

UNIVERSITE DE LIMOGES

ECOLE DOCTORALE Science - Technologie - Santé

FACULTE des Sciences et Techniques

Laboratoire XLIM

Thèse N°

Thèse

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE LIMOGES

Discipline : Informatique

présentée et soutenue publiquement par

Jean DRAGONAS

le 26 Juin 2006

Modélisation déclarative collaborative

**Systemes collaboratifs pour la modélisation déclarative en synthèse
d'image**

Thèse dirigée par le Professeur Dimitri PLÉMÉNOS

Coencadrement Professeur Georges MIAOULIS

JURY :

rapporteurs

Mme. Maître de conférences-HDR Véronique GAILDRAT, Université Paul Sabatier
Toulouse III

M. Professeur Associé Nikolaos VASSILAS, Institut d'Education Technologique d'Athènes
examineurs

M. Professeur Georges MIAOULIS, Institut d'Education Technologique d'Athènes

M. Professeur Djamchid GHAZANFARPOUR, Université de Limoges

M. Professeur Dimitri PLÉMÉNOS, Université de Limoges

Remerciements

Je souhaite remercier Madame Véronique Gaildrat, Maitre de Conférences HDR à l'Université Paul Sabatier, pour avoir bien voulu rapporter sur cette thèse, et pour ses commentaires qui m'ont permis d'améliorer ce travail.

Je remercie Monsieur Nikolaos Vassilas, Professeur Associé d'Informatique à l'Institut d'Education Technologique d'Athènes, pour avoir bien voulu rapporter sur cette thèse, et pour ses commentaires qui m'ont permis d'améliorer ce travail.

Je remercie Monsieur Georges Miaoulis, Professeur d'Informatique à l'Institut d'Education Technologique d'Athènes, pour avoir accepté de faire partie à mon jury. Je tiens à le remercier également pour avoir guidé mes études depuis de nombreuses années.

Je remercie Monsieur Djamchid Ghazanfarpour, Professeur d'Informatique à l'Université de Limoges et directeur du Laboratoire MSI, pour avoir accepté de faire partie à mon jury.

Je tiens à remercier Monsieur Dimitri Pléménos, Professeur d'Informatique à l'Université de Limoges, pour avoir guidé mes études depuis de nombreuses années, et pour son soutien aussi précieux tout au long de ces travaux.

Je remercie aussi Dimitrios Makris, docteur de l'Université de Limoges, Vasilis Golfopoulos, George Bardis, et Ioanna Ravani pour leur collaboration. Je remercie également Apostolos Lazaridis pour sa contribution à l'implémentation de ce travail.

Résumé

Ce mémoire contribue au développement de systèmes collaboratifs de modélisation déclarative par décomposition hiérarchique en synthèse d'images.

Nous avons étudié dans un premier temps les systèmes actuels qui existent dans le domaine de la conception collaborative basée sur Internet et sur les technologies Web. Ensuite, nous avons développé un modèle d'utilisateur et de processus de modélisation collaborative déclarative. Puis un mécanisme de collaboration pendant la phase de description déclarative de scènes a été développé, basé sur la gestion de l'arbre de décomposition de la scène et du contenu sémantique du modèle déclaratif. Une méthode et des outils appropriés de gestion de projets en Modélisation Déclarative Collaborative ont été développés, focalisés sur le support du processus d'administration des flux de travail.

Mots clés : modélisation déclarative, conception collaborative, administration des flux de travail, conception distribuée, synthèse d'image.

Abstract

“Collaborative Declarative Modelling”

“Collaborative systems for declarative modelling in image synthesis”

This thesis is concerned with the contribution to the development of collaborative systems for declarative modelling by hierarchical decomposition in image synthesis.

Initially, we have studied the existing systems in the field of collaborative design based on Internet and Web technologies. Next, we have developed user models and collaborative processes for declarative modelling. A mechanism for collaboration during the phase of declarative description of scenes has been developed, based on the management of the decomposition tree of the scene and the semantic contents of the declarative model. Finally, a method and the appropriate tools for project management in Collaborative Declarative Modelling have been developed, focused on the administration support of the process workflow.

Keywords: Declarative modelling, collaborative design, project workflow management, distributed design, image synthesis.

Table de matières

Chapitre 1 Introduction	1
1.1. Introduction.....	1
1.2. L'automatisation de la conception et la CAO	4
1.3. Systèmes de CAO déclaratifs.....	4
1.4. Les objectifs de ce travail	5
1.5. L'organisation de la thèse	7
Chapitre 2 Etat de l'art.....	11
2.1. Introduction.....	11
2.2. Le processus de conception	11
2.3. Systèmes de conception dans le domaine de la CAO	14
2.4. Conception collaborative	15
2.4.1. Architecture des systèmes de collaboration.....	17
2.4.1.1. La CAO collaborative horizontale	17
2.4.1.1.1. Les systèmes de conception basés sur la visualisation.	18
2.4.1.1.2. Les systèmes de co-conception	18
2.4.1.1.3. Les systèmes de collaboration hiérarchique	23
2.4.2. Modeleurs de conception	24
2.4.2.1. Modeleurs Géométriques	24
2.4.2.2. Les modeleurs basés sur les caractéristiques	25
2.4.3. Systèmes de collaboration.....	26
2.4.3.1. CSCW-FeatureM	28
2.4.3.2. COLLIDE	28
2.4.3.3. TOBACO	29
2.4.3.4. ARCADE	30
2.4.3.5. Collaborative Solid Modelling.....	31
2.4.3.6. NetFeature.....	32
2.4.3.7. Web Based Collaborative CAAD	33
2.4.3.8. WebSpiff.....	34
2.4.3.9. Syco3D.....	34
2.4.3.10. CyberCAD	35
2.4.3.11. Systèmes commerciaux de CAO collaboratifs	35

2.4.3.12. Conclusion	37
2.5. La modélisation déclarative	39
2.5.1. Le fonctionnement général d'un modèleur déclaratif.....	41
2.5.1.1. La modélisation déclarative par décomposition hiérarchique	45
2.5.1.2. Décomposition hiérarchique dans MultiFormes.....	46
2.5.2. MULTICAD	48
2.5.2.1. L'architecture MultiCAD.....	48
2.5.2.1.1. Phase de Description.....	49
2.5.2.1.2. Phase de génération.....	50
2.5.2.1.3. Phase de prise de connaissances	51
2.5.2.1.4. La gestion des informations et des connaissances	51
2.5.2.2. Le cadre DKABM (DKABM Framework).....	54
2.5.2.2.1. Analyse du cadre.....	55
2.6. Discussion.....	58
Chapter 3 Research Proposal	61
3.1. Introduction.....	61
3.2. Kernel – CDMS framework.....	62
3.2.1. Framework analysis	63
3.2.2. Collaborative declarative process	64
MultiCAD designers	64
3.2.3. Collaborative Declarative Module (CDM)	66
3.2.4. Server facilities	75
3.2.5. Collaborative Base.....	75
3.2.6. Flowchart	76
3.3. Enhanced-CDMS	78
3.3.1. Framework analysis	79
3.3.1.1. KAU Entities' General Description.....	80
3.3.1.2. WAU Entities' General Description.....	82
3.3.1.3. Flowchart	84
Chapter 4 Prototype system implementation	87
4.1. Introduction.....	87
4.2. Kernel-CDMS application software	87
4.2.1. Operating System.....	88
4.2.2. Network platform – Hardware requirements	88

4.2.3. Kernel-CDMS implementation framework.....	89
4.2.4. Kernel-CDMS modules	91
4.2.4.1. Collaborative Declarative module	91
4.2.4.2. Autorisation Layer	92
4.2.4.3. Collaborative Mechanism Layer.....	94
4.2.4.4. Collaborative Base layer	100
4.2.4.5. Collaborative Base module	101
4.2.5. MultiCAD Designer module.....	102
4.3. Enhanced-CDMS system implementation.....	105
4.3.1. Enhanced CDMS modules.....	106
4.3.1.1. Kernel Administration Unit (KAU).....	108
4.3.1.2. Workflow Administration Unit (WAU).....	110
4.3.2. Collaborative Event Record Mechanism	111
Chapter 5 Case Study – System Evaluation.....	113
5.1. Introduction.....	113
5.2. Case study of collaborative activity	113
5.3. Case study for Enhanced-CDMS	127
5.3.1. KAU case study	127
5.3.2. WAU case study	132
5.4. Case study conclusions	134
5.4.1. Kernel – CDMS	135
5.4.2. Enhanced-CDMS	136
Chapitre 6 Discussion-Conclusion.....	139
6.1. Discussion	139
6.2. Conclusion	142
6.3. Travaux futurs – Perspectives.....	145
Index of Abbreviation	147
Bibliography	149

