Utilisation de SysML pour la simulation d'environnements virtuels

Paola Andrea Vallejo Correa

Utilisation de SysML pour la simulation d'environnements virtuels

Réalisé par :
Paola Andrea VALLEJO CORREA

Encadré par :
Ronan QUERREC

5 juin 2012
Master Recherche en Informatique
Systèmes Informatiques Centrés sur l'Humain

Remerciements

Je tiens à remercier les personnes qui m'ont accompagnée de près ou de loin dans ce parcours de formation et m'ont permis de mener, avec autant d'intérêt, ce travail de recherche.

Je remercie particulièrement mon encadrant Monsieur Ronan QUERREC qui a dirigé mon travail de recherche de Master. Je lui remercie pour tout ce qu'il m'a apporté, par leurs conseils, leur patience, leur présence, pour m'avoir fait confiance et m'avoir laissée la liberté nécessaire à l'accomplissement de mes travaux, tout en y gardant un œil critique et en me donnant ses remarques et critiques avec justesse.

Je remercie également Monsieur Pierre DE LOOR par m'avoir accepté au sein du Centre Européen de Réalité Virtuelle.

Résumé

Le CERV utilise la réalité virtuelle pour la formation dans un cadre professionnel (gestion aviation sur porte avions, sécurité routière), dans un cadre universitaire (simulation en biologie et médecine) et dans un cadre scolaire (apprentissage de la lecture). Nous traitons la problématique de réutilisation des outils et des méthodes développées dans le laboratoire via un niveau d'abstraction qui permet d'aborder différents domaines de formation en utilisant les travaux sur les scénarios pédagogiques et les tuteurs intelligents développés au CERV.

L'objectif du stage est de d'accéder aux informations décrivant le système directement à partir des outils de gestion de la vie du produit (PLM) pour renforcer la généricité de ce méta-modèle d'environnements virtuels. Il s'agit ici de leur fournir une implémentation précise et une sémantique opérationnelle dans le cadre des environnements virtuels. Décrire le domaine métier se compare à la conception d'un système et dans notre cas nous nous appuierons sur SysML pour exprimer les objets du système.

Nous proposons :
- Une technique pour la récupération des données graphiques et statiques à partir des outils de PLM ;
- Une méthode pour récupérer les données qui décrivent les comportements ;
- L'intégration des données pour permettre à MASCARET d'utiliser la modélisation du domaine métier tel que données d'entrée pour générer l'environnement virtuel.

Mots-clés : Réalité virtuelle, Méta-modèlisation, Environnement Virtuel pour l'Apprentissage Humain.

Abstract

The CERV uses virtual reality for training in a professional environment, in a university context and in a school environment. We treat the problem of reusing tools and methods developed in the laboratory using a level of abstraction that allows to approach different training areas using the works on the pedagogical scenarios and the intelligent tutors developed in the CERV.

The aim is access to the information that describe the system directly from the management tools of product life cycle management (PLM) to enhance generic programming of this meta model of virtual environments. It is a question of supplying a precise implementation and an operational semantics in the frame of the virtual environments. Describe the business domain is comparable to system design and in our case we will rely on SysML to express system objects.

We propose :
- A technique for retrieving data from static charts and tools of PLM ;
- A method to retrieve the data that describe the behavior ;
- Data integration to allow use MASCARET modeling the business domain used as input data to generate the virtual environment.

Keywords : Virtual reality, Metamodeling, Virtual Environments for Human Learning

Table des matières

Table des figures

Introduction

La gestion du cycle de vie des produits est considérée comme étant aussi importante que les autres types d'activités de l'entreprise comme la planification des ressources, la chaîne de production et les relations avec les clients.

Le PLM (Product Lifecycle Management) [Sta11] intègre plusieurs systèmes (les personnes, les processus, les méthodes, les outils et la technologie) avec le but de gérer le cycle de vie des produits. Le processus de PLM améliore la création et l'échange d'information, ainsi les processus sont plus fiables. De plus, une de principales préoccupations des industriels est d'être capables de réaliser des changements dans les méthodes ou les produits sans perdre les niveaux de production à cause de la formation qu'ils doivent donner aux employés. Ceci nécessite d'améliorer les techniques de formation sur les produits des industriels pour les employés.

Les techniques traditionnelles de formation sont parfois très coûteuses, dangereuses et peu pédagogiques. C'est pour cette raison que les techniques de formation à l'aide de simulateurs et de la réalité virtuelle sont de plus en plus utilisées. Elles permettent de mieux comprendre le système, d'analyser les résultats, de les partager et les utiliser pour améliorer soit le produit comme tel, soit le processus.

Une des difficultés de l'utilisation de la réalité virtuelle est qu'il faut bien représenter le comportement des objets et faire que la sensation de l'utilisateur soit comme s'il interagissait avec le produit réel dans le monde réel. Pour atteindre ce but, il est nécessaire de bien comprendre le produit ou système d'étude ; de prendre toutes ses caractéristiques ; de faire le tri pour ne prendre que les caractéristiques pertinentes pour un cas particulier et finalement les reproduire dans le monde virtuel. Le fait de prendre les caractéristiques, les manipuler et les introduire manuellement à l'environnement virtuel peut prendre beaucoup de temps. Cela est équivalent à réécrire, pour la réalité virtuelle, le modèle qui à déjà été écrit par des experts.

Nous savons que ses caractéristiques peuvent être trouvées dans les outils du PLM, notre intention est d'en profiter et obtenir les données nécessaires pour en faire une simulation qui prenne en compte toutes ses propriétés. Si nous pouvions prendre les caractéristiques physiques et comportementales et les transmettre directement vers la simulation, cela nous permettrai de réduire les efforts de modélisation et de nous concentrer sur la partie de simulation et d'apprentissage.

Ce stage porte principalement sur la transformation des modèles issus d'outils du PLM vers un modèle destiné à la création d'environnements virtuels pour la formation. Il a lieu dans le cadre du projet SIFORAS dont « l'enjeu est de positionner l'humain et le besoin de formation à sa nouvelle place, et proposer un ensemble de méthodes, technologies, et outils, permettant de franchir un haut seuil de productivité, à la fois dans la préparation des contenus pédagogiques, et dans l'assistance aux interventions

... »[1]. Il s'agit d'un projet FUI 11. Le projet fédère 10 partenaires industriels (Nexter Training, Deltacad, DAF Conseil, DELPHI, DCNS, ALSTOM, Virtualys, SNCF, Nexter System, RENAULT) et quatre laboratoires de recherche (ENIB/CERV, INSA Rennes, CEA/LIST, ENISE). L'ENIB est financée par l'OSEO.

Le principe adopté ici est de rendre explicites les connaissances métiers (les modèles définis par les experts métiers) grâce au méta-modèle MASCARET [MQC]. Pour ce méta-modèle, les modèles métiers sont des données d'entrée qui deviennent des connaissances pour le raisonnement des agents. MASCARET fournit une sémantique opérationnelle qui permet de simuler automatiquement le système en réalité virtuelle.

Ce document développe la progression d'une réflexion et de la méthodologie pour obtenir des résultats décrits. Le document est articulé de la manière suivante :

- Dans une première partie, nous situerons le contexte technique et scientifique de ce projet. Nous y introduirons MASCARET et PLM ;
- Dans une seconde partie, nous présenterons les deux principaux axes qui nous permettront d'arriver au modèle de simulation souhaité. Ces axes sont la modélisation du domaine et la modélisation de la partie graphique ;
- Dans une troisième partie, nous présenterons, avec un exemple simple d'application, la méthodologie qui nous permettra d'arriver à un modèle de transformation d'un système réel vers un système virtuel pour la formation ;
- Dans la dernière partie, nous présenterons des conclusions et proposerons des améliorations à la démarche suivie.

1. D'après la définition du projet SIFORAS

Environnement et problématique 1

Les systèmes ne cessent d'évoluer et les contraintes à remplir deviennent plus exigeantes. Les développements au niveau technologique ont comme conséquence l'augmentation de la complexité. Cette complexité est inhérente aux matériaux, aux produits, aux processus et aux moyens d'offrir une formation technique aux employés des entreprises. Il y a de nouveaux enjeux de formation auxquels il faut répondre. Un aspect très important est que la formation doit être efficace et à la fois rapide et elle doit également servir à améliorer la productivité.

L'utilisation des nouvelles technologies est de plus en plus fréquente dans ce domaine. En particulier, la réalité virtuelle est très utilisée pour ces fins, car elle peut fournir une application avec laquelle les utilisateurs peuvent interagir et réagir comme s'il s'agissait d'un environnement réel. La contrainte principale sur la conception du système de réalité virtuelle est de fournir aux utilisateurs un environnement transparent sans qu'ils prennent en compte l'architecture et complexité internes du système.

Le système de réalité virtuelle doit être alimenté par des données provenant directement du système étudié. Favorablement, il existe de systèmes associés au système principal et qui sont capables de gérer tout ce qui est joint à ce système pour garantir le fonctionnement dès la conception jusqu'à la fin de sa vie. Cela veut dire que toutes les informations nécessaires pour concevoir l'environnement virtuel sont réunies au même endroit (le PLM). Le défi est pouvoir les récupérer et les utiliser.

Chaque entreprise a la liberté de définir les paramètres à surveiller et à inclure dans ses systèmes de gestion. Par conséquent, les données sont hétérogènes et changent au cours du temps. Il faudra, en permanence, récupérer les données, identifier lesquelles sont importantes pour la simulation, réaliser les modifications nécessaires et introduire au système de simulation les données correctes. Actuellement, si un changement est réalisé sur le système, il est nécessaire de remodéliser la simulation.

Pour éviter de refaire le système de réalité virtuelle, la représentation des connaissances métiers est faite à travers un niveau d'abstraction supplémentaire proche du métier du modélisateur. Nous proposons d'utiliser le principe de méta-modélisation. Il aide à construire progressivement une représentation d'un point de vue particulier sur des modèles du système d'étude, à les analyser et à définir certaines théories utiles pour

la modélisation d'autres systèmes, à réduire les ambiguïtés et à intégrer les données hétérogènes. La justification de cette proposition est que utilisant un méta-modèle, le modèle est une donnée et l'effort de développement porte sur l'implémentation de ce méta-modèle, cela implique que la tâche de faire évoluer le modèle et l'environnement virtuel est facile et que les connaissances en cours de simulation sont explicites.

Dans le cas de ce projet, il s'agit, de représenter des systèmes composés d'objets en interaction possédant des propriétés géométriques et sémantiques telles que des systèmes : mécaniques, physiques, électriques, entre autres et de leur fournir une implémentation précise et une sémantique opérationnelle dans le cadre des environnements virtuels pour le projet SIFORAS. L'objectif du stage est de permettre aux méta-modèles développés d'accéder aux informations décrivant le système directement à partir des outils de modélisation et de conception du PLM.

