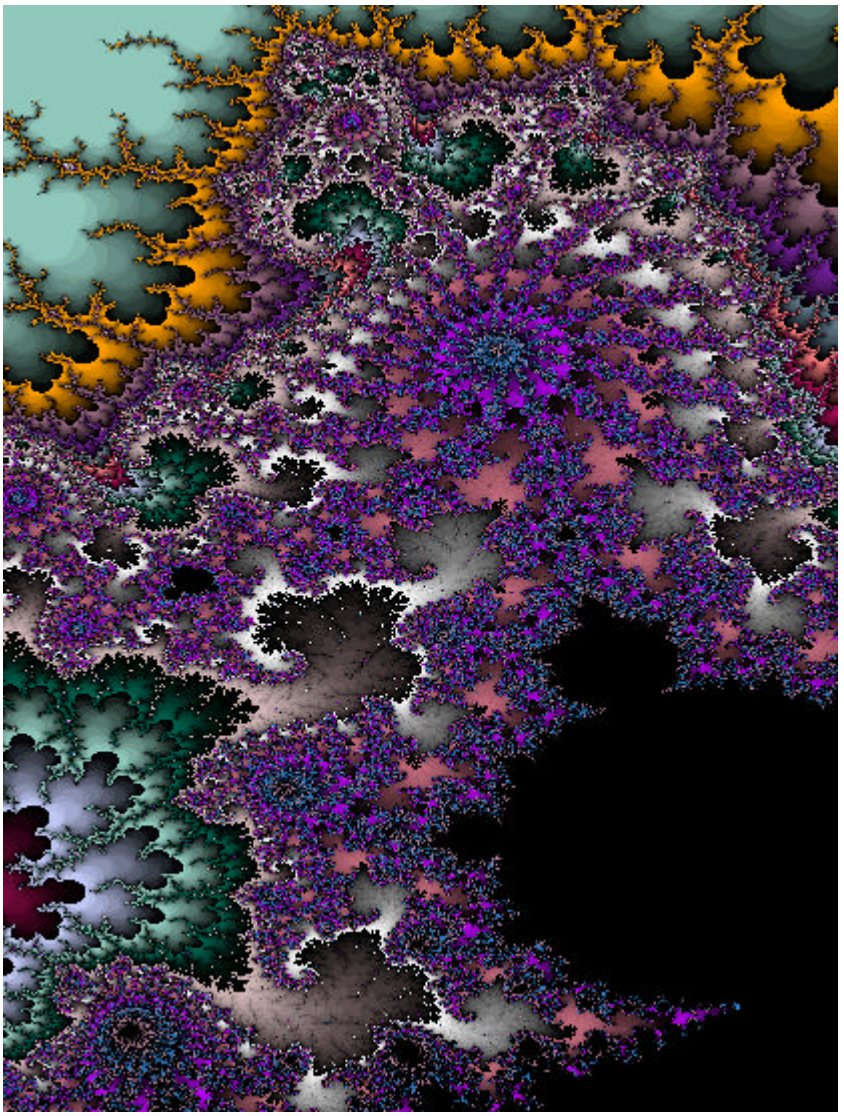
CHAPITRE 18

Exemple d'outil de modélisation



18.1 *A quoi sert l'outil ?*

Un outil de modélisation n'est, dans sa plus simple expression, qu'un éditeur graphique spécialisé. Ainsi, il existe d'excellentes extensions à des éditeurs graphiques populaires (Micrografx Graphics Suite, par exemple) permettant de définir les diverses vues requises par UML. Ces solutions ont l'avantage souvent déterminant d'un prix pratiquement nul, puisque la licence de base est souvent déjà acquise.

Dans le cas d'une école, ces solutions sont le plus souvent amplement suffisantes, dans la mesure où une école n'a pas de contraintes impératives de production, et n'a pas forcément l'obligation de passer du modèle à l'implémentation. Ainsi, pour enseigner UML et la modélisation objet, qui sont des activités sans aucune dépendance d'un quelconque langage de projection, une extension à un programme de dessin vectoriel ou un programme de génération d'organigrammes est-elle amplement suffisante.