Liste de figures

Figure 2-1 Les phases d'un cycle de conception	13
Figure 2-2 Classification des outils et des 'frameworks' de conception	15
Figure 2-3 L'architecture de CSCW - FeatureM	28
Figure 2-4 L'architecture de <i>CollIDE</i>	29
Figure 2-5 L'architecture de <i>TOBACO</i>	30
Figure 2-6 L'architecture de <i>ARCADE</i>	31
Figure 2-7 L'architecture de <i>CSM</i>	32
Figure 2-8 L'architecture de <i>NetFeature</i>	33
Figure 2-9 L'architecture de <i>WebCAD3D</i>	34
Figure 2-10 L'architecture de <i>CyberCAD</i>	35
Figure 2-11 Comparaison entre modelleur conventionnel et modelleur déclarative	40
Figure 2-12 Modélisation détaillée du processus de conception déclarative	43
Figure 2-13 La décomposition hiérarchique de la description d'une scène	47
Figure 2-14 L'architecture de MultiCAD projetée auprès du cycle déclaratif	49
Figure 2-15 Le méta-modèle de la scène et la connaissance des descriptions dans MultiCAD	53
Figure 2-16 L'architecture de MultiCAD vue d'ensemble	54
Figure 2-17 DKABM, ensemble d'entités avec leurs relations	56
Figure 2-18 Description de bâtiment <i>MY_House</i> basé sur le cadre DKABM	57
Figure 3-1: Kernel – CDMS scheme	63
Figure 3-2 Kernel – CDMS framework	64
Figure 3-3 Collaborative process in the basic declarative conception cycle	65
Figure 3-4 Project decomposition	65
Figure 3-5 <i>Collaborative Declarative module</i>	66
Figure 3-6 Collaborative Description Phases	67
Figure 3-7: The basic cycle of Assignment sub-phase	67
Figure 3-8 Scene graph	68
Figure 3-9: Global and Local constraints	69
Figure 3-10: A typical situation of a tree with one registered node	71
Figure 3-11: Different states of an object	72
Figure 3-12 Relations for a registered object	74
Figure 3-13 Server facilities	75

Figure 3-14 Collaborative base.....	76
Figure 3-15 Flowchart of Kernel – CDMS.....	77
Figure 3-16 <i>Enhanced-CDMS</i> scheme.....	79
Figure 3-17 <i>Enhanced-CDMS</i> framework.....	80
Figure 3-18 Development Process.....	83
Figure 3-19 Enhanced-CDMS flowchart.....	85
Figure 4-1 Kernel-CDMS system framework.....	90
Figure 4-2 CDMS within the MultiCAD environment.....	91
Figure 4-3 Restriction layers.....	92
Figure 4-4 Authorisation Layer.....	93
Figure 4-5 Collaborative Mechanism Layer.....	95
Figure 4-6: <i>Conceptual Scene graph tree</i> construction algorithm.....	97
Figure 4-7 <i>Node Registration</i> algorithm.....	99
Figure 4-8 Collaborative mechanism layer.....	100
Figure 4-9 Collaborative Base Layer.....	101
Figure 4-10 Collaborative base module.....	102
Figure 4-11 MultiCAD designer’s browser.....	103
Figure 4-12 Graphic visualisation interface.....	104
Figure 4-13: System framework.....	105
Figure 4-14 Enhanced-CDMS within the MultiCAD environment.....	106
Figure 4-15: Collaborative base schema.....	107
Figure 4-16 CAB Algorithm flowchart.....	108
Figure 4-17 CAB Algorithm operation.....	109
Figure 4-18 Different states of an entity.....	110
Figure 4-19 CERM operation.....	111
Figure 5-1: Project hierarchical tree.....	114
Figure 5-2 Designers login.....	116
Figure 5-3 Designers registration.....	116
Figure 5-4 Object addition.....	117
Figure 5-5 Object addition.....	117
Figure 5-6 Collaborative tree.....	117
Figure 5-7 Object addition.....	118
Figure 5-8 System information message.....	118
Figure 5-9 Object addition.....	119

Figure 5-10 Collaborative tree	119
Figure 5-11 Collaborative tree with roof	120
Figure 5-12 Designer registration	120
Figure 5-13 Object addition	121
Figure 5-14 Collaborative with Building2	122
Figure 5-15 Relation addition	122
Figure 5-16 Relation addition	122
Figure 5-17 Relation addition	122
Figure 5-18 Relation addition	122
Figure 5-19 Relation addition	123
Figure 5-20 Relation addition	123
Figure 5-21 Relation addition	123
Figure 5-22 Relation addition	123
Figure 5-23 Relation addition	123
Figure 5-24 Properties' values	123
Figure 5-25 Properties' values	124
Figure 5-26 Relation addition	124
Figure 5-27 Relation addition	124
Figure 5-28 Relation addition	124
Figure 5-29 "Santorini House" geometric model (spaces and roofs)	125
Figure 5-30 "Santorini House" geometric model (spaces)	126
Figure 5-31 Designer logout	126
Figure 5-32 User1 creation	127
Figure 5-33 Existing User1 load in	128
Figure 5-34 Request rejection	128
Figure 5-35 User2 creation	129
Figure 5-36 User1 edit	129
Figure 5-37 Team Leader definition	130
Figure 5-38 Team Member definition	130
Figure 5-39 KAU example	131
Figure 5-40 WAU case study	132
Figure 5-41 Set of due date	133
Figure 5-42 Project working period	133
Figure 5-43 Proposals collection	134

Liste de tables

Tableau 2-1 Les infrastructures des systèmes de CAO repartis [Fuha et al 05]	19
Tableau 2-2 Projets de recherche en conception collaborative.....	37
Tableau 2-3 Légende du Tableau 2-2	38
Table 3-1 Relationship between the state of a node and the Modify operation.....	70
Table 3-2 Relationship between the state of a node and the Modify-Insert-Delete operation (for another user)	73
Table 3-3 Relationship between the state of a node and the Modify-Insert-Delete operation (for same user)	73
Table 4-1 <i>Designer</i> information table.....	93
Table 4-2 <i>Login</i> table	94
Table 4-3 <i>DKABM</i> table.....	95
Table 4-4 <i>Actual</i> objects	96
Table 4-5 <i>Analysis</i> table.....	98
Table 4-6 <i>Closed_Objects</i> table	99
Table 5-1 Building 1 spatial relations	114
Table 5-2 Building 2 spatial relations	114
Table 5-3 Objects placement and dimensions	115
Table 5-4 Objects placement and dimensions	115
Tableau 6-1 Comparaison qualitative de projets collaboratifs	140
Tableau 6-2 Légende de Tableau 6-1.....	141

Chapitre 1

Introduction

1.1. Introduction

Au cours des dernières années les spécialistes de la science des ordinateurs ont fixé leur attention sur les systèmes dits « de collaboration » et sur ce qui peut contribuer à une coopération plus rentable entre travailleurs, étudiants, dessinateurs et autres groupes. Grâce aux dernières évolutions de la technologie, comme Internet par exemple, la complexité du support à la collaboration a diminué, tandis qu'augmentaient son efficacité et sa demande.