Ce projet s'inscrit dans le projet MASCARET (MultiAgent System for Collaborative, Adaptive & Realistic Environments for Training), lequel représente les connaissances du système via le langage de modélisation UML (Unified Modeling Language).

Les enjeux sont :
- d'identifier les données issues des industriels et nécessaires à la pédagogie ;
- d'accéder aux définitions des phénomènes auxquels le système est régit ;
- de leur donner un sens dans le scénario pédagogique afin d'être manipulées en ligne ;
- d'extraire le mieux et le plus automatiquement possible, les données du PLM pour la formation technique ;
- de former mieux, plus rapidement et moins cher.

1.1 Environnements virtuels pour l'apprentissage

Nous aborderons ici la définition et quelques avantages des environnements virtuels pour l'apprentissage.

La réalité virtuelle est la technologie qui fournit des expériences presque réels et/ou crédibles de façon virtuelle ou synthétique. Elle utilise le spectre entier des technologies multimédia actuelles, telles que l'image, la vidéo, le son et le texte, également des nouvelles tendances telles que les stimulus tactiles, visuels et auditifs [Fur08]. « La finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité sensori-motrice et cognitive dans un monde artificiel, créé numériquement, qui peut être imaginaire, symbolique ou une simulation de certains aspects du monde réel. » [FM06].

Les compétences qui peuvent être améliorées au niveau de l'apprentissage avec la réalité virtuelle sont : les habilités motrices, les habilités pour la prise de décisions, la communication et les aptitudes opérationnelles. Comme exemple d'applications qui uti-

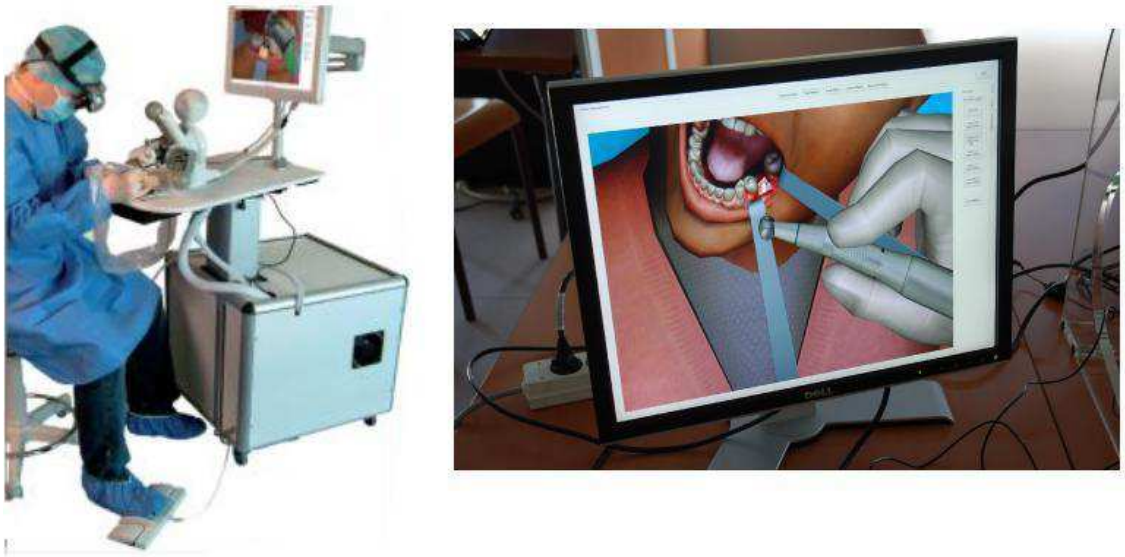


FIGURE 1.1: VirTeaSy

lisent la réalité virtuelle pour la formation nous citons VirTeaSy [CPS+10], un outil de formation à l'implantologie chez les chirurgiens dentistes illustré dans la FIGURE 1.1 et SécuRéVi [Que02], un environnement virtuel de formation pour la sécurité civile illustré dans la FIGURE 1.2.

Un EVAH (Environnement Virtuel pour l'Apprentissage Humain) est un espace virtuel dans lequel un ou plusieurs apprenants sont immergés dans le but d'acquérir du savoir ou du savoir faire. Il est définit comme « un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant. Ce type d'apprentissage mobilise des agents humains [...] et artificiels [...] et leur offre des situations d'interaction [...] ainsi que des conditions d'accès à des ressources formatives. » [TBB+04]. Contrairement à la méthode d'apprentissage traditionnelle (immersion dans un environnement réel), cette méthode offre plusieurs avantages pour atteindre les objectives pédagogiques.

Les avantages de la formation, via la réalité virtuelle ([FM06]) sont :
- Elle élimine le danger, en rendant possible l'exécution de tâches dangereuses sans prendre de risque physique, même si certaines erreurs sont commises ;
- Elle offre plus de probabilités d'apprentissage, car elle rend possible la modélisation des environnements inaccessibles ou inexistants, comme la simulation des conditions difficiles ou de stress ;
- Elle est économique et flexible, puisque l'espace occupé est minimum et la reconfiguration du système autorise son utilisation pour d'autres thèmes d'apprentissage ;
- Elle est adaptative, parce que l'enseignant a la facilité de suivre plusieurs appre-

FIGURE 1.2: SécuRéVi

nants en simultanée dans plusieurs entraînements en sauvegardant les données qui représentent le comportement des élèves ;
- L'enseignant peut faire évoluer l'environnement selon les capacités de l'apprenant et limiter à des fins pédagogiques les actions de l'apprenant ;
- Elle est pédagogique parce que l'enseignant peut manipuler l'environnent, percevoir le niveau de connaissance des élèves, changer les situations, insister sur des situations qui ont posée des problèmes et interagir avec l'élève pour améliorer la qualité des résultats.

Les scénarios pédagogiques que nous voulons développer, doivent refléter les propriétés et comportements du système réel. Ces propriétés sont consignées dans les outils de PLM que nous présentons maintenant.

1.2 PLM (Product Lifecycle Management)

Le but principal du PLM est d'atteindre un équilibre entre le coût global, la performance et la disponibilité opérationnelle. Le PLM permet de diminuer le temps de développement, améliore la flexibilité et la qualité des produits, augmente la productivité, améliore l'utilisation des ressources et réduit les coûts de développement. Le PLM prend en compte les métriques, les personnes, la structure de l'organisation, les méthodes, le matériel, les applications, les données, les processus et les produits ; au cours de la conception, de la définition, de la réalisation du support et de la destruction ou

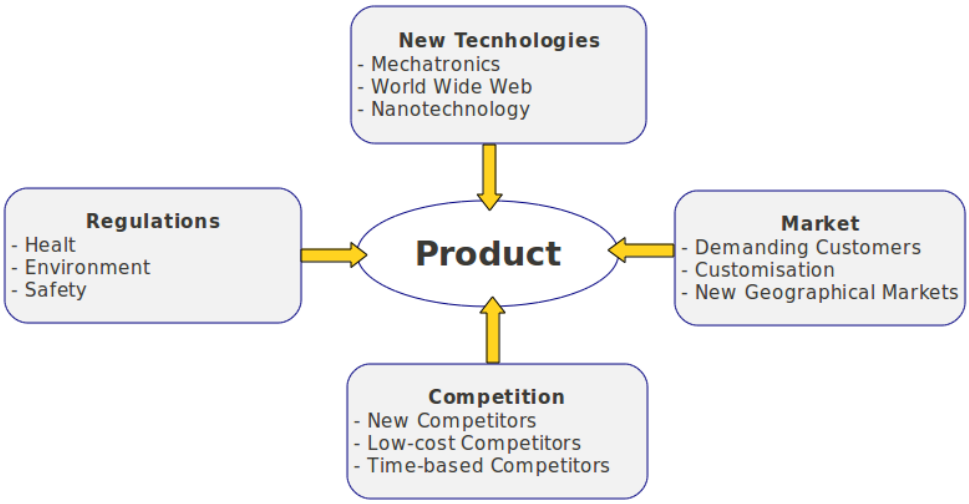


FIGURE 1.3: Le produit est le cœur de l'entreprise. D'après [Sta11]

récupération du produit [Sta11].

Le PLM fixe son attention sur le produit, car le produit c'est le cœur de l'entreprise. Tous les éléments externes doivent contribuer à l'amélioration du produit (voir FIGURE 1.3).

Parmi les applications qui aident à gérer les produits, nous pouvons citer les applications CAO (Conception Assisté par Ordinateur) (FIGURE 1.4) qui sont utilisées pour « traduire une exigence ou un concept dans un design » [Sta11]. Ce type d'application sert spécialement à générer un modèle 3D. Il existe aussi d'autre type d'outils pour définir le fonctionnement du système (FIGURE 1.5).

Nous voulons profiter du PLM pour fournir un environnement virtuel qui soit complet et qui représente de manière correcte le système du monde réel. Nous proposons une alternative qui consiste à prendre les données directement à partir des outils de PLM et transformer ces données pour générer automatiquement la représentation virtuelle, non seulement qu'un point de vue géométrique mais également fonctionnel. Cette procédure permettra de mettre à jour les produits dans l'environnement de simulation sans avoir besoin de refaire toute la scène après chaque changement à niveau des propriétés du produit.

Ayant fait une présentation des points de départ et d'arrivée de notre problème, nous allons continuer avec un aperçu de MASCARET, le méta-modèle qui servira pour simuler l'environnement de réalité virtuelle pour l'apprentissage avec en entrée le PLM.

