INTRODUCTION

Contexte de recherche

Faire face à l'évolution des systèmes logiciels demeure actuellement un problème majeur du génie logiciel. En effet, dès que le logiciel est livré, il débute une longue phase de maintenance qui est censée faire face aux changements des besoins des utilisateurs, améliorer les performances, l'adapter aux changements de la plateforme matérielle ou simplement corriger les erreurs non détectées durant la phase de test.

Faire évoluer un logiciel est souvent problématique pour diverses raisons. Au-delà du fait qu'il faut disposer d'une documentation adéquate du logiciel pour le comprendre, il faut aussi opérer une phase de reverse engineering qui vise à retrouver des décisions de conception non apparentes dans la documentation ou le code du logiciel. Sans une bonne compréhension, les changements opérés peuvent engendrer d'autres changements envahissants, altérer l'architecture du logiciel (on parle alors de dérive architecturale) ou ne plus garantir diverses propriétés importantes. Ce qui a pour conséquence d'altérer la qualité du logiciel et conduire à son retrait prématuré.

Bien que l'évolution constitue la phase la plus couteuse du cycle de vie du logiciel, elle est restée longuement menée par des approches ad hoc où règnent l'improvisation et le pragmatisme. L'approche des systèmes ontogénétiques aborde l'évolution d'une manière qui s'inspire de l'ontogénèse des systèmes biologiques. Son point de départ consiste à concevoir dès le début un système logiciel pour évoluer. Autrement dit, le logiciel n'est plus une infrastructure instantanée relative à une réalité donnée, à un moment donné, mais une infrastructure qui évolue continuellement et de manière autonome et cela dès sa création initiale.

Cette vision radicale de l'évolution sépare les changements possibles en deux groupes : les changements anticipés et les changements non anticipés.

Les changements anticipés sont les changements qu'on peut prévoir alors que le logiciel est en cours de développement. D'habitude, de tels changements sont pris en compte lors du développement et incorporés dans le code du logiciel moyennant une chaîne d'instructions alternatives éventuellement imbriquées (if … then…else… if … then…else…). Ce qui altère inévitablement les performances du logiciel.

L'approche ontogénétique reporte la prise en compte des changements anticipés jusqu'à ce qu'ils deviennent nécessaires (i.e. certaines conditions se réalisent), suite à cela le code est altérer dynamiquement pour incorporer ces changements. Cette approche a l'avantage de maintenir le code performant, simple et reflétant les besoins réels actuels. Concrètement le code est débarrassé des structures alternatives au profit d'une évolution dynamique des comportements des objets en fonctions des besoins réels actuels. Bien entendu cela suppose que le code du logiciel ait une structuration qui permet d'incorporer les changements de manière adéquate et que la plateforme d'accueil soit dotée de composants qui peuvent altérer le code du logiciel dynamiquement en fonction de la description du changement anticipé.

Les changements non anticipés sont des changements qui apparaissent une fois le logiciel livré. L'approche ontogénétique traite les deux types de changement de la même façon au niveau opérationnel. Cependant au niveau abstrait, le traitement diffère. La démarche utilisée doit tenir compte d'un existant qu'il faut comprendre, dont-il faut respecter certaines propriétés, etc.

La prise en compte de l'ontogenèse constitue un nouveau défi et une vision radicale de l'évolution qui a un effet aussi bien sur notre façon de percevoir les systèmes logiciels que notre manière de les concevoir. Nous nous intéressons dans cette thèse à la proposition d'une démarche pour le développement des systèmes ontogénétiques. Etant donné que la création d'une nouvelle méthode dédiée à ces systèmes est une tâche fastidieuse et demande des efforts qui outrepassent le contexte académique dans lequel nous travaillons, nous avons opté pour l'extension d'une méthode existante. Afin de faire face à la complexité croissante du développement logiciel, des méthodes de gestion des projets informatiques ont été introduites. Celles dites traditionnelles sont axées-plan et généralement lourdes, ces méthodes ont perdu de plus en plus de partisans en faveur des nouvelles méthodes dites « agiles ».

Nous avons étudié plusieurs méthodes agiles afin de choisir la plus apte à une extension supportant les systèmes ontogénétiques et nous avons opté pour l'extension du Rational Unified Process. Le RUP est l'une des méthodes agiles basées sur des pratiques du génie logiciel avérées et sur le formalisme UML (Unified Modeling Language), qui présente un méta-modèle d'architecture extensible et personnalisable [Kru 03].