La notion de collaboration peut être définie comme le développement de relations d'entraide entre des individus ou des groupes qui ont des objectifs communs. Le besoin de la collaboration se fait sentir au moment où les limites de leurs capacités empêchent les individus de réaliser un travail de façon isolée (par manque de connaissance ou de force), ou quand la collaboration peut les aider à réaliser ce travail plus rapidement et plus efficacement. On a remarqué, expérimentalement, que n'importe quel travail peut être mené à bien plus vite et avec de meilleurs résultats dans un environnement collaboratif ou de réseau, du fait de l'exécution parallèle des tâches par les membres du groupe. On constate de plus un accès aux informations plus rapide à partir d'une base de données commune, une amélioration de la volonté de coopérer des participants, une diminution des coûts et plus généralement une rentabilité accrue, dans la mesure où les échanges et la communication se font désormais électroniquement.

Si on demandait à plusieurs personnes différentes du même milieu de l'informatique, ce que sont *les systèmes de collaboration*, on recevrait très probablement plusieurs réponses différentes. Certains diraient que la collaboration ou la communication, c'est l'e-mail. D'autres mentionneraient les téléconférences et le World Wide Web (WWW). Il est même probable qu'on entendrait citer les moyens

modernes de discussion et le chat. En règle générale, il est difficile de définir cette notion qui est large, en raison du grand nombre de technologies disponibles aujourd'hui et, en réalité, toutes les réponses citées ci-dessus sont correctes.

Les systèmes de collaboration résultent en partie de l'utilisation de différentes technologies dans le même environnement, ce qui a pour but de faciliter et de promouvoir la distribution des informations, la gestion du travail et de la communication. Néanmoins, cette « unification » de la technologie ne représente qu'une part de l'ensemble de la *collaboration* telle que nous la définissons. Une autre part de celui-ci est la synchronisation. Nous connaissons tous les applications où l'on travaille simultanément avec d'autres et où chacun à son tour travaille et apporte des idées. Or les nouvelles technologies nous offrent un autre mode de *collaboration* – la collaboration non synchronisée où il n'est pas nécessaire que les participants se trouvent ensemble, et qui nous permet de collaborer quand on le veut. L'e-mail, internet, les intranets, les bases de données partagées sont des exemples de ces types de systèmes de collaboration.

Dans leur publication, Gero et Mc Neill [Gero et al 98] ont montré que la conception peut être vue comme une série d'activités distinctes, que le niveau de la spécialisation est déterminé par la façon dont sera réalisé le travail et que ce niveau est montré par le temps de réalisation. Quand les collaborateurs se réunissent pour la conception, on peut supposer que la nature de leur travail ne change pas dans la mesure où la collaboration exige d'un concepteur qu'il collabore, mais aussi qu'il contribue à certaines tâches de façon isolée. La collaboration est, selon toute vraisemblance, circulaire et répétée. Cela signifie que la conception demeure constituée par une série de démarches distinctes. Les collaborateurs travaillent ensemble au début, mais ensuite ils se séparent et travaillent de leur côté. Les participants agissent comme des spécialistes distincts qui parlent toujours en voyant les choses de leur optique. Leur spécialisation peut prendre différentes formes au cours de cette procédure du fait que leur compréhension est complémentaire et que c'est en participant à cette procédure qu'ils apprennent.

Cette analyse s'accorde à un modèle de « conception collaborative » qui serait constituée d'actions parallèles de spécialistes, chacune étant de courte durée, elles-mêmes encadrées par une action concertée de discussion et d'évaluation. Ainsi

l'action elle-même de concevoir est distincte, isolée, parallèle et finalement n'est pas complètement reliée aux autres. Dans ce modèle, on voit que le travail est coopératif de par sa nature et que le caractère coopératif découle de l'appréciation et de l'évaluation, activités existantes et indispensables dans la « conception collaborative » réalisée avec le soutien de l'ordinateur. Ce point de vue sur la conception est soutenu également dans le classement par catégorie de la « conception collaborative » présenté dans l'expérience de Maher et de son équipe [Maher et al 98]. De façon plus détaillée on distingue :

- La collaboration commune où les participants sont occupés à travailler ensemble.
- La collaboration exclusive où les participants travaillent sur différentes parties du problème en discutant régulièrement et en se consultant les uns les autres.
- La collaboration « dictatoriale » où les participants décident qui sera le leader qui va diriger la procédure.

De fait, c'est dans la collaboration exclusive que Maher et son équipe ont remarqué les résultats les plus productifs [Maher et al 98]. La collaboration commune, quant à elle n'a abouti à aucun résultat après un échange très intense des participants. Enfin, la collaboration « dictatoriale » a abouti à une solution quand le leader a pris la décision finale.

Ainsi donc, la conception exclusive, par rapport à une conception réalisée en simple collaboration, demande un sens de la coopération plus élevé pour atteindre un résultat plus créatif. Il s'agit d'une fonction beaucoup plus exigeante, plus difficile à mettre en route et à continuer que le simple accomplissement d'un but par un groupe. Il faut signaler que la collaboration ne comprend pas de réexamen de la décision par les membres du groupe, séparément ou après entente. Cependant si on regarde un peu plus loin dans la collaboration, un autre problème apparaît : celui des concessions faites par les participants sur leurs exigences. Quand cela se produit, cela signifie l'existence d'un système peu organisé qui ne satisfait qu'en partie les participants. Il ne touche pas en profondeur les problèmes existants, mais propose une ébauche rapide de solution. On voit que les concepteurs font très souvent ce type de capitulation. Chez Cross [Cross et al 95] par exemple, les concepteurs qui collaborent arrivent à des conclusions qui sont certes suffisantes mais ne constituent pas la solution

optimale. Ce qui ne signifie pas que ce type de solutions doive être rejeté, puisque celles-ci sont satisfaisantes et souvent originales.

1.2. L'automatisation de la conception et la CAO

L'informatique et son parcours de développement énorme pendant les dernières années en combinaison avec les recherches scientifiques de modélisation des produits fournissent au concepteur des outils et des logiciels informatiques diverses: des programmes d'analyse et de modélisation d'un produit, des interfaces graphiques, des systèmes de gestion des données etc.

La Conception Assistée par Ordinateur (CAO) est tout un domaine qui s'occupe de rechercher des méthodes, techniques et outils pour aider le concepteur à s'exprimer, à vérifier ses idées, à présenter ses esquisses, à les mettre à jour, même à ajouter ou extraire de l'information dépendant du produit final avant de procéder à la phase de production. Il existe actuellement des logiciels de CAO très puissants et la plupart d'eux sont commercialisés. Le fait est que tandis qu'ils offrent une gamme d'outils au concepteur, ces outils concernent plus la gestion des interfaces du logiciel que à l'aide effective à la conception elle-même. Ceci est dû principalement au fait que ce genre des logiciels est assez lourd et pour un simple dessin le concepteur doit savoir gérer beaucoup de détails avant de le créer.

La création d'un produit nécessite une description technique complète pour pouvoir se réaliser. De l'autre côté, pour que nous caractérisions un système comme étant un système de CAO, il doit comporter des fonctionnalités d'aide efficace à la conception.

1.3. Systèmes de CAO déclaratifs

Les besoins du concepteur pendant la phase de la conception peuvent être satisfaits par des outils de la CAO classique qui incorporent des techniques et interfaces de modélisation des produits, enrichis par des mécanismes de gestion intelligente de la connaissance (antérieure ou non) de l'utilisateur, et enveloppés dans des environnements abstraits qui permettent des descriptions non concrètes.

Cette spécification des outils coïncide avec les caractéristiques des systèmes de CAO déclaratifs. Ce sont des systèmes de CAO qui sont basés sur l'approche de la modélisation déclarative [Plemenos 91], [Lucas et al 95], [Desmontils 95]. En fait, l'avantage primordial de la modélisation déclarative est de pouvoir décrire un produit par des propriétés, éventuellement imprécises ou incomplètes. Ce type de description est plus proche de la perception d'une scène par le concepteur : l'utilisateur est supposé définir une scène à partir d'idées vagues. En modélisation déclarative, dans la phase de conception, l'utilisateur dispose d'un outil qui lui permet d'exprimer sa conception à travers des descriptions plus générales souvent textuelles.