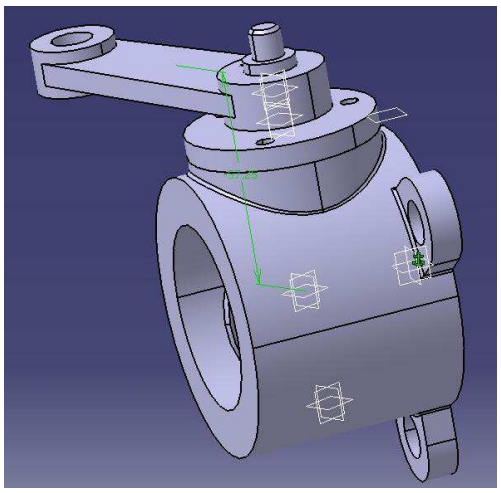


FIGURE 1.4: Vanne modélisée avec le logiciel Catia

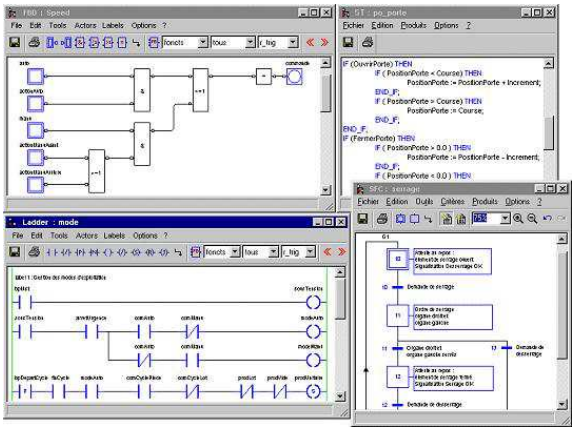


FIGURE 1.5: Modélisation du comportement avec ControlBuild

1.3 MASCARET (Multi-agent System for Collaborative And Realistic Environment for Training)

Le développement des applications se fait fréquemment sans la réutilisation des composants déjà existantes. Cet aspect dépense une grande quantité de temps. Il est nécessaire d'avoir un niveau d'abstraction accessible et compréhensible pour les personnes qui ne sont pas expertes dans le domaine. L'intérêt est d'avoir un modèle qui soit suffisamment générique pour être capables de traiter des problèmes de n'importe quel domaine et de produire l'environnement de formation adapté aux besoins des utilisateurs. Le niveau d'abstraction dont on parle est la méta-modélisation.

Le rôle de la modélisation [KH06] est d'offrir une représentation du problème et des solutions possibles à différents niveaux d'abstraction. Ces solutions doivent servir comme soutien pour appréhender, conceptualiser, concevoir, estimer, simuler, valider et justifier les choix. Les tâches principales de la méta-modélisation sont la conception de nouveaux modèles, l'adaptation de modèles existants et l'intégration d'éléments hétérogènes.

« MASCARET est un méta-modèle permettant de décrire l'environnement virtuel, non pas en tant qu'espace géométrique (tel qu'un graphe de scène) mais en fournissant une sémantique permettant à des agents artificiels ou humains de manipuler une représentation commune de l'environnement et d'y interagir conjointement pour atteindre leurs buts » [MQC]. Il décrit sous la forme de plans ce que l'apprenant peut ou doit faire, en plus il associe cette information avec une session d'apprentissage dans laquelle les propriétés des objets du monde virtuel sont inclus. Il fournit des modèles et des comportements d'agents spécifiques liés à la pédagogie.

MASCARET est une extension d'UML. Ceci lui permet de s'appuyer sur des outils de modélisation et de se servir des nombreux travaux existants sur les transformations de modèles. MASCARET introduit les concepts liés à l'environnement virtuel (entités, propriétés, comportements, etc.), à l'organisation sociale (rôle, équipe, actions, procédure, etc.) et à l'activité humaine dans l'environnement (actions primitives humaines). Une des caractéristiques de MASCARET, est que l'environnement, les agents et les activités sont modélises utilisant le même langage. MASCARET permet la création de modèles métiers décrivant des instanciations de ces concepts, qui peuvent ainsi être utilisés par le formateur pour écrire son scénario pédagogique.

Ce méta-modèle permet la communication bidirectionnelle entre une situation réel d'apprentissage et l'environnement virtuel de formation. Pour établir ce canaux de communication, le scénario pédagogique doit pouvoir faire référence aux concepts qui sont utilisés dans le monde virtuel. D'où l'importance de bien définir tous les concepts dans le domaine spécifique de l'application.

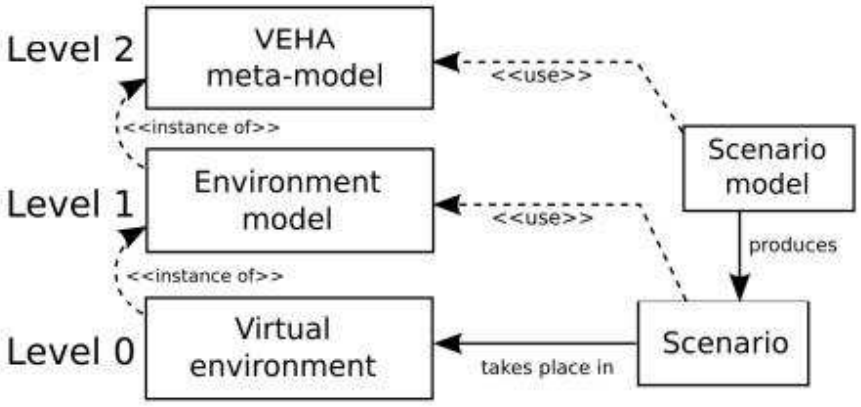


FIGURE 1.6: Représentation du méta-modèle. D'après [CTB+11]

MASCARET réduit le seuil entre le monde de l'ingénierie des systèmes et le monde virtuel. Par rapport aux niveaux d'abstraction, le niveau 0 correspond à l'environnement virtuel ; le niveau 1 représente les concepts utilisées dans le modèle du domaine spécifique le niveau 2 représente les concepts pour décrire un modèle de domaine en général (FIGURE 1.6).

Le processus de développement de mondes virtuels utilisant MASCARET demande quelques données d'entrée, en particulier la description du domaine et la description 3D qui conformeront le monde de simulation. La FIGURE 1.7 représente les entrées de MASCARET.

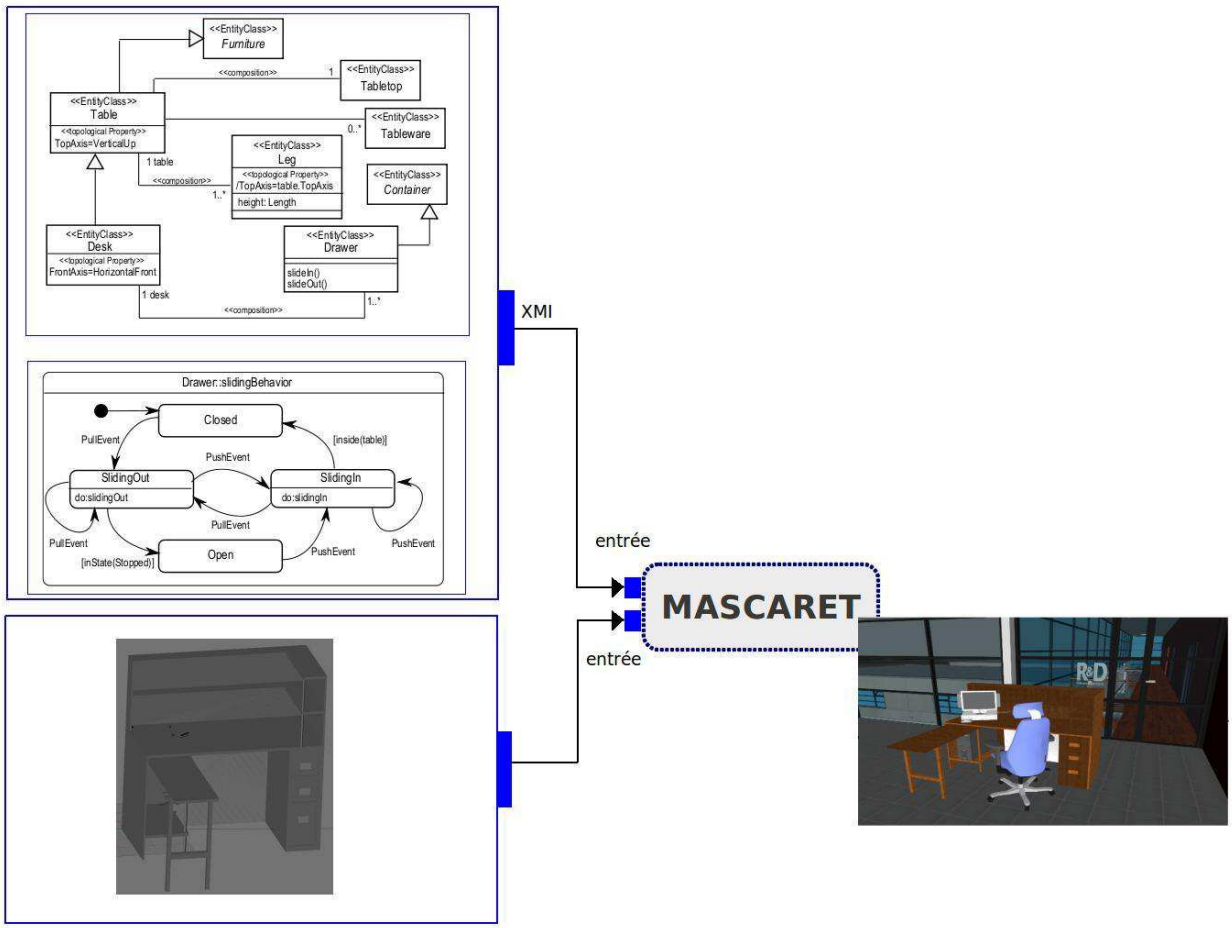


FIGURE 1.7: Entrées pour MASCARET

Dans le chapitre suivant nous allons expliquer comment doit être modélisé le système de simulation et quels sont les moyens existants pour y arriver.

2 Modélisation du système

Connaissant le problème et le contexte du projet, nous essaierons d'illustrer la démarche que nous proposons pour récupérer les données dont MASCARET a besoin pour créer les environnements virtuels pour la formation.

« Le principe est de s'assurer que l'ensemble des données nécessaires à la pédagogie peuvent être exprimées par le modélisateur métier à l'aide des outils de PLM et SLI en amont, c'est-à-dire lors de la conception de celui-ci. En effet, cela permettra d'éviter de devoir renseigner ce modèle plus tard lors de son utilisation en pédagogie. » [Tra08]. On cherche à maximiser la généricité, la simplification et la signification des données. On veut réduire le temps et les coûts associés à la création des scénarios pédagogiques en remplissant les contraintes opérationnelles et de productivité, les contraintes de formation pédagogique et les contraintes d'assistance en la formation.

« Il faut définir formellement comment ces données peuvent s'exprimer dans le métamodèle afin d'assurer suffisamment de généricité pour aborder les différents modèles métiers. Nous proposons alors d'exprimer ces connaissances à l'aide de MASCARET et de l'extension dont il fait l'objet ici grâce à SysML. SysML est un profil spécifique d'UML pour la description de systèmes ce qui correspond à notre objet d'étude » [Tra08].

SysML [Roq09] est un langage de modélisation commun pour les ingénieurs de systèmes. Il est conçu pour modéliser les systèmes complexes, il se base sur UML2, mais offre en plus la possibilité d'exprimer d'autres aspects du système via la modélisation des exigences, la modélisation structurelle, la modélisation dynamique (diagramme d'états pour représenter le cycle de vie des éléments) et la modélisation des équations.