Lorsque l'on veut gérer des projets, il peut être utile de disposer d'un outil de génération de code en fonction du diagramme de classes. Certains outils prétendent également générer des scénarios de test en fonction d'un diagramme d'utilisation: en l'état actuel (2000) des choses, il est difficile de considérer ces affirmations autrement que comme une expression d'un marketing douteux. Soulignons que la génération automatique de code représente un confort, non une nécessité.

Plus important est la possibilité pour un outil de modélisation de procéder à une vérification minimale de la syntaxe de modélisation, ce que ne peut faire un éditeur graphique. Ceci permet d'éviter bien des erreurs grossières qui font perdre du temps lors des confrontations entre développeurs.

Une autre possibilité intéressante est le "*reverse engineering*". Les développements plus anciens ne proposent pas forcément un diagramme UML accompagnant la livraison de leur librairie de classe; ceci complique singulièrement la compréhension d'une librairie de classes donnée. Un outil permettant le reverse engineering peut rétablir le diagramme de classes en analysant le code source d'un projet, dans la mesure où celui-ci est disponible, bien entendu!

18.2 *Together*

L'outil de modélisation proposé au cours de ce laboratoire est Together/J, qui utilise Java comme langage de projection. Il est bien entendu que les outils de modélisation sont d'une certaine manière équivalents; certains permettent d'utiliser d'autres langages de projection, supportent éventuellement les diverses vues de UML de manière plus complète; néanmoins, Together représente un outil facile à apprendre, et dont la manipulation correspond bien à la manière de travailler de l'ingénieur, contrairement à d'autres plus coûteux, mais aussi plus théoriques, et mieux adaptés à des environnements purement académiques. L'interactivité entre la production de code et la modélisation est particulièrement bien implémentée par cet outil. Il nécessite environ 128 MB de RAM pour fonctionner correctement sous Windows NT 4; écrit en Java, il peut être utilisé sous UNIX (Solaris, HP-UX), Macintosh (bonne intégration à l'outil de développement intégré CodeWarrior, aussi sous Windows et Linux) ou Linux.

Together permet de générer soit du Java, soit du C++ comme langage de projection. Entièrement écrit en Java, il nécessite une machine virtuelle appropriée; on consultera les recommandations de l'éditeur à l'adresse **http://www.togethersoft.com**. La version de démonstration, appelée "Whiteboard" peut être téléchargée gratuitement, mais ne permet que l'édition de la vue des classes. De plus, il n'est pas possible d'imprimer directement le schéma avec cette version : il faut passer par des impressions d'écran, suivies d'un éditeur graphique permettant de réunir les copies d'écran en une seule image.

Pour les écoles, il est possible de demander une licence site gratuitement, qui donne alors accès à la version "Enterprise", qui est celle que les étudiants utilisent au laboratoire de télécommunications. Pour l'évaluation par une entreprise, il est possible de demander gratuitement une licence limités à une durée de 30 jours.

Relativement au principal outil de modélisation, à savoir Rational Rose, de Rational Software, Together ne permet pas d'utiliser certains langages de projection (Ada, Visual Basic ou COBOL, par exemple) et possède des possibilités un peu plus restreintes. Néanmoins, Together couvre 95% des cas d'utilisation, et 100% des besoins d'une école pour un prix incomparablement plus modique, puisque nul. Noter qu'il existe d'autres outils, dont certains en freeware. Excel Software (qui n'a rien à voir avec Microsoft) fournit également un outil de modélisation intéressant, mais assez coûteux, encore que ridiculement bon marché comparé à Rose.

18.2.1 *Utilisation de Together*

Together est facile à installer, et se présente sous la forme d'une icône sur le bureau. Le chargement est relativement lent, surtout si l'on ne dispose pas de la mémoire suffisante (moins de 128 MB, ou trop d'autres applications actives). Il faut noter que certaines versions de Together ont de sérieux problèmes de gestion de mémoire : on démarre avec 128 MB utilisés, et on se retrouve relativement vite avec 250 MB d'occupation effective, et des messages assez gênants du système d'exploitation qui se retrouve avec une mémoire virtuelle insuffisante. Ceci est en particulier le cas pour les uti-