En plus, il existe des modeleurs qui utilisent la notion de modélisation déclarative par décomposition hiérarchique [Plemenos 95] c'est-à-dire, ils décomposent la description d'un produit en niveaux d'abstraction, permettent une description structurée de manière hiérarchique qui définit que le produit est découpé et décrit par ses composants, chacun entre eux pouvant être lui-même décomposé de la même façon. En incorporant ce genre de modeleurs, les dernières recherches se focalisent sur l'incorporation de la gestion des connaissances et des informations. MultiCAD [Miaoullis et al 96], [Miaoulis 02] est un tel cadre de systèmes d'information «intelligents» dédiés à la CAO. En particulier, MultiCAD propose toute une architecture pour traiter le développement de modèles de représentations et l'organisation des informations, des connaissances, des représentations et du savoir faire nécessaire aux concepteurs qui travaillent dans le cadre d'un processus de conception déclarative.

1.4. Les objectifs de ce travail

L'objectif principal de cette thèse est de proposer et implémenter un système collaboratif de modélisation déclarative par décomposition hiérarchique en synthèse d'image. C'est dans ce but qu'ont été développés deux mécanismes. Le premier est un mécanisme collaboratif durant la phase de la description déclarative de la scène, qui est basé sur la co-conception et la co-gestion de la description déclarative (de l'arbre de décomposition de la scène et de son contenu sémantique). Le deuxième mécanisme concerne la gestion de projets déclaratifs collaboratifs et se focalise sur le support

d'administration des flux de travail (workflow) pendant toute la durée de la conception du cycle déclaratif.

Ce système met à la disposition de plusieurs concepteurs parfois très éloignés les uns des autres, des outils permettant de concevoir ensemble une scène en apportant chacun ses compétences et son savoir faire, sans que le travail de l'un remette en cause le travail de l'autre. La conception architecturale des bâtiments est en soi multidisciplinaire : pas une seule personne n'est capable de produire un processus complet de développement pour un dispositif ou un système de conception de bâtiment. Au sein de la communauté des chercheurs est apparu récemment une orientation appelée système "Collaborative Product Development" (CDP), qui est défini comme étant : «une architecture informatique basée sur Internet qui permet le partage et le transfert de la connaissance et de l'information sur le cycle de vie du produit pour des compagnies géographiquement dispersées afin de les aider à prendre les bonnes décisions technologiques dans un environnement de collaboration» [Rodriguez et al. 02]. Durant ces dernières années de nombreuses technologies existaient déjà pour aider au développement de produits en collaboration et elles se focalisaient sur le partage des informations de production et sur l'approvisionnement en outils de collaboration pour maintenir ensemble une équipe multidisciplinaire.

Pour l'accomplissement de ce projet nous développons une plate-forme de conception collaborative déclarative, basée sur le WWW pour la conception en collaboration de bâtiments durant les premières phases de leur conception. Cette plate-forme est constituée de deux parties reliées entre elles. La première concerne le Kernel-CDMS, et la seconde le Enhanced-CDMS. En particulier, le '*Kernel-Collaborative Declarative Modelling System*' (Kernel-CDMS) propose ici aux utilisateurs/concepteurs de différents pays de participer au même processus de conception dans un système de conception déclaratif (MultiCAD). Le système a été conçu et implémenté pour fournir aux architectes- concepteurs un environnement de travail pour le développement de grand(s) projet(s) architecturau(x) de manière synchrone ou asynchrone et entre des lieux de travail géographiquement distants. La deuxième, le '*Enhanced-Collaborative Declarative Modelling System*' (Enhanced-CDMS) pourrait apporter des bénéfices supplémentaires résultant d'une nouvelle structure et de nouvelles fonctions. En particulier, ce système fait face à deux

demandes : premièrement, la gestion du rôle des concepteurs (Définir les hiérarchies et groupes de travail parmi les concepteurs. Introduire le concept de « rôle » dans un environnement de collaboration.). Deuxièmement, la gestion des projets (définir les sous-parties spécifiques d'un projet – définir les parts de travail des groupes de concepteurs). Le Enhanced-CDMS offre une série de facilités de collaboration pour la conception et la gestion workflow du projet. Un projet architectural peut être conçu avec l'aide d'un système spécifique qui est totalement basé sur l'architecture logicielle du système MultiCAD.

L'environnement spécifique de collaboration dans lequel les projets architecturaux sont implémentés fournit la possibilité à n'importe quel architecte-concepteur de travailler selon son agenda personnel et de participer à des projets dans différents lieux géographiquement distants de lui/elle. Il offre aussi à l'architecte-concepteur la possibilité de commenter le travail de conception de ses collègues, d'adopter de nouvelles méthodes de conception, ou encore de faire face à l'apparition des problèmes avec toute une équipe. Dans le travail présent ici les deux parties de la plate-forme CDMS (le Kernel-CDMS et le Enhanced-CDMS). les deux parties ont analysé et plus particulièrement le mécanisme de collaboration et la conception déclarative. En outre, un étude de cas est présentée où la manière d'opérer et les fonctions du système deviennent évidentes, de même que l'implémentation du mécanisme de collaboration et les bénéfices qui apparaissent lors de l'utilisation de cet environnement de conception en collaboration.

1.5. L'organisation de la thèse

Cette thèse présente l'analyse, le développement et l'implémentation d'un système de conception basé sur le WWW pour la conception en collaboration de scènes durant les premières phases de conception.

Le deuxième chapitre suivant l'introduction présente une revue de la littérature relative aux domaines de recherche fondamentaux de cette thèse. Ces domaines sont au nombre de trois: le processus de conception, le processus de conception collaborative, la modélisation déclarative.

Dans le troisième chapitre nous expliquerons, les approches et les méthodes choisies pour réaliser notre système de “Enhanced Conception Déclarative Collaborative” appelé Enhanced-CDMS.

Dans le quatrième chapitre nous décrivons la structure de l’architecture logicielle de Enhanced-CDMS et nous finirons ce chapitre par une évaluation de notre architecture logicielle tant au niveau des résultats attendus et obtenus qu'au niveau de la validation de son efficacité, de l’assistance fondamentale à la conception.

Dans le cinquième chapitre on présente un scénario expérimental du système Enhanced-CDMS.

Dans le sixième chapitre nous concluons cette thèse en insistant sur les fonctionnalités et les avantages du système proposé et nous présenterons des perspectives de recherches futures.

Chapitre 2

Etat de l'art

2.1. Introduction

Dans ce chapitre, nous allons décrire les principaux domaines de recherche de cette étude. Ce travail définit une zone de recherche qui intègre les techniques Collaboratives avec la Modélisation Déclarative (MD) dans le domaine de la Conception Assistée par Ordinateur (CAO). Ce chapitre passe en revue et évalue la recherche importante accomplie dans ces domaines. Dans la première section nous allons présenter en général le domaine de la conception, et particulièrement les phases du processus de conception.

Dans la deuxième section nous allons présenter la théorie de la conception collaborative. Plus spécialement nous allons centrer notre intérêt sur les environnements collaboratifs basés sur Internet. On soulignera l'impact de l'architecture client /serveur pour le développement d'un système collaboratif pendant la conception. Plus loin on présentera un nombre d'applications récentes des systèmes de conception collaborative.

Dans la troisième section nous allons faire une présentation détaillée d'un exemple de modélisation déclarative. La structure générale des modeleurs déclaratifs sera présentée. Nous focaliserons notre étude sur le fait de savoir comment on a fait face aux limites existant dans les applications récentes des modeleurs déclaratifs. Cependant, on soulignera le manque de modeleurs déclaratifs collaboratifs. Puis, nous présenterons un cadre de modélisation déclarative approprié à notre méthodologie de recherche. Ce cadre sera MultiCAD-II et sera adapté à nos exigences.

2.2. Le processus de conception

Au début de la conception, le concepteur a une idée vague d'une solution potentielle qui couvre les besoins du produit en conception mais cette idée ne

constitue pas le produit final. Pendant les différentes étapes de la conception, le concepteur fait des prédictions sur la performance des solutions intermédiaires afin de retenir celles qui correspondent le mieux à sa description de départ. Donc, la conception implique des efforts conscients afin d'arriver à un statut de solution désiré où la plupart des caractéristiques du produit final sont évidentes.