Notre travail dans cette thèse, concerne l'étude des opportunités offertes pour l'ontogénèse par le processus RUP et le formalisme UML en termes d'éléments de modélisation ou artéfacts qui sont créés et maintenus dans la démarche et font la base du développement de systèmes logiciels. L'approche que nous proposons, pour la prise en compte de l'ontogenèse, étend principalement le processus d'ingénierie des besoins de changements/évolutions anticipés, et propose également une discipline pour la prise en charge des évolutions non anticipées et la mise à jour dynamique d'un système.

Problématique

L'approche ontogénétique affecte toutes les phases du cycle de vie d'un logiciel. Le support d'une telle approche demeure un défi considérable. L'approche nommée Mage donne une implémentation possible de l'approche ontogénétique [Mes 06]. Cependant, comme souligné dans [Khe 09a], le défi de l'approche se situe plutôt au niveau de la démarche de développement. En effet, l'approche soulève de nombreuses question : Comment peut-on déduire les changements anticipés ou non anticipés ? Comment analyse-t-on l'état actuel d'un système et après plusieurs changements ? Comment garantir formellement que l'introduction d'un changement préserve les propriétés invariantes d'un système ? Comment préserver l'architecture d'un système ? Comment garantir le maintien des qualités de services ? Existent-ils des méthodes de développement qui pourront résoudre ou répondre à ces questions ?

Malgré l'importance de l'approche ontogénétique, il n'existe aucune méthodologie de développement pouvant la supporter.

En effet, tout changement des besoins est géré avant la livraison du produit final. Appréhender préalablement le changement, c'est-à-dire permettre de concevoir le système pour changer, est une vision non prévue dans toutes les méthodes de développements de systèmes logiciels, qu'elles soient traditionnelles ou agiles. En particulier, les approches agiles tiennent compte du changement durant le développement mais elles ne sont pas armées pour créer de systèmes ontogénétiques. En général, dans les méthodes existantes, le processus de changement est assuré par l'intervention de l'être humain sur un système logiciel pour le maintenir conforme au domaine qu'il représente et répondant aux besoins évolutifs des utilisateurs. En conséquence, la problématique, dans ce contexte, demeure la recherche d'un cadre méthodologique pour supporter les systèmes ontogénétiques.

Motivations

Le nombre d'opérateurs impliqués dans l'utilisation des logiciels, les enjeux qui en découlent ainsi que les ressources énormes engagées dans la production et la maintenance de logiciels sont des facteurs qui justifient et motivent les travaux de recherche sur l'évolution, les problèmes qu'elle cause et les solutions possibles.

L'approche ontogénétique est un paradigme, qui offre une nouvelle vision de l'évolution. Les systèmes logiciels ontogénétiques présentent la caractéristique d'évoluer dynamiquement et de manière autonome pour répondre aux besoins des utilisateurs et leurs changements qu'ils soient anticipés ou non. L'évolution de tels systèmes est perçue comme un processus qui les façonne dès leur création. Pour tirer profit de tels systèmes, il

est nécessaire d'investir dans une démarche de développement appropriées qui redonnent à l'évolution le statut de concept central.

Objectifs

Le travail de cette thèse se situe au carrefour de plusieurs domaines : l'ingénierie des méthodes de développement de systèmes logiciels, ingénierie des besoins, et la modélisation de l'évolution des besoins. Nous nous sommes fixés comme objectifs l'extension du processus RUP pour supporter les systèmes ontogénétiques.

Contenu du mémoire

Cette thèse se compose de cinq chapitres, de cette introduction et de la conclusion générale. Le partage que nous avons adopté reflète, à la fois, notre démarche de travail durant cette thèse et un ordre intelligible.

Chapitre 1. Ce chapitre traite de l'ontogénèse comme une nouvelle vision de l'évolution des systèmes logiciels. On y présente aussi l'approche ontogénétique Mage.

Chapitre 2. Dans ce chapitre nous discutons l'existant concernant les processus de développement de systèmes logiciel. Nous y décrivons particulièrement le processus Rational Unified Processus que nous avons choisi d'étendre.