D'abord, nous analyserons les différentes possibilités pour réaliser la récupération des données liées à la modélisation du domaine. En un deuxième temps, nous chercherons des alternatives pour représenter la partie graphique. Pour ces deux aspects, nous choisirons une alternative parmi les différentes possibilités.

2.1 Modélisation du domaine

Quand on parle du domaine, on fait référence aux entités qui forment le monde réel. Ces entités ont des caractéristiques qui les définissent, mais elles ont aussi des comportements.

Nous décrirons ici quelques standards, normes et protocoles existants qui facilitent la transformation de données à partir des outils de PLM. En premier lieu, nous parlerons de ceux qui sont liées aux attributs. Dans la deuxième sous-section, nous nous concentrerons sur ceux qui sont liés aux comportements.

2.1.1 Les entités

En général les langages, les concepts et le vocabulaire utilisés dans le PLM sont internes à l'entreprise, cet aspect empêche d'autres systèmes et parties prenantes de participer à l'enrichissement du produit et de sa gestion. Il devient important d'avoir des standards, des langages et des technologies plus globales et compréhensibles. Par l'exemple, STEP, XML et UML. Nous étudierons également deux standards et un langage pour manipuler les données du PLM.

2.1.1.1 STEP (STandard for the Exchanging Product data)

STEP est un des standards écrits par l'ISO [ISO], il est aussi connu comme ISO103-03 [RVLW04], ce standard définit une méthodologie pour décrire un produit durant tout le cycle de vie et rendre possible l'échange de données entre les systèmes qui interviennent pendant la durée de vie du produit. STEP utilise les systèmes CAE (Computer Aided Engineering Analysis), CAD (Computer Aided Design), CAM (Computer Aided Manufacturing) et CNC (Computerized Numerical Control). Les objets à représenter et à échanger utilisant STEP sont définis en EXPRESS, un langage dont on parlera dans la suite.

2.1.1.2 Langage EXPRESS

La formalisation d'EXPRESS est présentée dans l'ISO10303-11 (Description methods : The EXPRESS language reference manual) [fA12]. EXPRESS est un langage informatique utile pour définir les données formellement et sans ambiguïté, EXPRESS aide à définir les entités, les relations et la cardinalité des relations. L'écriture d'un modèle EXPRESS peut être faite sur une forme textuelle ou graphique (EXPRESS-G).

Les composants principaux d'EXPRESS sont :
- **Entités :** Tout ce qui existe dans le monde réel. Chaque entité est définie comme un ensemble d'attributs ;

- **Types :** Des attributs qui définissent les entités (BINARY, BOOLEAN, LOGICAL, NUMBER, REAL, INTEGER, STRING, LIST, SET, BAG, ARRAY, ENUMERATION, SELECT, NOMMÉS)
- **Attributs :** Définissent les caractéristiques des entités ;
- **Contraintes :** Restrictions des valeurs et types de données à utiliser.

Parmi les usages du langage EXPRESS, on trouve la création automatique du modèle de données pour autre technologie de modélisation de données (EXPRESS schema Mapping). Nous nous intéressons au mapping vers XMI. Le mapping du schéma EXPRESS à OMG XMI (un diagramme de classes UML) est défini. Nous allons présenter les types de donnés qui peuvent être transformées.

2.1.1.3 Mapping de EXPRESS à XMI (ISO10303-25)

Ce standard définit la façon de faire la correspondance des données spécifiées avec EXPRESS vers un langage de modélisation unifié (UML) pour générer des fichiers conformes au standard pour l'échange d'information de méta-données UML basé sur XML (XMI - XML MetadataInterchange) [feiaf12]. Il était créé spécifiquement pour permettre aux ingénieurs de systèmes d'utiliser UML pour réutiliser les schémas modélisés avec EXPRESS. Le tableau de la Figure 2.1 présente les transformations les plus importants pour aller d'EXPRESS vers UML.

On voit ici que seul les aspects statiques du système sont pris en compte et que le seul transformation XMI possible est vers un diagramme de classes. Nous cherchons un standard plus étendu pour pouvoir transformer tous nos données. Le standard PLCS sera étudié.

2.1.1.4 PLCS (ISO10303-239)

Le PLCS (Product Life Cycle Support) [CUB12] est un standard international des spécifications d'un modèle d'information adaptable aux différents secteurs de l'entreprise. Ce modèle flexible définit quelle information peut être échangée (à l'intérieur de l'entreprise et avec les clients) et représentée pour soutenir un produit pendant toute la durée de sa vie. Le PLCS assure la consistance, la maintenance et l'interchangeabilité des données entre les systèmes d'information durant l'ingénierie des systèmes, et les phases d'intégration et mise en service. La définition du modèle est fournie en utilisant le langage EXPRESS.

Le PLCS est supporté par OASIS (Organisation for the Advancement os Structures Information Standards). OASIS développe et publie les DEX (Data EXchange specifications) et des protocoles de transfert de données. Comme protocole de transfert, PLCS utilise l'ISO10303-28 (Implementation methods : XML representations of EXPRESS

EXPRESS schema mappings		
EXPRESS	***UML***	***Name in UML***
schema	Package	schema's name
EXPRESS simple data type mappings		
EXPRESS	***UML***	***Name in UML***
simple data type	Class	data type's name
string	String	
integer	Integer	
Boolean	Boolean	
logical	Class	logical
real	Double	
number	Double	
binary	Class	binary
reference to simple data type	reference to the corresponding Class	
EXPRESS entity data type mappings		
entity data type	Class	entity's name
EXPRESS attribute to UML Attribute		
EXPRESS	***UML***	***Name in UML***
attribute with a simple data type as its base type	Attribute defined within the UML Class mapped from the containing EXPRESS entity data type	explicit attribute's name
EXPRESS attribute to UML Association		
attribute with a single or SET OF data type as its base type	Association at the same level as UML Class mapped from the containing EXPRESS entity data type	explicit attribute's name

FIGURE 2.1: Transformation de EXPRESS à UML

schema and data), celui spécifie plusieurs formats des fichiers d'échange en XML.

Étant donné que le champ d'application de PLCS est suffisamment étendu et que quasiment toujours il suffit d'utiliser une partie spécifique du modèle pour obtenir la fonction souhaitée, il est nécessaire d'avoir un mécanisme de partition.

DEX (Data EXchange specifications) :

Le DEX est le mécanisme qui fractionne le PLCS en parties orientées à un domaine spécifique. Il est développé en DEXLib (un document basé sur XML). « Un DEX est une manière de diviser le modèle d'information ISO 10303-239 en sections adaptées pour un processus métier particulier. Un DEX fournit un sous-ensemble du modèle d'information et un guide d'utilisation » [Sup12].

L'étude menée précédemment nous a permis de comprendre qu'à partir des fichiers XML générés par PLCS et l'utilisation du langage EXPRESS, nous pouvions arriver à construire un diagramme de classes UML. Ce diagramme contiendra les entités et les relations d'héritage et association, par contre, il n'est pas possible de représenter les comportements parce que cette information n'est pas présente dans le PLCS. Mais on sais que les outils de PLM et modélisation 3D comme Catia sont conformes à ces standards. Nous pouvons donc récupérer certaines données directement à partir de ces outils, sures que les contraintes de formats et transfert seront respectées. La raison par laquelle nous nous intéressons à ce type d e protocoles basées en XML est parce que MASCARET reçoit comme paramètres des fichiers XMI.

2.1.2 Les comportements

Maintenant, la partie de récupération des données pour le traitement des comportements va être explorée. Nous cherchons concrètement une stratégie pour exprimer les procédures de nos systèmes.Nous avons trouvé que ControlBuild [fES12] est un logiciel qui facilite la gestion du cycle de vie des produits, il gère spécialement l'aspect comportemental.

ControlBuild est un atelier logiciel qui s'oriente vers la partie de contrôle/commande des systèmes automatisés. Cet atelier est un ensemble de modules qui aident à gérer le cycle de vie d'une application. ControlBuild supporte le format standard XML définit par le PLCopen, il supporte aussi tous les langages définis par la norme IEC 61131-3 et intègre des générateurs de code pour les automates programmables (Schneider Unity, Siemens STEP7, Rockwell Rslogix).

Les bases de ControlBuild seront explorées postérieurement. Elles vont nous servir pour mieux comprendre le fonctionnement des outils de définitions des comportements des systèmes et pour commencer à développer la stratégie de prise et transformation de données.

2.1.2.1 La Norme IEC61130-3

L'IEC (International Electrotechnical Commission) s'occupe de promouvoir la coopération internationale pour la création des standards dans les domaines électrique et électronique. Elle gère les activités qui exigent du réflexe plutôt que des calculs élaborés. Une de ses normes est l' IEC 61131. Cette norme est conformée de cinq parties : Informations générales, Spécifications et essais des équipements, Langages de programmation, Guide pour l'utilisateur et Communications.

Notre attention se concentrera sur la partie *Langages de programmation*. Elle définit la grammaire, la syntaxe et la sémantique de cinq langages de programmation de contrô-

leurs programmables. Ce standard nous permettra d'avoir un fichier XML qui contient les fonctionnalités d'un automate. Chaque langage a une représentation visuelle différente, cependant il existe des éléments qui sont communs à tous les langages, tels que les identificateurs, les mots clés, les commentaires, les libellés, les types de données et les variables. Les langages sont :

Langages de programmation textuels :
– IL (Instruction List) : Liste d'instructions
– ST (Structured Text) : Langage de Texte structuré

Langages de programmation graphiques :
– LD (Ladder Diagram) : Langage à contacts
– FBD (Function Block Diagram) : Langage de Boîtes fonctionnelles

Langage structural :
– SFC (Sequential Function Chart) : Structure de l'organisation interne du programme

2.1.2.2 Le PLCopen

« PLCopen est un organisme international qui promeut l'utilisation des langages définis par la norme IEC IEC61130-3 » [feiaf12]. Le PLCopen découple le matériel du logiciel et à la fois établit un lien entre le système principal et d'autres logiciels externes. Il est indépendant du logiciel et du produit. Le PLCopen est constitué des comités suivants : PLCopen benchmarking, PLCopen certified training, PLCopen conformity level, PLCopen motion control, PLCopen reusability level, PLCopen safety et PLCopen XML. Du PLCopen nous ne nous intéresserons qu'aux schémas XML du comité PLCopen XML et au langage SFC ; il seront décrits dans la suite.

Le PLCopen XML ou TC6 :

L'objectif est de permettre l'échange de données entre différents programmes. Le PLCopen XML essaie de démontrer qu'il est possible de transférer les informations entre logiciels sans perte de données. Ce principe est aussi ce que nous cherchons pour pouvoir recevoir des informations prévenantes des outils de PLM et les utiliser sur MASCARET (avec une couche intermédiaire qui fasse une transformation qui facilite l'échange dans les deux sens). Le fichier PLCopen_XML contient tout ce qui est dans le système et la structure du fichier peut être validée à l'aide des schémas fournis par le comité XML.