De plus, la conception semble également être mal structurée dans le sens où il n'y a aucun processus direct à suivre. Un concepteur peut débiter par certains buts et contraintes et travailler vers la conception des objets qui satisferont ces buts, ou également il peut commencer par la description des objets et vérifier si les propriétés de ces objets s'accordent à celles des spécifications posées au départ. Ceci peut comporter respectivement une approche de résolution descendante (top-down) ou ascendante (bottom-up) ou une combinaison des approches.

La littérature relative à la conception nous fournit plusieurs tentatives d'organisation du processus de conception en phases. Il y a eu plusieurs tentatives d'élaboration de modèles d'organisation du processus de conception [Cross 94].

Miaoulis et Plemenos décrivent la conception comme un processus cyclique, où chaque étape itérative raffine le dessin [Miaoulis et al 96] - [Miaoulis 02]. Deux hypothèses sont faites. La première hypothèse est que la conception, en tant que démarche intellectuelle, est décomposable en plusieurs étapes plus ou moins distinctes. La seconde hypothèse est que le processus de conception évolue du général au spécifique, de l'abstrait au concret. On suppose que, dans la plupart des cas de conception, on part d'une idée très générale qu'on va détailler, étape par étape, et dont on va compléter les différents aspects [Miaoulis 02].

Les étapes ou cycles de conception peuvent aussi être décomposés en plusieurs phases. La Figure 2-1 montre comment un cycle de conception peut être décomposé.

Un cycle du processus de conception comporte quatre phases :

- L'idée,
- La description,
- La génération des solutions,

- L'évaluation

En général, la conception commence par une idée ou une notion sur l'objet qui est conçu. Cette phase de conception se produit dans le cerveau humain et n'est pas directement assistée par l'ordinateur.

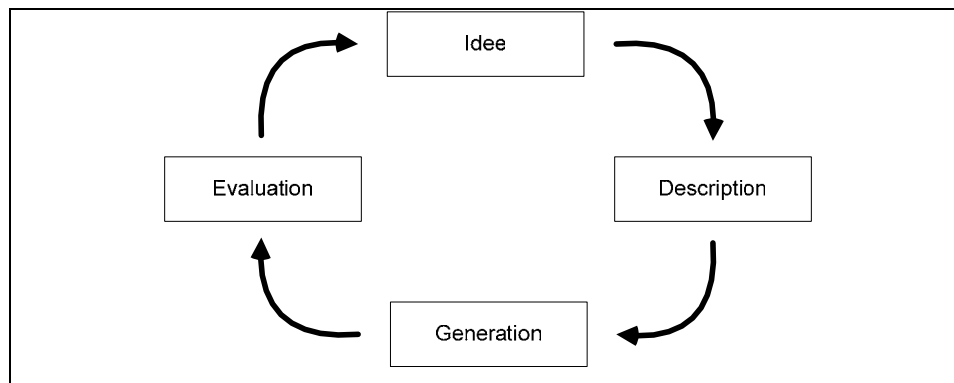


Figure 2-1 Les phases d'un cycle de conception

La seconde phase du cycle de conception est la phase de description. Cette phase se fait à partir d'une idée et la description est faite à l'aide du langage naturel ou d'un langage spécifique. Cette phase peut être assistée par l'ordinateur.

La troisième phase, la génération des solutions est une phase où, à partir d'une description, des réalisations concrètes de tout ou partie du concept de départ sont proposées. Dans cette phase, il est possible d'obtenir plusieurs solutions alternatives qui dérivent du concept de départ. Les solutions obtenues peuvent faire l'objet d'une représentation graphique pour mieux concrétiser le concept. La phase de génération des solutions est typiquement la phase qui peut être prise en charge par l'ordinateur.

La quatrième phase, celle de l'évaluation, permet au concepteur de mesurer assez tôt les conséquences de son idée. A partir de la prise de connaissance des solutions obtenues, le concepteur peut évaluer leur pertinence et faire des choix.

2.3. Systèmes de conception dans le domaine de la CAO

Bien que de nombreux avantages et méthodes de conception effectives soient identifiés, très peu d'outils informatiques existent qui fournissent de l'aide au concepteur humain travaillant dans cette étape du processus de conception. La majorité de ces outils qui existent sont trouvés, la plupart du temps, dans des laboratoires de recherche des universités. Beaucoup de ces outils disponibles ne se focalisent pas sur le processus de conception préliminaire mais concernent le processus entier de la conception. En conséquence, ils ne respectent pas les particularités de chaque produit en conception ou de chaque utilisateur, et fournissent des outils et interfaces d'assistance généraux. Cependant, un concepteur peut être limité par les politiques de conception de sa compagnie (par exemple, son cabinet d'architectes) ou peut être prié d'effectuer les tâches de conception dans un ordre particulier dû aux outils de conception employés. Alors que les politiques de conception de la compagnie devraient être soutenues, en fait imposées, par de bons outils de conception, le concepteur devrait être autorisé à effectuer les tâches de conception dans l'ordre qu'il souhaite. Pour satisfaire ce besoin, dans notre démarche nous prenons en compte l'appui interactif du concepteur pendant la phase de conception. Wang L. et al. [Wang. et al. 02] ont classifié les outils de conception en fonction de leur dépendance ou non du domaine d'application. Ils présentent aussi les cadres (framework) de conception (Figure 2-2) existants dans la littérature. On a ajouté au tableau suivant les deux systèmes commerciaux, CATIATM et AutoCADTM.

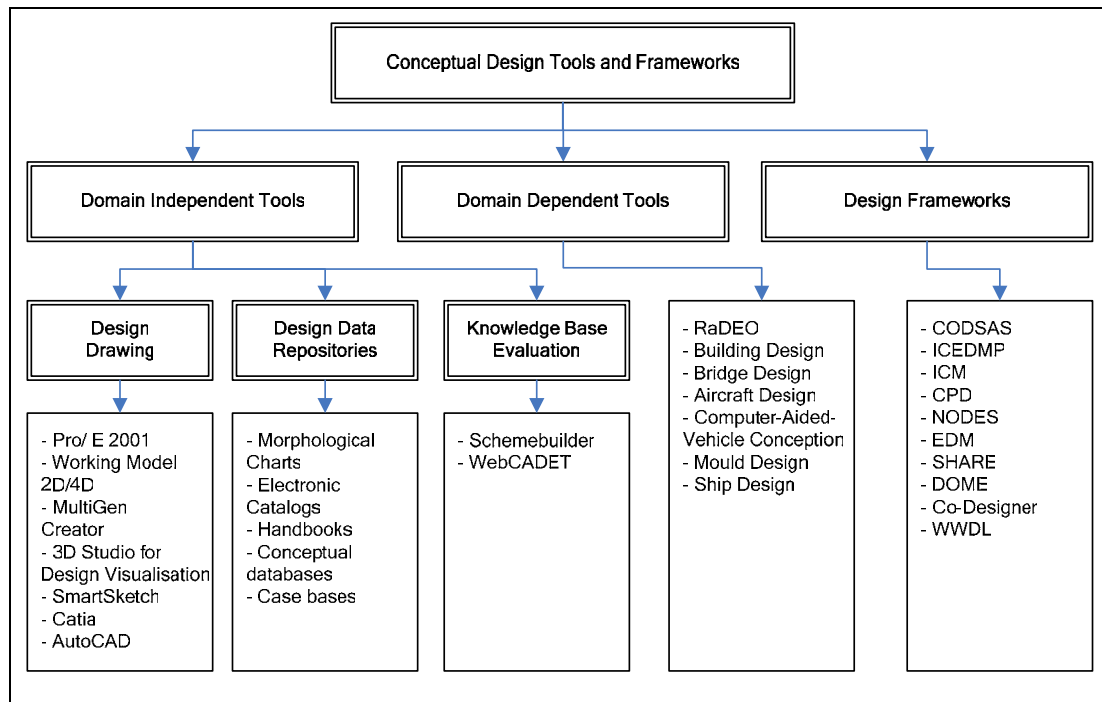


Figure 2-2 Classification des outils et des 'frameworks' de conception

Nous nous concentrerons plus loin sur les systèmes de conception collaborative dans la sous-section qui leur est consacrée.