Chapitre 3. Dans ce chapitre nous présentons une étape clé dans le cycle de vie des systèmes logiciels et ontogénétiques, il s'agit de l'ingénierie des besoins. Nous donnons des concepts et approches existants dans la littérature et nous terminons par les paradigmes et approches utilisés dans le processus d'ingénierie proposé dans notre démarche.

Chapitre 4. Décrit l'approche proposée pour le support du développement des systèmes ontogénétiques.

Chapitre 5. Présente des exemples d'application pour montrer le scénario de développement dans le cadre de la démarche proposée et donne un aperçu sur l'aspect implémentation.

CHAPITRE 1
LES SYSTEMES ONTOGENETIQUES ET MAGE

1.1. Introduction

En analysant les organismes vivants, on peut constater deux types d'évolutions. Une évolution qui façonne l'organisme de sa naissance à sa mort et une évolution qui tend à différencier les descendants d'un organisme par rapport à ce dernier. Les systèmes biologiques renferment des mécanismes et des processus longuement éprouvés qui leur ont permis de survivre face à un environnement changeant et contraignant. Ce fait a conduit l'être humain à les considérer comme source d'inspiration dans la conception des systèmes artificiels. Ce chapitre s'intéresse aux processus naturels dédiés aux changements et à l'évolution, et présente l'approche Mage qui modélise l'ontogenèse d'un système logiciel sous la forme d'un génome embarqué, dont le rôle consiste à façonner continuellement le système en fonction des changements anticipés et non anticipés.

1.2 Facettes de l'évolution

Il est actuellement communément admis que les organismes biologiques sont façonnés par trois processus ou trois niveaux d'organisation : ontogenèse, épigenèse et phylogenèse. Ces trois processus, connu aussi sous l'appellation du modèle POEtic (appellation issue de Phylogenesis, Ontogenesis et Epigenesis) ont été identifiés dans le cadre du projet " Reconfigurable POEtic Tissue", lequel projet fut, conduit sous l'égide du programme européen des technologies de la société d'information (European Program of Information Society Technologies) [Mes 06]. Nous présentons dans ce qui suit les trois processus puis nous discutons des métaphores qui en sont issues.

1.2.1 Ontogenèse

Les organismes vivants ne naissent pas complètement formés tels que nous les voyons. L'organisme commence sa vie en tant que cellule unique, dotée d'un programme de développement codé dans son génome. Ce dernier est continuellement exécuté par la

cellule, ce qui conduit à sa division répétée engendrant une multitude de cellules identiques ayant le même programme [Mes 06]. Suite à cela, une forme de communication s'opère entre les cellules permettant à chacune d'exécuter la partie du programme du génome qui correspond à sa position dans l'ensemble. Cette différenciation cellulaire conduit, en fin de compte, à la formation d'organes et donne à l'individu la morphologie et le comportement spécifiques à son espèce. Le processus de développement qui façonne un organisme tout au long de sa vie est appelé ***ontogenèse***. L'ontogenèse est un processus déterministe dont l'exécution est influencée par l'environnement où elle se déroule.

1.2.2 Phylogenèse

Dans une espèce donnée, la reproduction consiste à transmettre le génome d'un ou deux ascendants vers le descendant. Le génome de la première cellule du descendant est obtenu de celui de ces ascendants par mutation et entrecroisement. Etant différent de celui de ces ascendants, le génome contrôle l'ontogenèse et produit un organisme différent des ascendants. Grâce au changement du génome, le descendant acquière de nouvelles propriétés dont dépend sa survie. La mutation et l'entrecroisement produisent progressivement un changement et une évolution de l'espèce d'une génération à l'autre. Cette évolution est appelée *phylogenèse*. La phylogenèse est un processus non déterministe qui n'a aucun effet sur l'organisme lui-même mais en a sur l'espèce. La phylogenèse introduit la diversité dans les organismes vivants et elle est importante pour leur survie, leur adaptation continue à l'environnement ainsi que l'apparition de nouvelles espèces. Le processus phylogénétique est basé sur la sélection naturelle qui permet la survie des individus adaptés à leur environnement. Par conséquent, à travers la phylogenèse, l'environnement peut avoir un impact majeur sur l'évolution des espèces [Mes 06].

