

Abstraction et composition en conception de logiciel

Michel Sintzoff

Université catholique de Louvain, Département d'Ingénierie Informatique

1 Utilité de l'abstraction et de la composition

Il ne faut plus démontrer aujourd'hui l'importance de l'abstraction et de la composition dans la construction de systèmes logiciels (souvent appelées « machines » ci-dessous). Rappelons simplement quelques échantillons et variantes de ces concepts : couches, niveaux et affinement, en ce qui concerne l'abstraction, et modules, objets et composants, pour la composition. Ces deux principes frères, avec d'autres comme la redondance et l'économie, facilitent grandement le passage à l'échelle industrielle. Ils ont ainsi constitué des facteurs décisifs dans de remarquables réussites informatiques, entre autres les langages de programmation, les systèmes d'information et de communication, et la conception assistée par ordinateur. La réflexion scientifique sur l'abstraction et la composition a permis la mise au point de notions marquantes telles que l'affinement graduel de machines abstraites, les paradigmes et stratégies de programmation, et les modèles de conception. Ces recherches se poursuivent activement, en particulier sur la composition de modèles de conception.

L'abstraction et la composition sous-tendent également diverses méthodes effectives de validation et de production : mentionnons entre autres l'inspection organisée de logiciels, leur test systématique et simplifié, leur adaptation économique, et l'automatisation partielle de leur conception. Dans ce contexte, une préférence pour des techniques formelles devrait avant tout viser un objectif de raison et de clarté. Comme telle, la formalisation n'est pas une panacée, et même entrave l'abstraction et la composition quand elle impose des démarches pédantes ou fermées. Elle doit être servante et non maîtresse.

Il nous semble de plus en plus que les questions d'abstraction et de composition dépassent de loin celle du style de modélisation. L'approche impérative se justifie très bien à tous les niveaux, pas seulement dans l'implémentation : l'on modélise avec succès des cahiers de charges et des spécifications par des automates munis d'états abstraits. Il vaut mieux considérer le style impératif - ou dynamique, et le style déclaratif - ou statique, comme des aspects complémentaires d'une seule et même réalité : c'est bien ainsi que l'on associe automates et prédicats temporels, ou équations différentielles et fonctions. Le vrai problème consiste plutôt à découvrir les machines pertinentes aux divers niveaux d'abstraction utiles : par exemple, quels automates expriment au mieux les implémentations, les spécifications et les besoins ?

2 Nouvelles applications et problèmes ouverts

Dans la conception de systèmes dits hybrides, l'analyse intégrée de systèmes dynamiques, continus et discrets, se base sur l'abstraction du temps continu - physique ou quantitatif, en un temps discret - symbolique ou qualitatif ; la discrétisation joue ici un rôle d'abord qualitatif et non calculatoire. Ainsi s'explique l'impact récent des automates finis en théorie du contrôle.

Les dérivations formelles, qu'il s'agisse de preuves de théorèmes ou de déductions de logiciel, deviennent de plus en plus utiles à mesure que des méthodes de vérification ou de conception sont mathématisées de plus en plus clairement. L'assistance à la construction de dérivations doit faciliter

leur élaboration par composition et décomposition - ce qui est classique, autant que par abstraction et affinement - ce qui l'est moins. Nous voyons en effet une dérivation abstraite, intellectuellement plaisante, comme une esquisse qu'il s'agit d'affiner progressivement en une dérivation totalement détaillée, probablement inhumaine. L'automatisation d'étapes d'affinement de dérivations s'intègre harmonieusement dans ce cadre.

Dans le cas des systèmes concurrents, les schémas de composition adéquats s'avèrent orthogonaux aux structures syntaxiques, fort centralisées. La difficulté consiste donc à exprimer indépendamment de la syntaxe apparente les structures conceptuelles essentielles, par exemple les ondes de communication ou les instantanés d'états répartis. Un autre défi est l'abstraction de primitives indépendantes qui puissent se composer facilement et qui permettent d'assurer l'efficacité et la robustesse de logiciels formés d'objets répartis et mobiles.

En ingénierie des besoins, l'abstraction et la composition jouent aussi un rôle croissant. Les besoins sont affinés en objectifs, composés entre autres à l'aide de la conjonction logique. Cependant, les fréquents conflits entre objectifs exigent des modes de raisonnement plus souples, tels que le bon sens ou le compromis. De plus, l'expression des besoins caractérisant un logiciel demande une expression conjointe de son environnement. Ceci constitue un problème ardu. Souvent en effet, une spécification de machine s'obtient comme la différence entre les descriptions de l'environnement sans machine et de celui avec machine. Il faudrait dès lors composer la machine par différence entre les compositions des environnements avec et sans machine ; hélas, l'opérateur de différence se marie mal à celui de composition.

L'on a appris à combiner des vues complémentaires dans des cas limités, par exemple celui des automates et de la logique temporelle, mais cela reste toujours une difficulté majeure en général : comment donc composer les facettes architecturales, fonctionnelles et comportementales d'un même système, à un niveau proche de l'application ? Le problème se corse si l'on envisage de nouveaux domaines tels que les multimédia.

3 Conclusion

L'abstraction et la composition constituent bien deux principes importants en conception de logiciel. Leur utilisation effective fut et reste une entreprise de longue haleine. À cet effet, il importe de suivre le mode de travail qui s'est avéré si fructueux jusqu'ici : d'une part, associons étroitement la réflexion scientifique à l'élaboration d'applications caractéristiques ; d'autre part, établissons dès que possible des cadres de référence communs et élégants pour l'utilisation croisée de techniques. Comme toujours, il s'agit de mettre la main à la pâte, de raisonner avant de formaliser, et d'ouvrir les fenêtres et les yeux.