2.4. Conception collaborative

Au cours des dernières décennies, le monde de l'informatique a détourné son attention des systèmes classiques de prise de décision (par des individus isolés) pour se tourner vers des groupes qui fonctionnent en collaboration. Avec les dernières évolutions technologiques dans des secteurs comme Internet et le World Wide Web (WWW), l'efficacité et la complexité de ces systèmes se sont améliorées considérablement. Mais, parallèlement, les exigences en support du travail à distance et en collaboration se sont multipliées. Bien que les possibilités offertes par la téléconférence diminuent certainement le coût des grands travaux et collaborations dans le domaine de la conception, ce qui est recherché, ce n'est pas seulement d'assurer la communication entre les collaborateurs. D'autres questions importantes se posent :

- Au-delà de la communication entre les membres, comment assurera-t-on la disponibilité des informations du travail? Celles-ci devront être complètement renouvelées et organisées dans une base à laquelle tous les membres auront accès.

- Comment assurer l'intégrité d'une telle base ?

La première question se rattache au prototype et on a beaucoup écrit et réalisé dans ce domaine (ex : modèle STEP (STandard for the Exchange of Product model data) de ISO (International Standards Organization) [ISO 92] [Eastman 99]). La deuxième question concerne la possibilité d'apporter des changements dans la base et le règlement des conflits qui pourraient éventuellement survenir entre les points de vues opposés des spécialistes sur ce sujet.

La conception collaborative est l'application des principes et des modèles des systèmes de collaboration dans le secteur de la conception pour les ingénieurs, les architectes, les concepteurs et de nombreuses autres branches professionnelles auxquelles elle peut être utile. Dans ce domaine donc, on observe des exigences de plus en plus accrues pour un petit nombre de produits de haute qualité et de grande variété. Cela a rendu nécessaire la création de groupes d'ingénieurs et de concepteurs qui devront collaborer depuis des régions géographiquement éloignées, avec l'aide de réseaux et autres services informatiques. Ces groupes « décentralisés » exigent un bon cadre de fonctionnement qui leur permettra de collaborer correctement et d'utiliser complètement les informations concernant la Conception Assistée par Ordinateur (CAO) disponibles.

De nombreux modèles originaux basés sur le réseau et offrant des solutions aux problèmes ci-dessus, se sont développés. Le WWW et le VRML (Virtual Reality Modeling Language) sont des exemples des technologies utilisées. Mais il existe aussi des plateformes de réseau indépendantes. Le Net (WWW), en combinaison avec le navigateur approprié (Internet Explorer – Mozilla Firefox) est un outil très efficace pour le stockage et la récupération des informations.

Il peut aussi stocker des informations qui résultent de la collaboration et des différentes activités des membres collaborateurs. La conception, en constante évolution, est publiée sur le WWW et le groupe y accomplit différentes opérations (perfectionnements et modifications) jusqu'à ce qu'elle acquière sa forme finale. Evidemment de futures équipes de concepteurs peuvent facilement exploiter cette information pour leurs futures conceptions. Le bas coût, l'indépendance de tout protocole, des plateformes et des systèmes d'exploitation, tout cela, combiné avec une possibilité d'extension illimitée, rend un tel système idéal pour les systèmes de

conception ainsi que pour les systèmes de gestion d'informations (Product Data Management Systems – PDM).

2.4.1. Architecture des systèmes de collaboration

Un système collaboratif CAD a besoin de deux sortes d'aptitudes et de facilités : la distribution et la collaboration. En général ces deux termes soulignent les différents aspects d'un système [Li et al. 05]. D'abord matériellement, la distribution sépare les systèmes CAO géographiquement dispersés et les développe pour supporter les activités de conception éloignées. Ensuite au point de vue fonctionnel, la collaboration associe et coordonne les systèmes de conception individuels pour atteindre une cible et un objectif de conception globale.

Dans une activité de conception, les utilisateurs participants remplissent un certain nombre de fonctions et de rôles. Ainsi une collaboration CAO peut être organisée d'une manière soit *horizontale*, soit *hiérarchique* [Li et al. 05]. La collaboration *horizontale* met l'accent sur le partage d'une conception de groupe de la même discipline pour mener à bien un travail de conception complexe de façon synchrone ou asynchrone. La collaboration *hiérarchique* peut établir un canal de communication efficace entre une conception en amont et une fabrication en aval.

2.4.1.1. La CAO collaborative horizontale

Les systèmes de collaboration de CAO horizontale peuvent être divisés en deux types : les systèmes de conception basés sur la visualisation qui offrent aux utilisateurs une manière légère d'aider à la conception collaborative en visualisant, en annotant et en inspectant les modèles de conception dans un environnement Web ou CAO, et les systèmes de co-conception qui supportent des fonctions interactives de co-modélisation et de co-modification pour un travail de groupe.

2.4.1.1.1. Les systèmes de conception basés sur la visualisation.

Les systèmes CAO basés sur la visualisation ont été utilisés pour soutenir la visualisation, l'annotation et l'inspection des modèles de conception afin de fournir une assistance aux activités de conception collaborative.

Les systèmes sont soit branchés généralement sur un navigateur Web, soit reliés à des visionneuses dans certains systèmes CAO. En général ces systèmes de visualisation sont "light-weight", facilement déployés, installés sur une plateforme indépendante, et ils peuvent aider une équipe en ligne à entreprendre une discussion de conception, à revoir un produit, à observer une conception, et à suivre le consommateur afin de rehausser la conception collaborative de nouveaux produits.

2.4.1.1.2. Les systèmes de co-conception

Les systèmes de co-conception peuvent efficacement supporter les fonctions de co-modélisation et de co-modification entre les concepteurs. Une organisation de groupe, une coordination et une négociation efficaces peuvent assurer le succès d'un processus de collaboration et il est important de proposer une architecture efficace basée sur des infrastructures de technologie d'information disponibles, comme : client/serveur, égal à égal et Web service.

Les architectures de système partagé

Les architectures pour les systèmes collaboratifs CAO développés peuvent être classées en trois types :

- « serveur faible + client fort »
- « serveur fort + client faible »
- « égal à égal »

Les caractéristiques majeures, les stratégies d'implémentation et les comparaisons sont résumées dans le Tableau 2-1.

Types of infrastructures	Functions	Characteristics	Diagrams
Thin server + strong client	Clients are equipped with whole CAD systems and some communication facilitators. A server plays as an information agent and exchanger to broadcast CAD files and commands generated by a client to other clients.	Standalone CAD systems can be Conveniently distributed through this mechanism. Due to the heavy-weighted client mechanism, it is hard to be migrated to web applications.	
Strong server + thin client	The data structures in clients are light-weighted and they primarily support visualisation and manipulation functions. The main modelling activities are carried out in a common workspace in the server side	Modules can be rent out as an Application Service Provider (ASP). Data consistency is easily kept since the primary models are created and maintained in the server.	
Peer-to-peer	The services or modules of a system can be shared and manipulated by other systems. For the Inventor collaborative tool, an MS Netmeeting tool is embedded for application sharing	This mechanism enables a convenient manipulation on remote services or applications. Due to the heavy burden of networks, the manipulation efficiency of systems is low	

Tableau 2-1 Les infrastructures des systèmes de CAO repartis [Fuha et al 05]

L'architecture client / serveur

L'architecture client / serveur est très répandue sur Internet. Le Web en est sûrement l'exemple le plus évident : les navigateurs envoient des requêtes aux serveurs Web qui leur fournissent des pages HTML (Hyper Text Markup Language) et autres éléments de présentation en retour. Les serveurs hébergent donc des données, mais ils peuvent aussi exécuter des opérations à l'aide par exemple de scriptes CGI (Common Gateway Interface) ou de servlets Java. Du fait de sa topologie, le modèle client / serveur contraint tous les messages à passer par le serveur. Pour qu'un message passe d'un client à un autre, deux transmissions sont ainsi nécessaires : du premier client au serveur, puis du serveur au second client. Il apparaît alors que si le serveur tombe en panne, la session de travail collaboratif ne peut plus continuer. Le serveur en lui-même présente aussi des contraintes : il faut une machine de plus que le nombre de participants, le côté serveur de l'application doit être développé séparément et il doit être administré. Malgré ces contraintes, l'architecture

client/serveur propose aussi des facilités intéressantes. La gestion du groupe de travail peut être centralisée, ainsi qu'une partie de la gestion des messages. De plus, on peut imaginer que le serveur remplisse d'autres services comme la sauvegarde ou la gestion des différentes versions des données. Nous allons maintenant présenter les deux sous-catégories particulières d'architecture client/serveur.