1.2.3 Epigenèse

Vu que le génome est limité dans la quantité d'informations qu'il peut stocker, et vu que l'altération du génome par l'environnement à travers l'ontogenèse et la phylogenèse est lente et limitée, les organismes complexes sont façonnés par un troisième processus appelé *épigenèse*. Ce dernier utilise des structures spécifiques (dans l'individu) pour stocker et gérer un nombre important d'interactions avec l'environnement. Le processus épigénétique est supporté par les trois systèmes : le *système nerveux*, le *système endocrinien* et le *système immunitaire*. Les structures utilisées par ces systèmes sont facilement altérables et permettent aux organismes vivants complexes d'apprendre et d'effectuer un traitement symbolique de l'information [Mes 06].

1.2.4 Métaphores Biologiques et processus d'Evolution

On entend par métaphore biologique une analogie qu'on cherche à déterminer entre le monde biologique et le monde artificiel, de façon à pouvoir proposer des approches qui imitent certains aspects du premier, tout en en ignorant d'autres. Fondamentalement, les métaphores ne cherchent pas à reproduire ce qui est biologique, mais plutôt à l'interpréter en fonction de ce qu'il est possible et raisonnable de réaliser. De ce fait, on peut déduire que les métaphores biologiques sont évolutives et dépendent de notre compréhension de la réalité et notre aptitude à en extraire des éléments pratiques et bénéfiques.

Les tendances actuelles des systèmes bio-inspirés proposent de nouvelles métaphores biologiques et leur combinaison (i.e. hybridation) pour construire des systèmes exhibant les propriétés désirables comme les comportements émergeants, l'adaptabilité à l'environnement, la cicatrisation, etc. Nous décrivons ci-après ces processus et les métaphores qui en ont été déduites.

Ontogenèse et métaphores. Le processus de développement qui façonne un organisme tout au long de sa vie est appelé ontogenèse. L'ontogenèse est un processus déterministe dont l'exécution est influence par l'environnement où elle se déroule. Concernant l'ontogenèse, la première tentative d'inspiration fut celle de Von Neumann avec sa machine qui s'auto duplique (self replicating machine). Cette dernière est un automate capable d'effectuer un calcul universel (i.e. équivalent à la machine de Turing) et une construction universelle. Les métaphores courantes tentent de mimer d'autres mécanismes de l'ontogenèse comme la division ou la différentiation cellulaire [Mes 06].

Phylogenèse et métaphores. La métaphore phylogénétique courante est celle des algorithmes génétiques et leurs diverses variantes. D'une manière générale, les algorithmes génétiques imitent la phylogenèse en résolvant des problèmes d'optimisation. Les solutions candidates d'un problème donné sont considérées comme des individus et codées sous forme d'un génome ayant la forme d'une chaîne de symboles abstraits. Les individus dans l'espace des solutions candidates sont ensuite évolués par des opérations de mutation et d'entrecroisement puis sélectionnés d'une génération à l'autre moyennant une fonction d'adaptation (i.e. un critère de qualité donné). Cette procédure itérative continue jusqu'à ce qu'un nombre d'itération soit achevé ou qu'aucune amélioration sensible n'est enregistrée. Malgré la simplicité et la nature aveugle de l'approche, elle a été utilisée avec succès à travers un large éventail d'applications [Mes 06].

Epigenèse et métaphores. Des métaphores de l'épigenèse (système nerveux, système immunitaire et système endocrinien) sont actuellement utilisées. Le système nerveux fût le premier à être étudié et à recevoir un maximum d'attention, ce qui a donné naissance au domaine des réseaux de neurones artificiels. La métaphore des réseaux de neurones

artificiels est une tentative à mimer les caractéristiques fondamentales des neurones biologiques. Les réseaux de neurones artificiels peuvent être vus comme des graphes dirigés avec des connections pondérées entre les neurones (i.e. les synapses). La possibilité d'apprendre par des processus d'apprentissage dans le contexte des réseaux de neurones artificiels est vue comme un problème d'ajustement de l'architecture du réseau et des poids des connexions, de telle sorte à ce que le réseau puisse effectuer une tâche spécifiques de façon performante. Les réseaux de neurones artificiels sont convenables pour les problèmes où la compréhension est limitée ou incomplète et où existe une abondance de données. C'est le cas typique du problème de reconnaissance de formes. Le succès des réseaux de neurones artificiels est comparable à celui des algorithmes génétiques ; toutefois leur domaine d'application est relativement limité [Mes 06].

