

L’IDM de demain : un accès facilité, un usage intensif pour des performances accrues

David Bihanic, Jean-Michel Bruel, Philippe Collet,
Benoît Combemale, Sophie Dupuy-Chessa, Xavier Le Pallec, Thomas Polacsek

1^{er} novembre 2014

1 Contexte

Pour beaucoup de chercheurs en Génie Logiciel, l’Ingénierie Dirigée par les Modèles (IDM) est maintenant largement intégrée à l’ingénierie logicielle. Pourtant, en y regardant de plus près, la réalité semble différente. Dans bon nombre de domaines, comme les Interfaces Homme-Système (IHS) ou l’ingénierie des besoins, les travaux qui adoptent une démarche dirigée par les modèles se limitent souvent à l’abstraction comme seule dimension de modélisation. Industriellement, on observe des gains par l’application de l’IDM, mais plus par la seule explicitation des abstractions (documentation) que par les autres bénéfices attendus (maîtrise de la complexité, génération d’applications). La communauté de l’IDM a pourtant réalisé un certain nombre d’avancées significatives sur des aspects complexes, mais spécifiques comme notamment la composition, l’adaptation, le temps-réel ou bien encore le passage à l’échelle.

2 Problématique

Cependant plusieurs problèmes cruciaux persistent du point de vue de l’usage des modèles.

Ergonomie. Une part limitée des travaux et outils liées à l’IDM prennent en compte les processus cognitifs généralement à l’œuvre lors de la conception et le développement. Le non-alignement est donc fréquent et entraîne généralement des processus cognitifs supplémentaires chez le concepteur pour adapter les modèles et/ou leurs outils à son contexte de travail. Cela implique malheureusement une augmentation de l’effort à fournir, du nombre d’erreurs possibles et une diminution de la vitesse de travail. Cet impact important sur l’utilisabilité des modèles et outils est un frein important à l’adoption de l’IDM et à son passage à l’échelle.

Design. Si la syntaxe visuo-graphique/diagrammatique classiquement usitée pour décrire ou représenter les modèles répond pour partie aux attentes et visées fonctionnelles (qualités opératoires), reste que celle-ci n’est pas suffisamment efficace. En effet, les apories et difficultés de traitement ‘utilisateur/concepteur’ sont aujourd’hui nombreux. Aussi, la contribution du design peut s’avérer d’un aide précieuse en vue d’inventer, de concevoir de nouveaux langages de représentation et de visualisation graphique des modèles (cf. data design ou design de données). Ces langages, ouvrant à une réelle plus-value de traitement, seraient également en capacité de dénouer nombre de problèmes ou écueils d’ordre logique et conceptuel auxquels

doivent faire face les utilisateurs, concepteurs et opérateurs. A noter que cette contribution, touchant ici à la représentativité des modèles (langage visuo-graphique), appelle à être complétée sur le plan des interfaces.

Multiplicité. Les applications courantes et futures nécessitent que les différentes sources de complexité soient maîtrisées en même temps (personnalisation et réactivité, accès aux non-informaticiens, prédictions et analyse, prise en compte des points de vue et des niveaux de compétences, ...).

Ces problèmes entravent grandement l'adoption de l'IDM dans de nombreux contextes industriels. L'IDM peut-elle tomber aux oubliettes comme ce fut le cas des outils CASE, car certains aspects de l'approche furent totalement occultés? Une décennie riche en résultats scientifiques sera-t-elle perdue car l'IDM n'est pas assez facile d'accès, ne supporte pas l'usage intensif du génie logiciel actuel?

3 Challenges identifiés

Partant de ce constat, un certain nombre de challenges peuvent être identifiés en croisant les considérations sociétales et cognitives avec des défis à plus longue échéance, comme illustré par la table 1. Ces défis à long terme sont issus d'un travail de réflexion à l'échelle internationale sur les grands problèmes à résoudre par la communauté IDM dans le cadre de l'évolution de notre société¹. Quatre points de ruptures précis sont ainsi obtenus.

Représentativité. Le caractère de plus en plus trans-diciplinaire des applications et des systèmes doivent nous inciter à, non seulement anticiper les difficultés techniques, mais aussi profiter de cette richesse. Notre discipline reste trop souvent cantonnée à l'ingénierie des logiciels ou des systèmes sans s'intéresser à ce que peuvent apporter des disciplines comme l'intelligence artificielle, le design, le web sémantique, mais aussi la médecine, les sciences cognitives, sociales, etc. Les efforts actuels sur la fusion de modèles doivent intégrer cette dimension multi-diciplinaire. Cela exige aussi à repenser les formalismes de représentation, de description et de visualisation des modèles tout autant que les mécanismes utilisés en IDM pour les notations visuelles. Ces dernières sont souvent limitées aux réseaux - terme employé en sémiologie graphique -, alors que l'espace de conception graphique est beaucoup plus riche.