Dans l'architecture « serveur faible/client fort » les clients sont équipés de fonctions CAO entières et de facilités de communication. Un serveur joue le rôle d'échangeur d'informations pour diffuser des archives CAO ou des ordres créés par un client à d'autres clients pendant le processus de conception collaborative. Parmi les systèmes développés de ce type sont inclus CollabCADTM [CAD Group 06], IX DesignTM [ImpactXoft 05], CyberCAD [Tay et al. 03].

Dans l'architecture serveur fort+client faible, les structures des données sont "light-weight" et elles supportent d'abord des fonctions de visualisation et de manipulation (comme la sélection, la transformation, le changement des propriétés de visualisation des parties exposées) Les principales activités de modélisation sont exécutées dans un espace de travail commun du côté du serveur. Une représentation client faible / serveur fort a été proposée pour relever efficacement la performance du système [Li et al. 04] Les systèmes développés comprennent Alibre DesignTM [Alibre 05], One SpaceTM [CoCreate 01], Van den Berg et a. [Van den Berg et al. 00], Li et al. [Li et al. 04], Bidarra et al. [Bidarra et al. 02a], [Bidarra et al. 02b].

L'architecture d'égal à égal

L'architecture « d'égal à égal » comporte le partage et la manipulation de services ou de modules d'un système par d'autres systèmes.

L'architecture d'égal à égal se passe de serveur central. Dans cette architecture l'idée est que chaque machine est à la fois cliente et serveur. L'architecture d'égal à égal est un type du protocole de réseau dans lequel chaque poste de travail a des possibilités et des responsabilités équivalentes et il diffère des architectures client/serveur dans lesquelles quelques ordinateurs sont consacrés à servir les autres. Les réseaux d'égal à égal sont généralement plus simples et moins chers, mais habituellement ils n'offrent pas la même exécution sous les charges d'information grande.

Les avantages de cette architecture sont intéressants : La communication entre les machines est directe, donc rapide. Si l'une des machines tombe en panne, la session de travail collaboratif peut continuer ; on se passe naturellement d'un serveur et du travail qu'il faut lui consacrer. Mais on ne retrouve pas les facilités de l'architecture client/serveur : il faut mettre en place une gestion distribuée des groupes et des messages, et il faut ouvrir un port de communication sur les machines pour qu'elles soient aussi serveurs.

Comparaison des architectures

On peut utiliser les critères suivants pour comparer ces deux architectures:

- Communication : la transmission des informations est-elle directe ou indirecte ? Si elle est directe (elle ne passe pas par un serveur), elle est donc plus rapide.
- Gestion du groupe : a-t-on des facilités pour gérer le groupe de travail ?
- Tolérance aux pannes: quelle incidence peut avoir une panne sur la session de travail collaboratif ?
- Sécurité: l'architecture nécessite-t-elle une ouverture spécifique pouvant poser des problèmes de sécurité, notamment au niveau d'un éventuel pare feu ?
- Installation: l'ajout de matériel est-il nécessaire ? Un développement spécifique est-il nécessaire pour le serveur ? Y a-t-il des coûts supplémentaires en administration ?
- Gestion des messages: la gestion des messages est-elle centralisée ou distribuée ?

En prenant en considération les caractéristiques des systèmes de CAO, les trois architectures montrent des potentialités dans différents aspects. La mise en œuvre de la première architecture est la plus directe comparée aux deux autres. Equipés de facilités de communication, des systèmes de CAO *standalone* peuvent être commodément redéveloppés comme des clients concepteurs et liés ensemble par le serveur avec un échange d'informations et des fonctionnalités de coordination de collaboration. Cette architecture peut efficacement satisfaire les exigences d'une conception assistée par ordinateur pour des opérations interactives à temps réel, depuis que le calcul (computing) géométrique pour la modélisation et la modification est effectué dans l'espace local des clients. Pendant ce temps, il peut comporter des

systèmes de modélisation hétérogènes chez les clients et un format d'échange d'informations neutre, par exemple XML (eXtensible Markup Language), peut être conçu pour la communication dans l'environnement. Cependant l'adaptabilité de cette architecture n'est pas facilement maintenue. Si un nouvel utilisateur est ajouté dans l'environnement, un paquet entier de systèmes de CAO doit être ajouté et configuré. Pendant ce temps il est difficile de faire émigrer une telle architecture sur une application Web. La seconde architecture devient plus populaire depuis qu'elle apporte un nouveau type de modèle commercial – Application Service Provider (ASP). Avec une telle architecture, les petites et moyennes entreprises ou même des concepteurs individuels avec des connaissances dans un domaine spécifique peuvent louer en ligne des systèmes de CAO high-end, de façon à être capables de pouvoir coopérer et participer au processus de conception avec de grandes sociétés. L'échelle de dimension du système peut être relevée depuis qu'il est commode d'ajouter de nouveaux postes dans le système distribué. La troisième architecture a utilisé le type de calcul d'égal à égal. Les services d'un égal avec la fonction CAO peuvent être manipulés par un autre égal. Cette architecture est très performante pour la communication d'un point à un autre point. Cependant, elle n'est pas convenable pour un groupe d'utilisateurs qui travaillent ensemble. En comparaison avec la première architecture, la difficulté d'exécution des deux dernières est accrue alors que leur échelle de dimension est élevée.

Gestion du groupe et coordination de la conception

Les fonctions de gestion du groupe et de coordination de la conception sont des facteurs cruciaux pour créer un groupe bien organisé capable de conduire un travail de conception collaborative. Un mécanisme de session de travail est efficace pour la gestion du groupe. Chaque session peut être utilisée pour organiser un travail collaboratif et les concepteurs dans la même session peuvent partager l'information de conception de façon dynamique (Alibre Design™) [Li et al. 04]. Dans un système, différents travaux de conception peuvent être exécutés en même temps dans des sessions différentes. Dans une session, les concepteurs peuvent jouer des rôles aussi différents que ceux de *chef de projet*, *membres* ou *supporters*. Un *chef de projet* est responsable de la gestion d'une session et supervise tout le processus de conception. Il est autorisé à programmer le processus et à éviter les impasses provoquées, au cours

de la conception, par les problèmes liés au réseau. Un *membre concepteur* peut exécuter la conception en collaboration et un *supporter* peut apporter des commentaires sur la conception ou les ressources dont le besoin peut apparaître au cours de la conception collaborative. L'envoi de messages est une fonction d'assistance importante pour la conception collaborative (Alibre DesignTM, One SpaceTM, CollabCAD) [Tay et al. 03]. Par des messages textuels, vidéos ou audios, les concepteurs pendant une session peuvent communiquer les uns avec les autres pour échanger des idées.

Plusieurs mécanismes sont utilisés pour la coordination de la conception. Un mécanisme de contrôle "token" (pièce) a été utilisé dans Alibre DesignTM [Li et al.04] pour programmer une activité de conception collaborative. Chaque session a un contrôle "token" et, au cours de celle-ci, seul l'utilisateur qui détient le contrôle token est le concepteur actif qui peut éditer une partie, alors que les autres utilisateurs, pendant la même session, reçoivent seulement les informations à jour et restent des observateurs. L'utilisateur qui exécute la fonction d'édition peut devenir observateur en transférant son contrôle token à un autre utilisateur. L'avantage du mécanisme du contrôle token est qu'il permet d'éviter des conflits pendant un processus de conception simultané. Mais son inconvénient est son efficacité peu élevée. Un autre mécanisme est basé sur le système d'agent. Mori et Cutkosky [Mori et al. 98] ont proposé un système basé sur l'agent pour coordonner la conception basée sur la théorie de l'optimalité de Pareto. Les agents réagissent et peuvent suivre les changements de l'état de la conception et y répondre quand un concepteur change son modèle et crée des conflits. Cependant la communication entre les concepteurs au moyen du système d'agent est simple et limitée, et cette architecture ne convient pas à certaines activités de conception complexes. Dans une situation où les concepteurs travaillent pour différents sous-groupes, Shyamsundar et Gadh [Shyamsundar et al. 02] et Chen et al. [Chen et al. 04] ont défini une nouvelle session unie pour contraindre les groupes de conception assignés à un concepteur individuel à former ensemble un groupe entier développé collaborativement. [Case et al. 96] Pendant ce temps des méthodologies sont recherchées pour détecter et gérer les conflits survenant au cours d'une activité collaborative.