En appliquant les processus précédents, l'aspect structurel de l'organisme peut être vu comme composé d'un génome et d'un phénotype. Ce dernier comprend les propriétés dérivées du génome en utilisant n'importe lequel des processus Poetic. Le phénotype a donc une partie innée dérivée par l'ontogenèse et une partie acquise dérivée par épigenèse. Les deux étant façonnés à long terme par la phylogenèse.

Actuellement, le processus de changement est vu principalement comme l'intervention de l'être humain sur un système pour le maintenir conforme au domaine qu'il représente et répondant aux besoins évolutifs des utilisateurs. En effet, dans l'activité de maintenance, on considère un modèle comme une entité passive ou factuelle. Lorsqu'on modélise le processus ontogénétique, on appréhende préalablement le changement : on conçoit le système pour changer. L'ontogenèse est vue comme une nouvelle dimension des systèmes logiciels qui est orthogonale à leurs dimensions structurelle et comportementale. En tant que nouvelle dimension, l'ontogenèse présente la caractéristique d'englober tout type de changement qu'un système subit, pendant son développement et sa maintenance.

Le changement de façon autonome est une caractéristique souhaitable dans les systèmes adaptatifs qui doivent s'accommoder d'environnements d'exécution différents. L'autonomie implique la mise à jour dynamique d'un système. C'est une caractéristique souhaitable pour les systèmes embarqués et nécessaire dans les systèmes critiques qui ne peuvent tolérer un arrêt d'exécution. On cite : les systèmes de contrôle du trafic aérien et ferroviaire, la gestion des commutations téléphoniques, la gestion des transactions financières et la gestion des centrales téléphoniques, la gestion des transactions financières et la gestion des centrales électriques [Mes 06].

1.3 L'ontogénèse biologique

1.3.1 Définitions

Un être vivant est un organisme qui présente une forme bien définie, qui est animé, qui respire, s'alimente, se développe et se reproduit. Nous entendons par système biologique, un organisme isolé aussi bien qu'un ensemble composé de plusieurs organismes vivants, semblables ou différents. L'étude des systèmes biologiques est une tâche difficile, car un être vivant pris isolement, présente déjà une grande complexité. En effet, dans sa forme la plus élaborée, l'organisme vivant est composé d'organes liés et ordonnés de façon à assurer sa survie et sa reproduction. Plusieurs être vivants peuvent s'assembler dans un système où règnent un ordre et des interactions complexes. Dans la nature, tout système est sujet à l'évolution.

Il existe plusieurs définitions de l'évolution biologique. Bien qu'elle désigne un nombre important de concepts, les définitions existantes vont dans le même sens : c'est-à-dire le changement à long terme des espèces.

Dans l'encyclopédie Universalis, l'évolution est définie par [Uni 14a] :

Le processus par lequel, au cours des âges, se succèdent et s'engendrent, tout en variant, les espèces végétales et animales.

Dans [Rid 96], on trouve la définition suivante :

Evolution signifie changement dans la structure et le comportement des organismes, au fil des générations. Les aspects divers des organismes actuels, à tous les niveaux depuis la séquence de leur ADN jusqu'à leurs structures macroscopiques ou jusqu'à leur comportements sociaux, sont issus de la modification de ceux de leur ancêtres.

1.3.2 Développement des organismes

Dès sa naissance, et même avant, un organisme se caractérise par un ensemble de propriétés que l'être humain peut percevoir par ses organes sensoriels ou par ses capacités d'analyse et de déduction. Cet ensemble de propriétés, appelé ***phénotype***, renferme aussi bien des propriétés physiques que comportementales. Par exemple, une plante peut être caractérisée par sa taille, sa morphologie, sa saison de floraison, son mécanisme de reproduction. De même, un animal peut inclure dans son phénotype des propriétés telles que la couleur, le poids, la possession des ailes, la stratégie de défense contre les prédateurs, etc.

Durant sa vie, un organisme change continuellement son phénotype et passe d'une phase à une autre. Le développement désigne ce processus de changement continuel et couvre deux fonctions consistant à engendrer la diversité et à assurer la continuité de la vie d'une génération à la suivante.