PLCopen XML inclut les langages de programmation standard, des commentaires sur le projet, information graphique comme la taille des blocs, la position des composants sur l'écran et les lignes de connexion entre blocs, et d'autres informations importants par rapport au système. Cela veut dire que le fichier contient tout ce qui est dans le système

et la structure du fichier peut être validée à l'aide des schémas fournis par le comité XML.

Les formats proposés par TC6 sont spécifiés à travers des schémas XML. Ces schémas définissent les caractéristiques des éléments qui font partie de la structure du projet et la fonctionnalité des POUs (programmes). D'abord, dans le schéma les éléments qui sont communs à tous les langages graphiques sont inclus. Puis, dans la partie de définition de chacune des grammaires des langages, les caractéristiques de ces éléments sont personnalisés. Chacun des éléments du langage est représenté comme un élément du schéma XML, dont leurs attributs font référence aux caractéristiques de fonctionnalité et non à la représentation graphique sur l'écran.

SFC (Sequential Function Chart) :

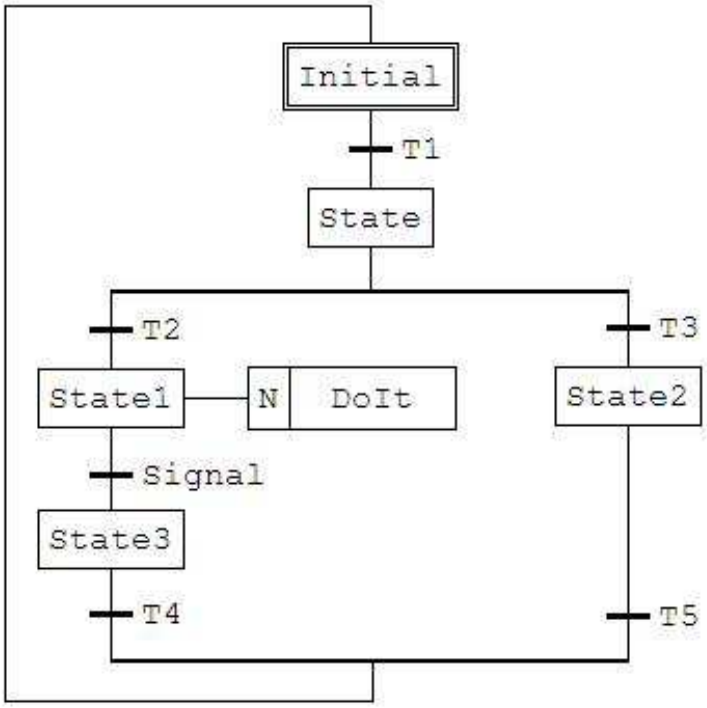


FIGURE 2.2: Exemple de diagramme SFC

Même si nous pouvons utiliser n'importe quel langage (IL, ST, LD, FBD ou SFC), le langage choisi dans notre cas est SFC. Ce langage est celui qui se ressemble le plus au type de diagramme (états-transitions UML) que nous utilisons dans MASCARET.

Les diagrammes de fonctions séquentielles structurent les tâches séquentielles d'une application à travers des programmes et des blocs de fonctions. La programmation en SFC peut être textuel ou graphique.

Les éléments d'un diagramme SFC sont :
- **Step :** Il indique l'état du système à un moment donné. Il a souvent une ou plusieurs actions associées ;
- **Initial step :** Représente l'étape initiale ;
- **Action :** Est un processus à exécuter. Il est possible de définir plusieurs actions et leur ordre d'exécution ;
- **Transition :** Elle Permet de passer d'une étape à une autre quand les conditions sont atteintes. Les conditions peuvent être exprimées à l'aide des autres langages ;
- **SelectionDivergence :** Elle indique que le graphe se décompose en plusieurs chemins selon un choix conditionnel. Un seul chemin est suivi ;
- **SelectionConvergence :** Elle représente que plusieurs chemins du graphe convergent vers un seul chemin ;
- **SimultaneousDivergence :** C'est la représentation du parallélisme. Plusieurs activités peuvent se dérouler en même temps ;
- **SimultaneousConvergence :** Elle permet de continuer avec le déroulement d'une seule activité après avoir fini plusieurs activités qui se déroulaient en parallèle.

On connait quels sont les éléments du langage et on peut déjà penser au mapping d'un diagramme SFC vers un diagramme états-transitions. Maintenant, on a besoin d'un outil qui nous permette de visualiser les diagrammes SFC, les modifier et récupérer les données sémantiques. L'outil que nous utiliserons pour atteindre cette finalité est Beremiz.

Beremiz :

Beremiz est un logiciel libre développé en python, dont le méta-modèle se base sur le schéma officiel de TC6 (XSD) et qui génère les fichiers XML associés aux automates. Il permet de construire des automates selon les normes IEC61131 et PLCopen en utilisant les cinq langages standard. Cet éditeur ne fait pas la vérification du code. Nous utiliserons cet éditeur pour construire la représentation en langage SFC selon les machines à états décrites en UML.

Par rapport à la partie de comportements du système, nous pouvons conclure que PLCopen sera pris comme alternative pour modéliser le comportement du système. PLCopen nous offre la possibilité de représenter les comportements des entités et de les exprimer sous la forme d'un fichier XML. Pour réaliser les tests, nous prendrons seulement le langage SFC. Le fait d'avoir un diagramme qui ressemble au diagramme souhaité est très intéressant pour nous parce que nous savons que toutes les données dont nous avons besoin sont présentes. Il faut noter que Beremiz n'est qu'un des éditeurs de PLCopen, nous l'avons choisi parce que c'est un logiciel libre et facile à utiliser.

2.2 Synthèse

En ce qui concerne la modélisation des entités du domaine, nous avons identifié que les données les plus importantes au niveau de PLM sont les noms des entités, les attributs et les types de relations existants entre les entités (pour nous, une entité sera une partie d'un produit). Nous allons créer un script (code) capable d'accéder à Catia et utiliser les fonctions définies dans l'API pour prendre des données. Avec les données récupérées, nous conformerons un fichier XMI qui représentera un diagramme de classes UML. Nous n'utiliserons pas directement le langage EXPRESS parce qu'il est un langage assez vieux et qui a encore des restricciones par rapport à la représentation de certains aspects du domaine. Mais, nous allons utiliser PLCS (dont la basse est EXPRESS), un protocole plus étudié et étendu au niveau des données qu'il permet de traiter.

Par rapport aux comportements, nous ne sommes pas encore capables de les récupérer à partir de Catia, en conséquence nous partirons du principe que le fonctionnement a été défini sous la norme de PLCopen ainsi nous développerons aussi un code pour prendre les fichiers issus de Beremiz (dans le cas de ce projet) et les transformer vers un fichier XMI. Ce fichier représentera un diagramme d'états-transitions UML.

Nous avons besoin de gérer l'environnement 3D qui intègrera le système cible de la formation. Pour atteindre cet objectif nous proposons de nous en servir de l'outil de CAO (Conception Assistée par Ordinateur) avec laquelle le système avait été dessiné. CATIA (Conception Assistée Tridimensionnelle Interactive Appliquée), par exemple, est un logiciel utilisé par tout type d'entreprises, les petites sociétés et les grandes entreprises. Il aide à la conception 3D, mais il offre aussi une approche PLM, pour cette raison, plusieurs entreprises l'utilisent comme moyen pour garantir la réussite de ses produits. Il est plus qu'un outil de CAO [Sys12].

CATIA fournisse des fichiers définis en VRML. Ces fichiers contiennent toute l'information graphique nécessaire. VRML (Virtual reality Markup Language) est un langage pour décrire les mondes virtuels en 3D. Ce type de fichiers peuvent être crées dans un éditeur de texte, bien que les logiciels de conception assistée par ordinateur ; les programmes de modélisation et d'animation et les logiciels de VRML [GPV].

Finalement, nous rendrons deux entrées à Catia. La première entrée correspondra à l'intégration des fichiers XMI (entités et comportements). La deuxième entrée correspondra aux fichiers VRML générés par Catia.

Implémentation 3

Dans cette partie, nous définirons une méthode pour extraire les données. Nous définirons aussi, quelles transformations appliquer aux données pour obtenir une maquette numérique optimale (pertinence de la description/volume des données). « Les principaux problèmes à traiter sont la sélection intelligente des composants utiles dans l'arbre produit, la simplification. La gestion des informations et des connaissances qui se créent pendant la durée de vie du système, sont très importants et pour cette raison elles doivent être bien gérées » [Tra08].

Grâce à un exemple, nous illustrerons la démarche suivie. Comme nous l'avons annoncé dans le chapitre précédent, la solution sera divisée en deux parties (domaine et 3D). D'abord, nous allons modéliser le domaine. Ensuite, nous modéliserons la partie graphique. La modélisation du domaine sera aussi divisée en deux parties, la première correspond à la modélisation des entités et la deuxième porte sur la modélisation des comportements.

Nous aurons donc, deux entrées pour MASCARET, l'une corresponde au fichier résultant de la modélisation du domaine. L'autre correspond aux fichiers issus de Catia et qui décrivent la partie graphique.

3.1 Domaine

Pour obtenir le fichier XMI qui décrit le domaine du système à simuler, nous allons d'une part réaliser une transformation de Catia vers un fichier XMI qui représente un diagramme de classes. D'autre part, nous allons réaliser une transformation d'un diagramme SFC vers un diagramme d'états-transitions. Finalement, nous allons faire une intégration des deux fichiers pour constituer le premier paramètre d'entrée de MASCARET.

3.1.1 Transformation de Catia vers XMI

La stratégie consiste à trouver une façon de récupérer les données à partir des objets modélisés en Catia. Et avec ces données, générer un diagramme de classes (fichier XMI).

