opérateurs des équipements de l'installation.

Moyennant une réflexion suffisante en amont, il est possible de définir une bibliothèque de modèles réutilisables dont la mise en place pour le déploiement de chaque nouveau site sera considérablement

réduite. Cette stratégie a par exemple été mise à profit dans le domaine des silos céréaliers, où un nouveau déploiement est désormais possible en quelques semaines.

Pour un maximum de bénéfices de l'approche *Core Model*, l'industriel devra étendre sa réflexion au-delà du domaine de l'informatique industrielle et y intégrer aussi d'autres domaines comme sa stratégie d'achats en termes d'équipements : le choix d'équipements permettant des économies d'achat immédiates, mais ne présentant pas le même profil fonctionnel, et donc entraînant la création de nouveaux modèles pour l'exécution peut par exemple s'avérer d'un coût global supérieur.

## Conclusion

Les premiers travaux du MESA (et donc le terme de MES) datent du début des années 90, soit près de vingt ans. En vingt ans, la connaissance de ce domaine, très floue dans les premières années, a beaucoup progressé. La question ne se pose plus vraiment de l'intérêt d'un MES mais du choix de la meilleure solution pour le mettre en œuvre.

Comme dans beaucoup de domaines de l'informatique, la « progicielisation » est un phénomène quasiment inéluctable (qui envisagerait aujourd'hui le développement « sur mesure » de sa comptabilité ? C'était pourtant la règle il y a une trentaine d'années). L'apport de l'ISA-95 a été considérable pour donner les bases de la réalisation de progiciels intégrés de MES pouvant garantir à l'industriel une couverture fonctionnelle suffisamment large et une réponse potentielle à ses besoins futurs.

Sur un angle plus technologique, les solutions intranet, les architectures de services distribués, permettent une adaptation plus aisée à la taille des sites. Une conception de plateforme logicielle favorise également les extensions éventuelles. La capitalisation des développements est aussi un enjeu majeur, non seulement par les réductions de coûts qui peuvent être engendrées dans un déploiement multi-sites, mais par la capacité qu'elle donne à l'industriel de déployer de nouvelles usines plus rapidement, capacité essentielle sur les marchés émergents.

Mais un projet de MES est avant tout un projet multidisciplinaire, qui met en jeu le plus souvent les équipes de production, la gestion de l'entreprise et la direction informatique, et à cet égard la conduite du projet et la conduite du changement sont déterminantes.

**Note** : une présentation complète du MES, ainsi qu'un projet de déploiement ont également été exposés dans une publication en deux articles des *Techniques de L'Ingénieur* (n° S 8005 et suivants), dont sont extraites certaines parties de ce livre blanc.