1.4 Vers des Systèmes Ontogénétiques

Plusieurs applications importantes doivent s'exécuter continuellement et sans interruption. C'est le cas des systèmes critiques, tels que la commutation téléphonique, les transactions financières, la réservation des places d'avion et le contrôle du trafic aérien ou ferroviaire. Un autre facteur vient augmenter le rang des applications critiques, il s'agit du développement rapide de l'utilisation du Web et d'Internet pour différents services. Ainsi, pour diverses applications, telles que le commerce électronique, l'arrêt total est préjudiciable : perte de certains états accumulés (états non persistants) et annulation des traitements en cours, pouvant engendrer des incohérences mal perçus par les utilisateurs de ces applications. Les systèmes mobiles représentent une catégorie de systèmes qui peuvent tirer un avantage certain de la mise à jour dynamique car elle leur permet une adaptation rapide aux changements fréquents de leur contexte [Mes 06].

Par ailleurs, il y a nécessité de changer ces applications pour permettre l'élimination de certains bugs, l'ajout/amélioration des services qu'elles offrent, le changement de plateformes, etc. De ce fait, une mise à jour dynamique est nécessaire. Par dynamique, on sous entend son déroulement pendant l'exécution de l'application. La mise à jour dynamique est un besoin ancien, cependant, les réponses qui ont été apportées jusqu'a ce jour sont, dans leur grande majorité, spécifiques aux applications [Hic 01].

D'une manière générale, la mise à jour dynamique se définit par la mise à jour d'un système logiciel pendant son exécution pour modifier son code ou ses données [Bie 03].

Bien que les systèmes conçus par l'homme ressemblent de plus en plus aux systèmes biologiques, la ressemblance n'est qu'apparente. Sans entrer dans un long débat sur les caractéristiques de la vie et du vivant, on peut facilement réunir sous l'intitulé « inaptitude à l'évolution » les caractéristiques de nos systèmes artificiels.

En effet les organismes biologiques sont autonomes, adaptatifs vis-à-vis de leur environnement, se reproduisent, se développent et guérissent suite aux maladies, se régénèrent et se cicatrisent suite aux blessures, résistent aux attaques et développent des mécanismes de défense, etc.

A l'opposé, nos systèmes artificiels logiciels ou autres, ne présentent que superficiellement quelques unes de ces propriétés. Nos systèmes les plus complexes peuvent s'arrêter de façon nette si le bit du code d'une instruction est altéré. Nos circuits électroniques les plus sophistiqués peuvent faillir à leurs fonctions suite à une augmentation ou diminution de température. Les systèmes hautement critiques peuvent subir des attaques virales sans qu'il y apparaisse une réaction convenable qui préserve leurs fonctions essentielles, etc.

Un système logiciel ontogénétique est un système doté d'un mécanisme qui fait évoluer ses propriétés statiques et comportementales de manière autonome, dynamique et continue en réponse à des stimuli internes et cela conformément à une description de l'évolution préalablement établie et embarquée dans le système lui-même.

Cette définition entraine des caractéristiques qui différencient un système classique d'un système ontogénétique. Nous discutons ces points dans ce qui suit :

- **Evolution des propriétés statiques et comportementales**

Modéliser un système revient à modéliser sa structure statique et son comportement. Dans les approches classiques les comportements font évoluer le contenu des variables qui représentent l'aspect statique du système mais ne font pas évoluer le type de ces variables ou leur nombre. De même les comportements ne sont pas sujets au changement. Dans les systèmes ontogénétiques, il est question de faire évoluer la structure et le comportement d'un système à l'image de ce que ferait un agent de maintenance.

- **De manière autonome**

Le système évolue de manière autonome sans intervention externe. Ceci revient à redonner au système une place plus active en lui affectant une charge plus importante dans les tâches qui le font évoluer.

- **De manière dynamique**

Le système évolue tout en restant opérationnel.

- **De manière continue**

L'évolution a lieu à tout moment. C'est un processus continu qui façon le système depuis sa création.

- **En réponse à des stimuli internes**

Faire évoluer un système vient suite à un besoin quelconque qui se manifeste par des stimuli qui émanent du système lui-même.

- **Description de l'évolution préalablement établie**

Toute opération d'évolution est décrite préalablement sous forme d'un patch puis injectée dans le mécanisme qui fait évoluer le système.

- **Embarquée au sein du système lui-même**

La description de l'évolution est stockée sous une forme ou une autre dans le système.

Ces caractéristiques sous entendent que :

Un système ontogénétique se compose de deux parties, une partie opérationnelle qui exécute des fonctionnalités et une partie qui prend en charge l'évolution continue de la première.

L'évolution prend un statut central dans le sens où le système est conçu pour changer.