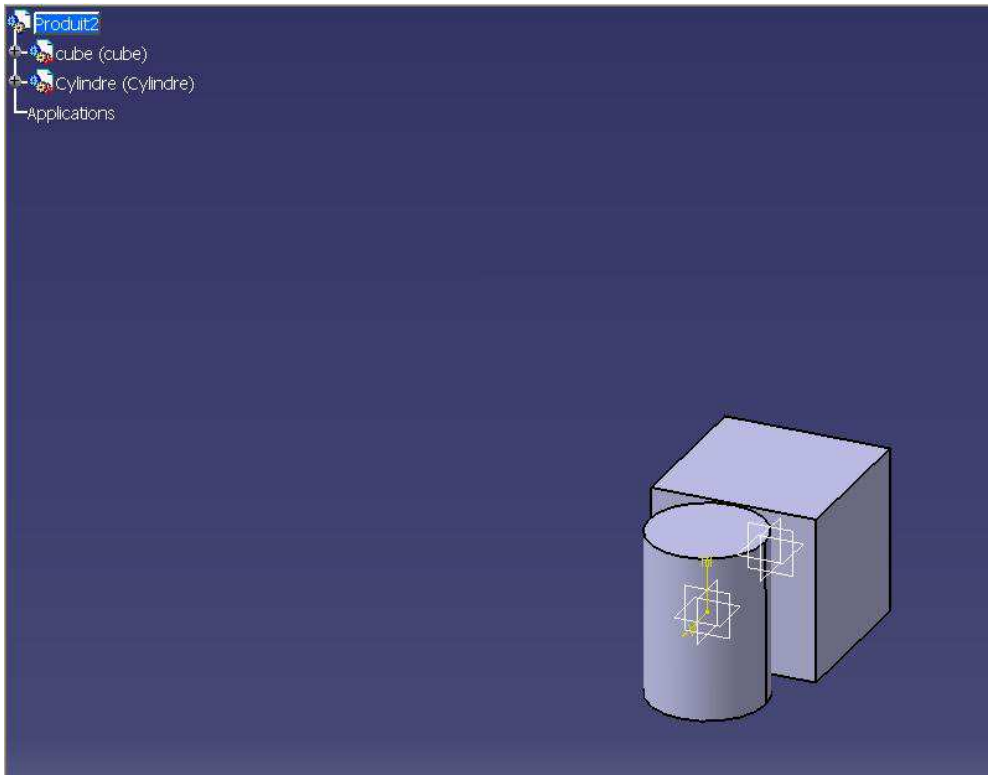


FIGURE 3.1: Objet modélisé avec Catia

Nous disposons d'un objet modélisé en Catia, nous avons appris qu'il existe des méthodes fournies par Catia. Ces méthodes peuvent être utilisées pour développer des programmes capables d'accéder à certains caractéristiques des objets. Nous avons développé un programme qui prend les noms des parties de l'objet graphique, les attributs (non graphiques) définis pour chaque partie et les types de relations entre les parties. Cette information sera retenue et utilisée pour constituer les balises et informations du fichier XMI.

Le code est chargé d'accéder à l'arbre de l'objet, de détecter toutes les parties existantes et de remplir le fichier XMI, ainsi chaque partie deviendra une Classe UML. Ensuite, nous parcourrons chaque partie pour connaître si elle contient des attributs ; s'il y en a, le fichier XMI sera enrichi. Finalement, il détecte le type de relation entre chaque pair de composantes et les représente encore sur le fichier XMI.

Les données Catia devienent des éléments du diagramme UML. Les transformations qu'on fait sont illustrées dans le tableau de la FIGURE 3.7.

L'objet de la FIGURE 3.1 contient deux parties, un cube et un cylindre. Le cube n'a pas d'attributs. Le cylindre a un attribut de type *Integer* et dont le nom est *Heure*. Il

Mapping de Catia vers UML	
Catia	***UML***
Product	Package
Product name	Package name
Part	Class
Part name	Class name
Property	Attribut
Property name	Attribut name
Property type	Attribut type
Property value	Attribut value
Distance contraint	Association relation
Inside contraint	Composition relation

FIGURE 3.2: Transformation de Catia vers UML

y a une relation d'association définie entre le cube et le cylindre. Nous espérons donc, qu'après exécuter le code, le fichier XMI résultat devra contenir deux classes (cube et cylindre). La classe cylindre possédera l'attribut Heure, tandis que la classe cube n'aura pas d'attributs. Une relation d'Association fera le lien entre ces deux classes.

La FIGURE 3.3 expose un bout du pseudo-code qui décrit le processus suivi pour obtenir le diagramme de classes à partir des objets modélisés en Catia.

Nous avons exécuté le code, lequel a bien généré un fichier XMI. Pour vérifier que la structure du fichier est correcte et que le contenu du fichier correspond avec ce qu'on attendait, nous allons utiliser *Modelio*. Modelio est un outil de modélisation de diagrammes UML, il offre la possibilité d'importer des fichiers XMI et d'afficher les diagrammes. Nous avons importé notre fichier XMI et le résultat obtenu peut s'apprécier dans FIGURE 3.4. Nous confirmons que le résultat est correct par rapport à ce qu'on attendait.

Un deuxième exemple (FIGURE 3.5), illustre la relation de composition entre deux classes (FIGURE 3.6).

3.1.2 Transformation de SFC vers XMI

Nous avons déjà le diagramme de classes requis par MASCARET, il faut construire le diagramme d'états-transitions. Pour ce diagramme, nous allons prendre un fichier XML issu d'un diagramme SFC (Grafcet) qui contient le fonctionnement du système. Ce fichier XML sera parcouru pour prendre les éléments dont nous avons besoin pour créer le nouveau diagramme. Ces éléments sont le nom des états, les signaux qui provoquent une transition d'un état à un autre, les actions exécutées lors du passage par un état, l'état initial et l'état final. Ayant les éléments nécessaires, le code *Transformateur* générera le

```
...
   Document = read CatiaDocument                // lecture du fichier Catia
   XMI = create xmiFile                         // création du fichier XMI
   open Document                                // ouverture du Document Catia
   Product = search product in Document         // recherche du produit dans le document
   open Product                                 // ouverture du produit
   IF Product has Parts
     FOR parts in Product                       // parcours des parties du produit
       Part  = open Part                        // prise des caractéristiques de la partie
       PartName = Part.Name
       PartId = Part.Id
       Class = tranform Part to Class           // transformation de la partie
                                                   vers une Class
       IF Part has Properties
         FOR properties in Part
           PropertyName = Property.Name
           PropertyId = Property.Id
           PropertyType = Property.Type
           PropertyValue = Property.Value
           Attribut = transform Property to Attribut  // transformation de la proprieté
                                                         vers un attribut
           print Attribute in Class                   // ajout des attributs à la classe
         END FOR
         print Class in XMI
       ELSE
         print Class in XMI                       // écriture des données sur le
                                                     fichier XMI
       END IF
     END FOR
   END IF
   Contraint = search contraint in Product          // recherche des contraintes
   ...
   Relation = Transform Contraint in Relation       // transformation de contraintes in relations
   transform Contraint
 ...
```

FIGURE 3.3: Pseudo-code

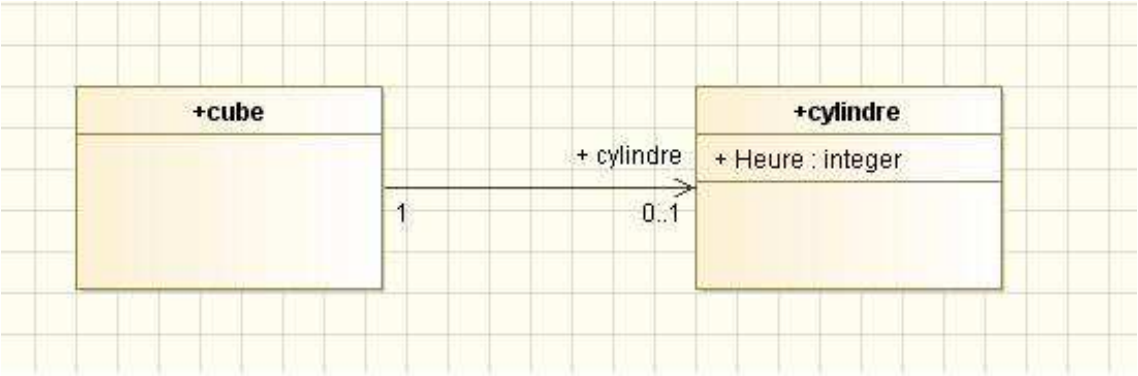


FIGURE 3.4: Relation d'association

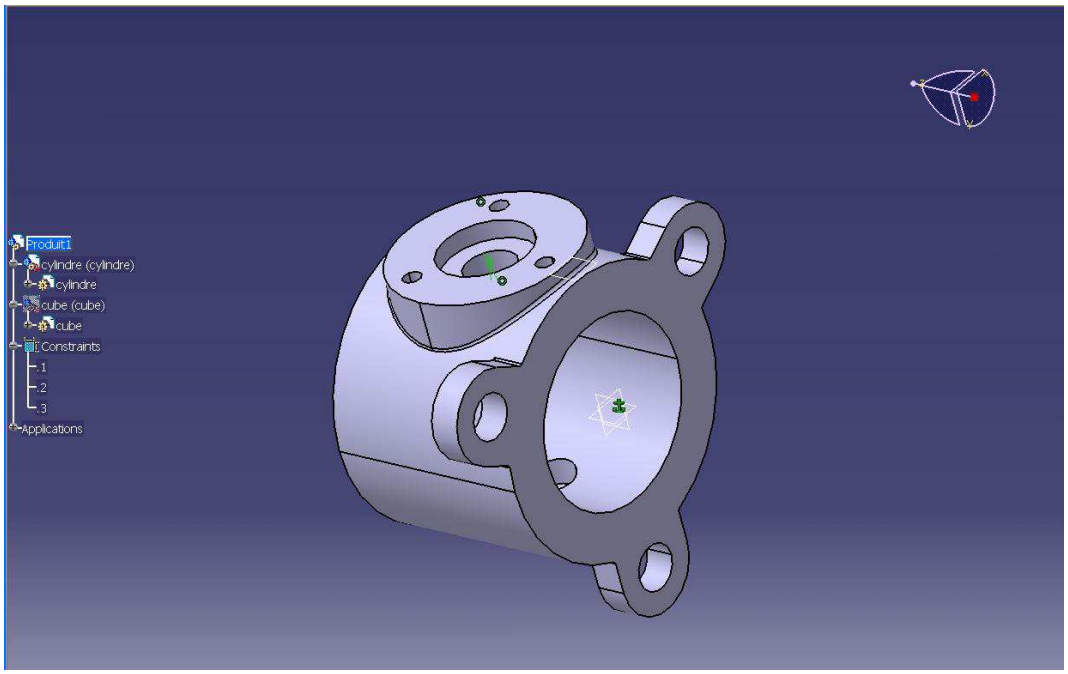


FIGURE 3.5: Vanne modélisé avec Catia

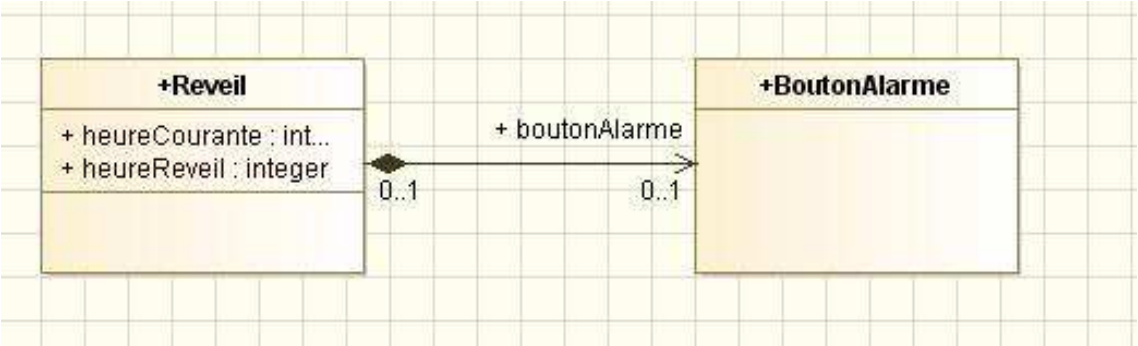


FIGURE 3.6: Composition

fichier XMI qui représente le diagramme UML.