Accessibilité. Le logiciel est devenu quelque chose de palpable dans le grand public depuis l'avènement des *AppStore* et autre *GooglePlay* où les applications s'installent, se partagent, s'évaluent de manière intuitive et abordable au plus grand nombre. L'étape suivante sera de faciliter le développement de ces logiciels. L'IDM prône de placer les modèles au cœur des développements. À nous de rendre l'utilisation, la manipulation et la réalisation de modèles accessibles au plus grand nombre. Ceci implique aussi des modes d'interaction efficaces et intuitifs. Les notations utilisées devront mettre en avant les informations pertinentes pour la préoccupation du concepteur lambda et donc exploiter au maximum les mécanismes visuelles répertoriés (associativité, sélection, imposition, abstraction. . .) afin de maîtriser la complexité

1. Gunter Mussbacher et. al. *The Relevance of Model-Driven Engineering Thirty Years from Now*. In IEEE/ACM Int. Conf. MoDELS 2014, Valence.

des modèles logiciels. Les périphériques d'interaction utilisés par les outils d'édition ne sont pas naturels au regard de la modélisation : un travail important est aussi à fournir afin de créer/déterminer des périphériques adaptés.

Flexibilité. Les modèles doivent être à l'image des applications modernes : flexibles pour faciliter leur manipulation et intégration de manière multiple et dynamique. Il s'agit de pouvoir tenir compte, dans un domaine d'applications donné, de toutes les préoccupations, techniques ou relatives au domaine, avec plus de confiance et de prévisibilité dans le résultat d'une intégration. Ceci doit être aussi valable dans un contexte dynamique, pour s'adapter à la volée aux changements de leur environnement (outils, besoins utilisateurs, interfaces, ...). Ceci pourrait être facilité par l'utilisation de modèles de référence, facilement customisable et intégrable. Par ailleurs, si les informations contenus dans un modèle ne sont pas en phase avec le reste du projet, cela exige un travail cognitif important du concepteur pour réactualiser mentalement le modèle. La désynchronisation entre le code et le modèle est un exemple récurrent en IDM. Ce travail cognitif devrait être facilité par des environnements adaptés .

Evolutivité. Un modèle n'est jamais aussi riche que la réalité. Il est donc par définition incomplet. Néanmoins, le modèle d'un système complexe se doit souvent d'être aussi complet que possible, notamment si on veut pouvoir effectuer des analyses et prédictions de propriétés. Les outils de modélisation se sont donc souvent intéressés au caractère complet, formel des modèles. Ceci est souvent perçu comme un frein à la créativité et à l'activité même d'abstraction. Il nous semble que les outils de modélisation devraient permettre, par exemple, qu'un modèle ne soit pas conforme à son méta-modèle pendant la phase d'élaboration. Ainsi les aspects cognitifs liés par exemple aux utilisateurs devraient être pris en compte dans les outils de modélisation dès leur création tout en impactant les aspects techniques de maintenance, partage, évolution, etc. Il devrait même être possible de générer le système modélisé ou tout au moins de le simuler lorsque le modèle est incomplet ou non conforme : ceci permettrait au concepteur d'avoir un retour concret de l'état d'avancement de son modèle et lui éviterait un effort important de projection mentale de ce dernier.

Challenge	Aspects sociétaux	Aspects techniques
Représentativité	relations cognitives inter-domaines ; exploitation de représentations graphiques non utilisées en GL	relations entre langages et outillages hétérogènes ; facilité de fusion de modèles
Accessibilité	modèles ouverts ; mécanismes visuels appropriés, interactions mieux alignés avec les gestes de schématisation	outillages ouverts ; modèles comme éléments essentiels du développement logiciel
Flexibilité	environnements facilitant le travail cognitif lors de l'intégration	modèles de référence exposants hypothèses et limites, et pouvant être facilement customisés et intégrés
Evolutivité	rendre compte de l'incomplétude ; simulation de modèles incomplets	outillage agile pour supporter le processus naturel de conception

TABLE 1 – Aspects sociétaux et techniques des points de rupture retenus