Le système est doté de moyens qui lui permettent d'appréhender les stimuli de l'évolution.

Le processus de l'évolution est modélisé d'une façon ou d'une autre moyennant des concepts spécifiques.

Toutes les formes d'évolution sont traitées de la même façon. Ce point est important car les systèmes ontogénétiques présentent l'avantage de considérer les deux formes de changements (anticipés ou non) de manière identique.

1.5 Le Modèle Ontogénétique Mage

1.5.1 Principes

Mage est une approche basée sur quatre principes :

- Tout changement qu'un système subit fait partie de son processus ontogénétique.
- L'ontogenèse est un processus continu. Les changements sont déclenchés dynamiquement par des événements internes et externes.
- L'ontogenèse est une dimension embarquée. Le système se développe de façon autonome.
- Les changements anticipés sont codés dans le génome et s'exécutent une fois les conditions de déclenchement atteintes. Les changements non anticipés sont codés puis injectés dans génome selon le besoin.

1.5.2 Motivations

L'approche Mage est motivée par un ensemble de propriétés dont on souhaite doter les systèmes logiciels. Nous les décrivons ci-après.

1.5.2.1 Modélisation du processus ontogénétique

La modélisation du processus ontogénétique implique deux aspects : l'intégration du processus dans le modèle et sa modélisation comme étant un processus continu.

Actuellement, le processus de changement est vu principalement comme l'intervention de l'être humain sur un système pour le maintenir conforme au domaine qu'il représente et répondant aux besoins évolutifs des utilisateurs. En effet, dans l'activité de maintenance, on considère un modèle comme une entité passive ou factuelle. Lorsqu'on modélise le processus ontogénétique, on appréhende préalablement le changement : on conçoit le système pour changer. Par modélisation, nous entendons : investir le futur d'un système pour déterminer ce qu'il deviendra, quand et sous quelles conditions il doit changer, puis ajouter un mécanisme au modèle du système, de telle sorte qu'il devienne capable de changer de façon autonome.

On peut considérer que les modèles orientés objet actuels ne sont pas factuels, car ils englobent le comportement des objets. Ceci est particulièrement vrai lorsqu'on les compare avec les modèles relationnel ou sémantique. Cependant, un objet qui possède un certain comportement, ou une certaine structure, correspond à un fait qui est susceptible de changer dans le temps. La modélisation de ce changement échappe aux approches actuelles : c'est la dimension manquante. Cette dimension est plus complexe que celles structurelle ou comportementale. De plus, elle leur est orthogonale et englobante.

Avec l'évolution des paradigmes de modélisation, nous sommes passés de la modélisation des structures statiques à la modélisation des comportements par l'intégration de ces derniers dans le modèle. En faisant cela, l'approche de modélisation est passée d'une vue statique à une vue dynamique du monde réel. Dans ce travail, nous suggérons d'aller plus loin dans cette direction en intégrant le processus de changement dans le modèle lui-même. Notons qu'aboutir à un tel modèle d'une réalité est une tâche difficile, qui demande une investigation plus profonde que celle qui consiste à extraire le comportement des objets. En effet, l'expérience avec le modèle orienté objet nous a montré que lorsqu'on passe d'une vue statique à une vue dynamique d'un système complexe, l'effort de modélisation devient considérable [Rum 96]. Cependant, l'intégration de l'aspect dynamique a permis d'obtenir des modèles plus proches de la réalité et cette proximité élimine de nombreux problèmes de maintenance. Nous prévoyons des retombées positives par le passage vers les modèles ontogénétiques.

1.5.2.2 Réduction des interactions avec le monde réel

Puisqu'elles impliquent souvent l'être humain, elles sont génératrices d'erreurs et, de ce fait, leur réduction est souhaitable. On distingue deux types d'interactions. Premièrement, les interactions qui visent à changer le modèle, elles sont réduites dans Mage, vu que les

changements sont anticipés et codés dans le génome : le système évolue de façon autonome. Les interactions correspondant aux changements non anticipés demeurent et nécessitent un changement du génome.

Deuxièmement, certaines interactions proviennent de l'utilisation des fonctionnalités du modèle : entrées, sorties et perception de stimuli externes. Même si ces interactions dépendent de la conception, elles peuvent être réduites si le modèle est toujours proche de la réalité. Par exemple, si un attribut n'est plus nécessaire, sa suppression évite de continuer à lire ou écrire sa valeur.