La FIGURE 3.7 montre la correspondance que nous utilisons pour passer du diagrammes SFC vers un diagramme UML.

Mapping de SFC vers UML	
SFC	***UML***
Step	State
Initial step	Initial step
Action	Action
Transition	Transition
SelectionDivergence	Fork
SelectionConvergence	Join
SimultaneousDivergence	Fork
SimultaneousConvergence	Join

FIGURE 3.7: Transformation de SFC vers UML

Par l'exemple, nous avons le fichier XML qui correspond avec le diagramme SFC de la FIGURE 3.8. Ce diagramme définit le comportement d'une prise programmable dans sa phase de configuration initiale. Nous donnons ce fichier comme entrée au Transformateur ; après son exécution, nous devrons avoir comme sortie, un fichier XMI. Pour vérifier que le fichier est bien constitué et qu'il contient tous les éléments dont on a besoin, nous allons utiliser Modelio pour l'importer et visualiser le diagramme. Le diagramme états-transitions génèré est illustré dans la FIGURE 3.9

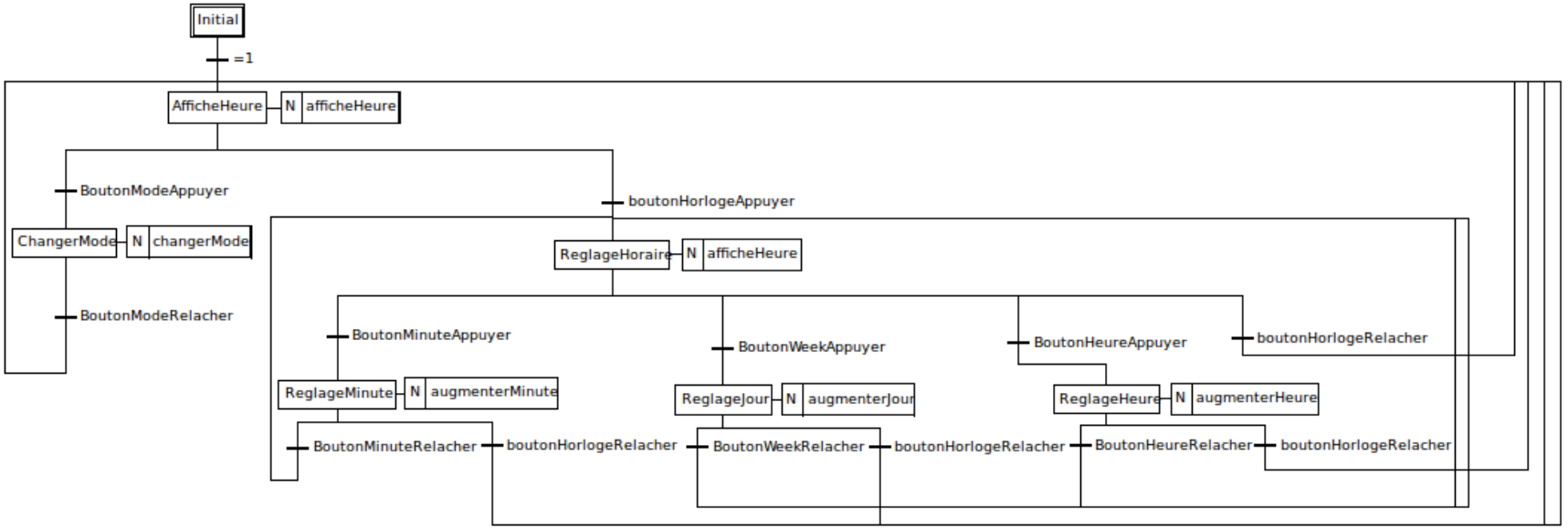
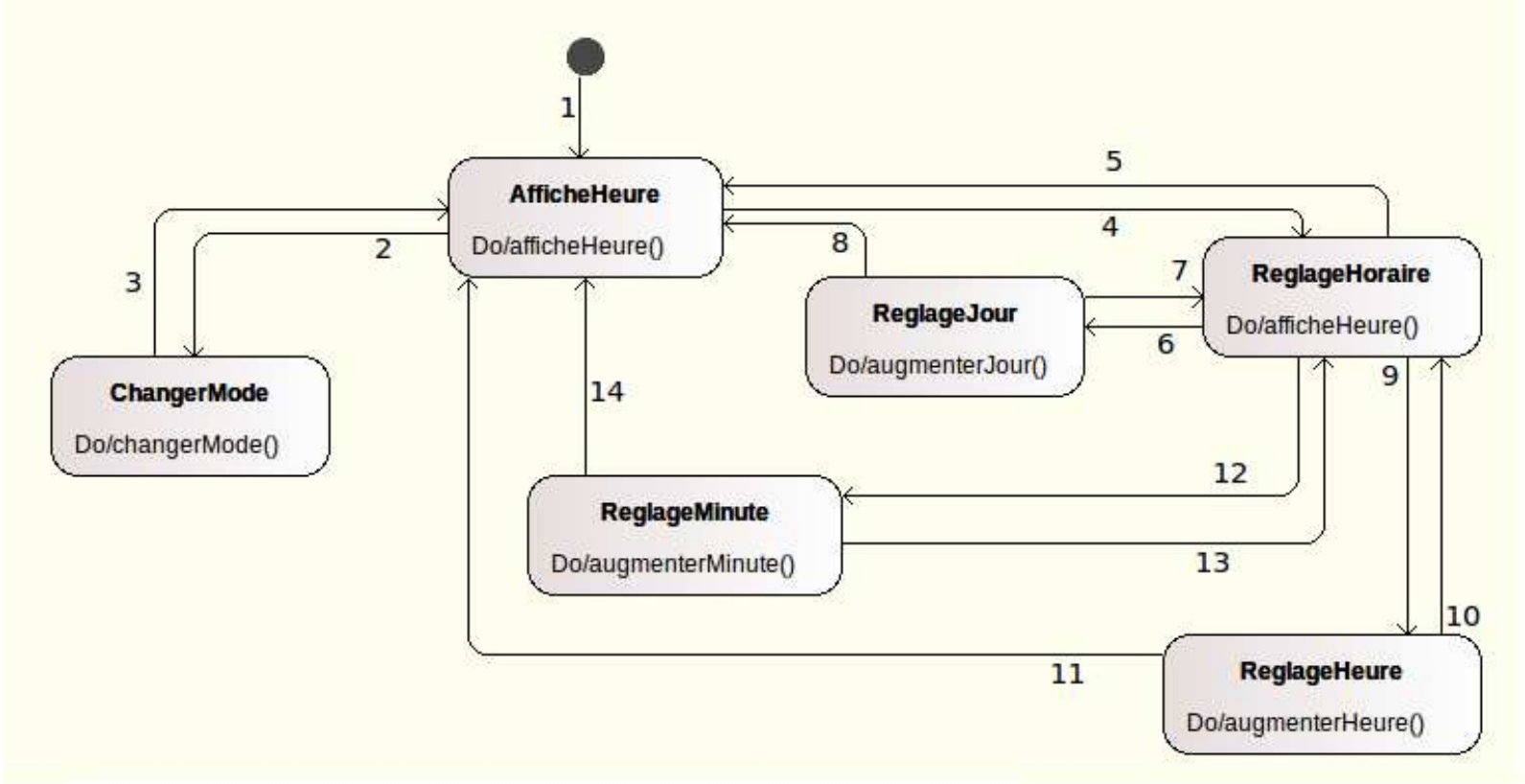


FIGURE 3.8: Diagramme SFC



1 : null
2 : BoutonModeAppuyer
3 : BoutonModeRelacher
4 : boutonHorlogeAppuyer
5 : boutonHorlogeRelacher
6 : BoutonWeekAppuyer
7 : BoutonWeekRelacher
8 : boutonHorlogeRelacher
9 : BoutonHeureAppuyer
10 : BoutonHeureRelacher
11 : boutonHorlogeRelacher
12 : BoutonMinuteAppuyer
13 : BoutonMinuteRelacher
14 : boutonHorlogeRelacher

FIGURE 3.9: Diagramme états-transitions UML

3.1.3 Intégration des résultats

L'idéal serait d'intégrer les fichiers résultat des deux démarches précédentes pour n'avoir qu'un seul fichier de modélisation du domaine. L'intégration consiste à mettre les comportements dans les classes.