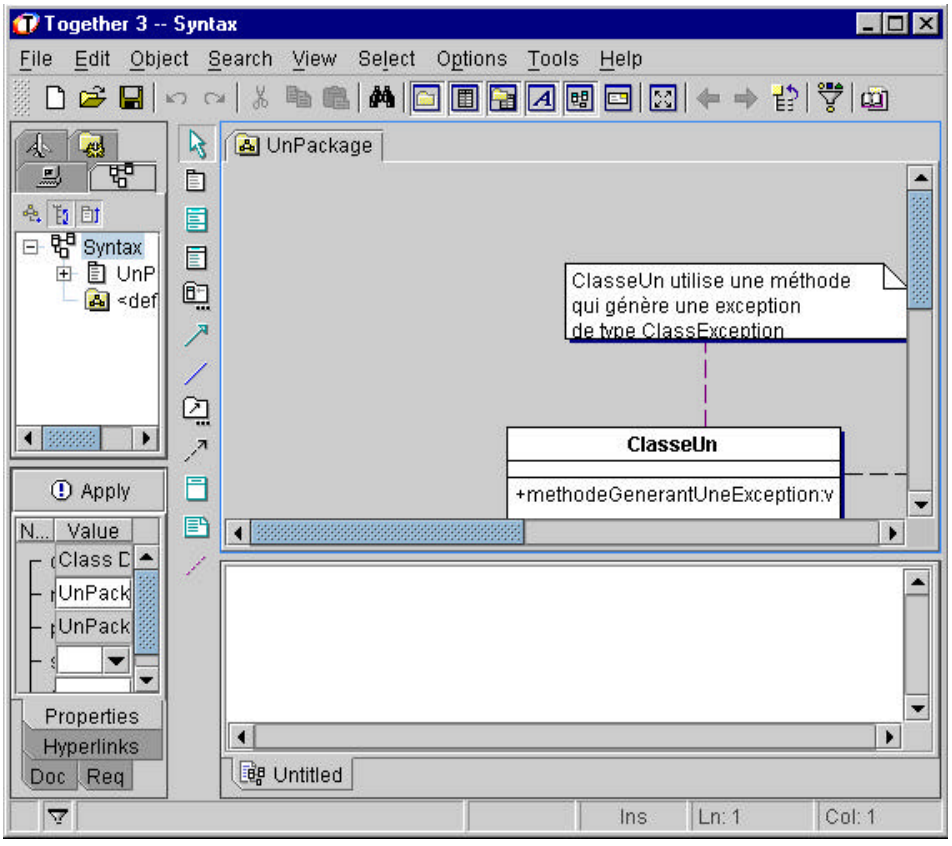
lisateurs de Windows. Il convient de vérifier sur le site de Together quelle est la version de la machine virtuelle la mieux adaptée, et d'installer cette dernière : ceci devrait, sinon résoudre, du moins diminuer l'importance du problème.

Une autre alternative, non exclusive d'ailleurs, est d'augmenter la mémoire virtuelle de Windows NT.

18.2.2 *Interface utilisateur*

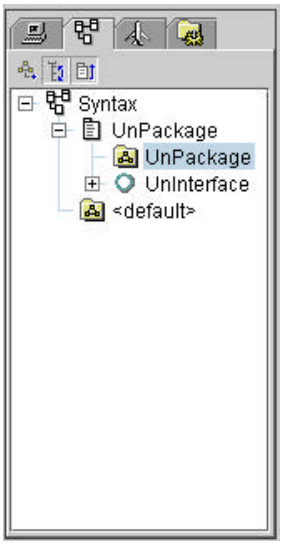
L'utilisation de Java et des classes Swing confère à Together un "look" particulier, inhabituel pour les utilisateurs de Windows, mais sans surprises particulières à l'usage. L'écran peut grossièrement être divisé en quatre zones distinctes, la principale étant la zone d'édition graphique (voir figure18.1, page148)

FIGURE 18.1 Interface utilisateur de Together (Whiteboard Edition)



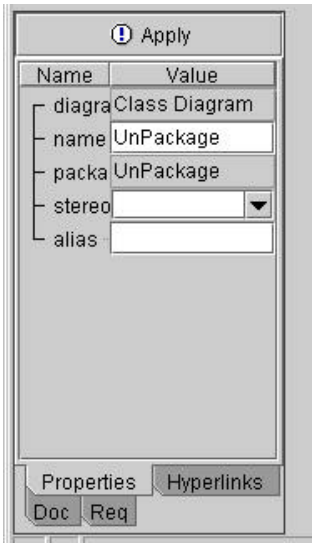
La zone de navigation (figure 18.2, page149) permet, un peu comme un explorateur dans la galaxie Windows ou le Finder dans le domaine Mac, de se déplacer entre les divers projets et paquetages en cours de développement.