1.5.2.3 Réduction de l'effort de maintenance

Bien qu'un système ontogénétique nécessite moins d'interactions externes pour changer, ce n'est pas une raison directe de réduction de l'effort de maintenance. En effet, il faut tenir compte du fait, qu'au départ, on consent un effort important pour modéliser le processus de changement. La réduction vient plutôt du fait que l'anticipation des changements futurs permet d'éliminer certains changements inappropriés et les retours arrière correspondants. Par exemple, si une analyse conduit à la conclusion que, dans une certaine phase de la vie d'un objet, un type d'attribut doit être réel, on peut, éventuellement, éviter des changements intermédiaires vers les types inclus dans le type des réels.

1.5.2.4 Amélioration des performances

Elle découle des possibilités de spécialisation de programmes et d'adaptation dynamique [Kis 03, Shu 03]. Pour appréhender cela, il faut comparer un système Mage avec un *`système stable`*. Par stable, nous entendons un système qui s'accommode des changements anticipés sans maintenance et qui peut tourner sur plusieurs plateformes. Par exemple, on peut imaginer une classe qui regroupe toutes les propriétés et prévoir des instructions conditionnelles qui évitent une utilisation incorrecte (par exemple la chenille ne peut voler, telle plateforme n'a pas de coprocesseur dédié au traitement en virgule flottante, etc.). Tenir compte de toutes les situations d'incompatibilité rend le système stable non efficace, complexe et enchevêtré. Il est préconisé, dans Mage, la mise à jour dynamique d'un système et l'utilisation d'un mécanisme d'exception général (inspiré de celui de Java), pour tenir compte de l'existence d'objets à différentes phases de leur vie. De même, il est possible d'adapter dynamiquement un modèle Mage à sa plateforme d'exécution.

1.5.2.5 Implémentation dynamique de la variabilité

L'approche Mage considère que la variabilité ne se termine pas une fois le logiciel livré car, suite à une certaine période d'utilisation, d'autres éléments de différence peuvent apparaître. Mage préconise que la variabilité est en partie postérieure à la livraison car ce qu'impose l'environnement sur une famille de produits ne peut être totalement anticipé [Cop 98, Gur 01]. Par exemple : Quel type de périphériques le système doit-il gérer ? Quel type de fonctionnalités doit être en arrière plan pour telle ou telle catégorie de clients ? etc., sont des questions pour lesquelles la réponse peut ne pas exister avant la livraison du produit. Une variabilité prise en charge par le système lui-même, une fois livré, peut tenir compte d'un nombre important de détails et simplifie la tâche des développeurs.

1.5.2.6 Création d'une représentation convenable pour la phylogenèse

L'évolution en génie logiciel est effectuée actuellement par le changement manuel et par la réutilisation. Si on souhaite améliorer un composant en utilisant les algorithmes évolutionnistes, nous devons coder ce composant comme un individu et déduire une fonction d'adaptation qui sélectionne le meilleur après reproduction, croisement et mutation (meilleur peut signifier performant, convivial ou toute autre qualité). Il est communément admis que l'une des tâches les plus difficiles dans l'approche évolutionniste est le codage des individus [Ada 98, Heu 94]. Dans ce contexte, le génome d'un modèle Mage constitue un codage qui nous semble convenable pour l'utilisation de l'approche évolutionniste en génie logiciel.

Selon Dellaert, la majorité des approches phylogénétiques utilisent un mapping direct entre génome et phénotype. Il existe alors une correspondance bijective entre les chaînes de bits du génome et les propriétés phénotypiques du système ou individu qu'on souhaite améliorer. Cette approche présente un inconvénient majeur, celui de la non convergence des algorithmes dès que le nombre de propriétés devient consistant [Del 96]. Dellart ajoute « When interpreted not as a direct encoding but as a set of developmental rules, a genetic description can lead to much more complex morphologies than those achievable with direct mapping ».

1.5.2.7 Réutilisation de l'évolution

Il est communément admis que la meilleure documentation qui permet de comprendre un logiciel est son code source. De ce fait, la meilleure étude de l'évolution subie par un système est celle effectuée sur son code source, changement après changement. L'étude de l'historique de l'évolution d'un système est grandement simplifiée si le système garde lui-même la trace des changements qu'il subit. Cette étude peut éviter certaines erreurs