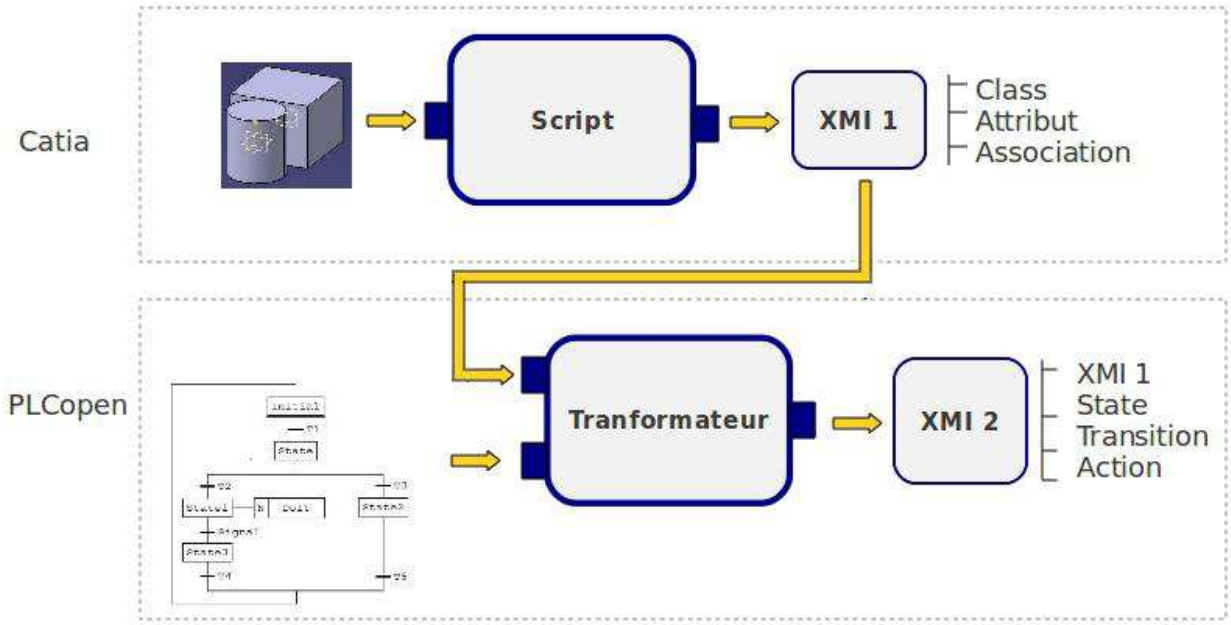


FIGURE 3.10: Intégration du model du domaine

Ce processus n'a pas été développé, mais nous avons pensé à la manière de le faire. D'abord, nous générerons le fichier XMI des classes. Le code Transformateur de SFC vers XMI sera modifié pour lui donner deux paramètres d'entrée, le fichier XML de SFC et le fichier XMI des classes. Le Transformateur écrira sur le fichier XMI en ajoutant les comportements dans les classes. La FIGURE 3.10 représente comment sera faite cette intégration.

3.2 Modélisation graphique 3D

La modélisation de l'environnement graphique est plus facile que la modélisation du domaine. Ici nous n'avons pas de transformations à faire puisque MASCARET est capable de recréer le contenu du fichier *.wrl* (VRML) dans l'environnement virtuel de formation. Catia génere des fichiers *.wrl*.

Conclusions et perspectives

Le but principal du projet est la réalisation d'un système logiciel pédagogique de formation technique, complet et intégré aux processus industriels pour améliorer la qualification des employés et la productivité de l'entreprise. Ce but a plusieurs enjeux.

Le premier enjeu c'était d'identifier les données issues des industriels et nécessaires à la pédagogie. Nous avons trouvé que chaque entreprise gère des données différentes et que le système de formation dépend aussi de ce que l'enseignant souhaite. Néanmoins, nous avons reconnu que les données les plus utilisées sont les noms des parties des produits, leurs identifiants uniques, leurs attributs et les types de relations existantes entre les parties, car ces relations permettront de reconstruire l'objet comme un ensemble. Les comportements et états auxquels peut arriver un objet sont aussi essentiels pour la création des scénarios pédagogiques. Il ne faut pas oublier la partie graphique parce que nous cherchons que l'environnement virtuel soit suffisamment proche du réel.

Par rapport à l'enjeu d'accéder aux définitions des phénomènes auxquels le système est régit. Nous avons exploré un des standards le plus utilisé (PLCopen) par des industriels pour décrire les comportements de leurs systèmes. À partir des informations découvertes, nous avons appris qu'il est possible de récupérer des fichier en format XML duquel on peut facilement extraire des données basiques, mais très importantes tels que les états d'un objet, les actions à exécuter lors du passage par l'état et les signaux qui permettent d'atteindre un état particulier.

Ayant les données, il fallait leur donner un sens dans le scénario pédagogique afin d'être manipulées en ligne. Avantageusement, nous pouvons utiliser MASCARET, un méta-modèle capable de créer et gérer les scénarios pédagogiques à partir des données statiques, graphiques et dynamiques des objets.

Compte tenu de l'enjeu d'extraire le mieux et le plus automatiquement possible, les données du PLM pour la formation technique. Nous pouvons affirmer que les outils de modélisation 3D et de PLM tels que Catia incluent et utilisent des protocoles et standards capables de fournir des informations avec un format acceptable par d'autres outils, ils facilitent l'extraction des données. L'interopérabilité entre STEP, XML et UML est garantie, c'est-à-dire qu'il est possible de faire les transformations nécessaires pour obtenir le format de données requise par MASCARET.Nous avons identifié quelles sont les données de PLM dont on a besoin, nous avons identifié la façon de récupérer les caractéristiques comportementales des objets. Nous pouvons, donc les récupérer et les transformer au format demandé par MASCARET.

Les entreprises ne prennent pas en compte le mêmes données et les données peuvent changer selon le scénario pédagogique. Nous avons établi un mécanisme pour récupérer les données directement des outils de PLM, si les données changent, le mécanisme ne changera pas, ce fait facilite l'automatisation dans la prise de données. Nos avons aussi défini les transformations nécessaires pour intégrer correctement les données, pour gérer leur hétérogénéité et constituer des environnements virtuels pour la formation technique.

Quant à l'enjeu de former mieux, plus rapidement et moins cher. Les avantages de la formation en utilisant la réalité virtuelle aident à atteindre ce but. De plus, la prise de données directement à partir des systèmes de PLM, réduit le temps de re-modélisation de l'environnement, réduit les erreurs humaines et permet de mettre à jour les données sans avoir besoin de reprogrammer toute la simulation parce qu'on réduit le seuil entre le monde de l'ingénierie des systèmes et le monde virtuel.

Notre idée c'est d'utiliser SysML comme langage pour modéliser les besoins du client et les comportements des entités du système. Comme outil pour créer les environnements virtuels, nous envisageons d'utiliser MASCARET. MASCARET permet de créer des environnements virtuels riches à niveau de son contenu sémantique. MASCARET prendre en compte la structure de l'environnement, le comportement des agents et l'interaction entre eux et les humains.

Par rapport au travail à réaliser, il manque encore le développement de l'intégration des différents fichiers qui conforment la modélisation du domaine.

Les perspectives portent sur le développement d'un mécanisme de retour d'expérience pour modifier les produits de basse à partir des modifications réalisées pendant la simulation. Inclure des modules pour promouvoir la qualification à distance. Définir les traitements afin d'optimiser les données pour la réalité augmentée (qualité du maillage pour la vision par ordinateur, etc.). Nous pouvions aussi identifier et définir les traitements afin d'optimiser les données pour la simulation interactive (détection de collision, dynamique, simulation de mécanisme, etc.) dans CATIA.

Il serait intéressant de faire de recherches sur d'autres protocoles, tels que l'AP233 (duquel nous n'avons pas parlé ici). Il semble être utile pour lier les processus de PLM avec SysML Dans le cadre de ce projet, ce standard pourrait devenir très utilisé, non seulement pour échanger les informations existantes, mais pour définir de nouveaux modèles.

Ce mémoire est le reflet d'une démarche de recherche. Il n'aborde pas tous les aspects théoriques nécessaires à une réflexion complète sur le sujet abordé. Son élaboration m'a amené à comprendre que le domaine de recherche est en constant évolution et que les résultats d'une recherche ne sont pas définitives et qu'il faut cependant se fixer une limite pour finir et pour présenter les résultats.

Bibliographie

[CPS+10] J. Cormier, D. Pasco, C. Syllebranque, C. De Keukelaere, and P. Chevaillier. Virteasy a haptic simulator for implantology surgical training designed by an activity analysis. *Virtual Reality International Conference*, 2010.

[CTB+11] P. Chevaillier, T. Trinh, M. Barange, P. De Loor, F. Devillers, J. Soler, and R. Querrec. Semantic modelling of virtual environments using mascaret. Février 2011.

[CUB12] PLCS CUBE. What is plcs ? http ://www.plcs-community.com/en/reminderoverview-what-is-plcs, Consulté le 13 février 2012.

[fA12] Beremiz Open Source Software for Automation. About beremiz. http ://www.beremiz.org/, Consulté le 29 février 2012.

[feiaf12] PLCopen for efficiency in automation for efficiency in automation. Plcopen. http ://www.plcopen.org/, Consulté le 25 février 2012.

[fES12] GREENSOFT Software Tools for Embedded Systems. Controlbuild présentation. http ://www.geensoft.com/fr/article/controlbuild, Consulté le 01 mars 2012.

[FM06] P. Fuchs and G. Moreau. *Le traité de la réalité virtuelle*, volume Volume 2 : L'interfaçage, l'immersion et l'interaction en environnement virtuel. Les Presses de l'École des Mines. Paris, deuxième édition edition, 2006.

[Fur08] B. Furht. *Encyclopedia of Multimedia*. Springer, 2nd edition edition, 2008.

[GPV] W. Goralski, M. Poli, and P. Vogel. *VRML : Exploring virtual worlds on the INternet*. Prentice Hall.

[ISO] ISO. Industrial automation systems and integration—product data representation and exchange—part 11 : Description methods : The express language reference manual.

[KH06] D. Karagiannis and B. Höfferer. *Metamodeling as an integration concept*, volume IData Technologies : First International Conference, ICSOFT 2006. CCIS 10, p. 37-50. Springer- Verlag Berling Heidelberg, 2006.

[MQC] N. Marion, R. Querrec, and P. Chevaillier. Integrating knowledge from virtual reality environments to learning scenario models : A meta-modeling approach. *LISyC. ENIB. UEB.*

[Que02] R. Querrec. Les systèmes multi-agents pour les environnements virtuels de formation. application à la sécurité civile. octobre 2002.

[Roq09] P. Roques. Sysml par l'exemple. un langage de modélisation pour systèmes complexes. 2009.

[RVLW04] S. Russel, P. Vijay, J. Lubell, and S. Waterbury. Step, xml and uml : Complementary technologies. In *ASME 2004 Design Engineering Technical Conferences and Computer Information in Engineering Conference*, 2004.

[Sta11] J. Stark. *PLM Challenges. 21st Century Paradigm for Product Realisation.* Springer, 2nd edition edition, 2011.

[Sup12] Product Life Cycle Support. Data exchange specifications (dex). http ://www.plcs-resources.org/, Consulté le 13 février 2012.

[Sys12] Dassault Systems. À propos de catia. http ://www.3ds.com/fr/products/catia/, Consulté le 31 mars 2012.

[TBB+04] P. Tchounikine, M. Baker, N. Balacheff, M. Baron, A. Derycke, M. Guin, J. F. Nicaud, and P. Rabardel. Palton-1 : quelques dimensions pour l'analyse des travaux de recherche en conception d'eiah. rapport technique, rapport de l'action spécifique fondement théorique méthodologique de la conception d'eiah. Technical report, Département STIC du CNRS, 2004.

[Tra08] Nexter Training. Synthèse générale. fiche technique/projet siforas. v5.0. Technical report, 2008.