FIGURE 18.2 La zone de navigation de Together



La zone d'éditions de propriétés de Together permet, lorsqu'un objet particulier est selectionné dans la zone d'édition graphique, de spécifier de manière simple les propriétés des objets, des membres et des méthodes. C'est dans cette zone que l'on pourra spécifier si un membre donné est public, protégé ou privé. Les diverses dépendances peuvent également être spécifiées dans cette zone de dialogue. Pour l'utilisateur expérimenté, cette zone n'offrira que peu d'utilité : il est souvent plus facile d'utiliser directement la syntaxe UML pour documenter ce genre de propriétés. Néanmoins, pour un utilisateur moins expert, cette zone de dialogue permettra de correctement documenter les propriétés d'un membre ou d'une méthode.

FIGURE 18.3 La zone d'édition de propriétés de Together



18.2.3 *Edition graphique du modèle*

Le modèle peut être défini sous forme graphique, en sélectionnant un outil dans la barre d'outils, et en cliquant à l'endroit approprié dans la fenêtre graphique. La zone d'édition graphique prend une signification différente selon la vue que l'on désire considérer.

FIGURE 18.4 La zone d'édition graphique de Together

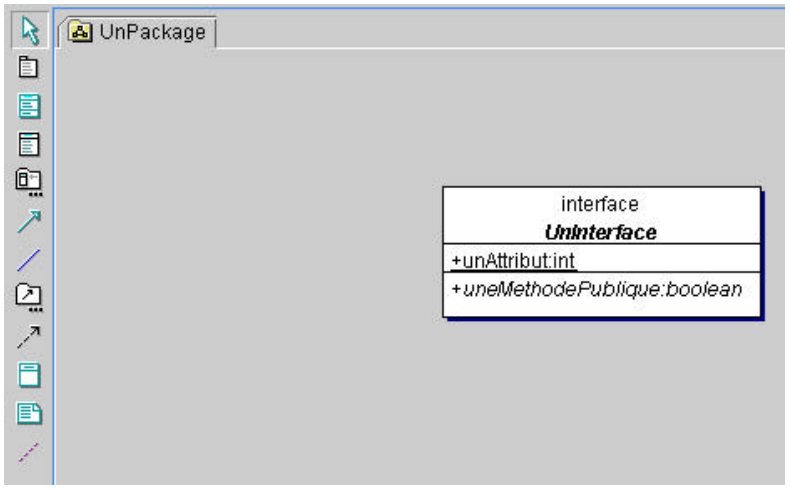
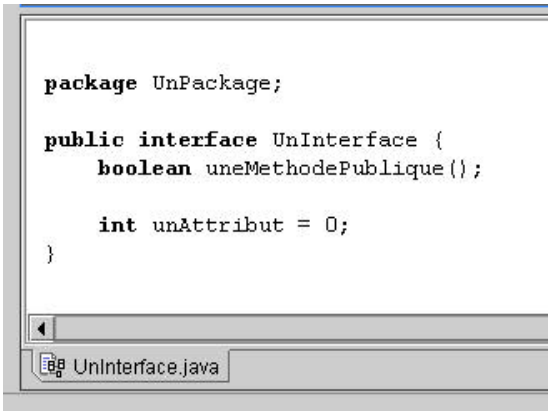


FIGURE 18.5 La zone d'édition de code de Together



Dans la zone d'édition de code, présente uniquement dans le mode "Diagramme de classes", Together génère automatiquement le code correspondant au modèle, ou vous permet d'éditer du code, et d'adapter automatiquement le modèle aux modifications apportées.

18.2.4 *Autres facilités fournies par un outil de modélisation*

Un tel outil est exceptionnellement puissant pour la génération de documentation, avec des formats extrèmement variés. On consultera le mode d'emploi en format PDF de l'outil pour des précisions.

Together permet une intégration aisée de patrons (patterns). Il est ainsi possible, très simplement, de définir une classe comme étant l'implémentation d'un pattern particulier dans le diagramme de classes, par exemple. Les Enterpirse Java Beans sont partie constituante de la bibliothèque de Together.