2.4.1.1.3. Les systèmes de collaboration hiérarchique

La collaboration hiérarchique organise une activité de conception de façon asynchrone. Différente du traditionnel « sequential engineering » qui est une approche “throwing-over-wall” cette collaboration est encore centrée sur des communications et interactions bidirectionnelles entre les concepteurs et permet d'éviter les conflits qui peuvent se produire au cours d'une activité de conception collaborative simultanée et une attente non nécessaire quand la conception est organisée dans différentes zones de temps. Entreposer, avoir accès aux modèles CAO et les conserver dans des lieux donnés en toute sécurité et de façon commode est crucial pour faire vivre ce type de fonction collaborative

2.4.2. Modeleurs de conception

En général les systèmes de conception collaborative utilisent deux types de modeleurs pour l'opération de manipulation sur les produits conçus : les modeleurs géométriques et les modeleurs basés sur les caractéristiques (feature-based modellers)

2.4.2.1. Modeleurs Géométriques

L'apparition de modeleurs solides (Boundary Representation (BRep)– Constructive Solid Geometry (CSG)) a permis de définir sans ambiguïté les modèles géométriques des parties du produit. Ce type de modeleur fournit une représentation complète de la forme du produit en offrant une information suffisante sur sa géométrie et sa topologie. Ainsi, la représentation géométrique est de plus en plus populaire dans les diverses applications utilisées dans le processus de développement du produit. En dépit de ces avantages, ces modeleurs tiennent compte seulement de la forme du produit mais n'incluent pas les différents types de connaissances exigées par les concepteurs pour le construire, l'évaluer et l'utiliser.

En particulier avec un modeleur géométrique le concepteur doit déterminer précisément chaque partie du produit dans le modèle choisi à l'aide des opérations qui sont propres à ce domaine : par exemple dans un modèle à base d'arbre Constructive Solid Geometry (CSG) à l'aide de combinaisons binaires de volumes de base ou bien dans un modèle purement géométrique, par la donnée d'une liste de points et de polygones construits à partir de ceux-ci. Le concepteur ne peut utiliser un niveau plus haut d'abstraction, où il pourrait s'attacher à décrire les propriétés de la scène à

laquelle il pense sans avoir à les traduire géométriquement [Shah 88]. Il est en effet plus simple de considérer une maison comme une association d'étages, d'un toit, de murs, de fenêtres, et cetera, plutôt que comme une liste de points interconnectés en polygones.

Les modeleurs géométriques présentent des avantages certaines : D'abord ils rendent possible la représentation de formes très complexes en 3 dimensions. Ensuite ils offrent au concepteur un grand degré de liberté. Cependant ils ont des limites. Les programmes sont en général très difficiles à apprendre et à utiliser. Pendant la conception, le concepteur peut passer beaucoup de temps à modeler même des formes simples. De plus il est difficile d'apporter des changements additionnels sur le modèle.

2.4.2.2. Les modeleurs basés sur les caractéristiques

Ces modeleurs, Feature-based modellers (FBM) sont de plus en plus utilisés pour modéliser des produits. L'un de leurs principaux avantages sur les modeleurs géométriques conventionnels, est leur capacité d'associer des informations fonctionnelles et techniques pour donner des informations sur le modèle produit. Ce peut être par exemple la fonction d'une partie du produit pour l'utilisateur final, ou une information sur la façon dont une partie du produit est fabriquée.

L'entité de base dans un modèle-feature est la caractéristique, définie comme une représentation des aspects de la forme d'un produit qui sont *mappables* à une forme générique et fonctionnellement significatifs pour une phase de cycle de vie du produit. Un aspect essentiel de la caractéristique est qu'elle a un sens bien défini, ou sémantique, pour une activité particulière du cycle de vie. L'introduction du concept de Caractéristiques permet d'associer la forme et la connaissance dans la compréhension d'un modèle CAO. Les caractéristiques sont des formes génériques ou spécifiques au moyen desquelles les concepteurs associent certains attributs et connaissances commerciales utilisés dans différentes phases du développement.

La modélisation basée sur les caractéristiques (feature-based) a permis aux ingénieurs de créer des modèles complexes en 3D en utilisant les outils software qui se rapprochaient le plus étroitement de la description de fabrication telle qu'elle existe dans le monde réel. Les caractéristiques de conception étaient essentiellement des

bibliothèques d'opérations prédéfinies qui pouvaient être utilisées pour construire un modèle solide. Ces opérateurs prédéfinis comprenaient des caractéristiques standard comme des trous, des bosses, des côtés, des fentes, des coins, des arêtes, et cetera. Le logiciel a utilisé le même langage que celui qui est familier aux ingénieurs quand ils décrivent, planifient et fabriquent leurs projets. Le logiciel a pris soin d'utiliser les mathématiques complexes nécessaires pour créer la géométrie.

Les modeleurs basés sur les caractéristiques présentent beaucoup d'avantages: Ils facilitent la modélisation de formes 3 dimensionnelles de complexité moyenne. De plus ils permettent au concepteur d'apporter des changements simples au projet. Enfin ils offrent un niveau élevé de stabilité à l'inverse de la modélisation géométrique.

Mais ces modeleurs ont des limites. D'abord, souvent ils ne peuvent pas manipuler des formes complexes et peuvent se bloquer. Ensuite, en raison de leur structure, il peut être souvent difficile, et même impossible, d'apporter des changements majeurs au projet. Enfin, ce type de modeleur est destiné à des concepteurs très disciplinés.

Deux aspects importants de la définition ci-dessus ne sont pas couverts par la plupart des feature-based systèmes courants de modélisation. D'abord la sémantique des caractéristiques (features) est mal définie, et cela limite la capacité de saisir le but du projet dans le modèle. Ensuite, la sémantique des caractéristiques (features) est mal maintenue, ce qui permet d'annuler le but du projet antérieur. On dit que de tels systèmes manquent de facilités pour maintenir leur validité.

2.4.3. Systèmes de collaboration

Un certain nombre de recherches se rapportant à des systèmes collaboratifs de développement du produit basés sur Internet, ont été entreprises par plusieurs auteurs. Passer en revue cette littérature a permis de mettre en lumière plusieurs exigences technologiques qui doivent être répertoriées afin de développer les technologies nécessaires à ce type de système. Il s'agit :

- de l'architecture du système d'information : les systèmes d'information et les applications de la conception sont intégrés à l'intérieur d'un cadre de façon

structurée et transparente, employant les protocoles de communication entre les éléments du système. [Molina et al. 95]

- des outils de communication : ces outils permettent la communication visuelle et audio entre les membres d'un groupe dispersés géographiquement.
- des outils virtuels de gestion du groupe : Ils coordonnent les membres dispersés du groupe.
- de la représentation géométrique de la conception : une application software facilite la visualisation de la conception du produit parmi les membres du groupe dispersés géographiquement.
- de l'intégration grâce au logiciel commercial CAO/ FAO (Fabrication Assistée par Ordinateur)/IAO (Ingénierie Assistée par Ordinateur) : des applications interfaces importent et exportent des dossiers à partir des systèmes commerciaux CAO/FAO/IAO.
- de la représentation de la connaissance : la documentation des leçons d'apprentissage et autres règles génériques qui sont stockées dans une base d'informations.
- des outils de gestion de projet : pour coordonner les activités de développement du produit.

En pratique, dans le domaine du CAO, on rencontre deux catégories principales de systèmes de collaboration:

- Les applications CAO qui utilisent espace de travail distribue (distributed workspace CAO) (par exemple : ARCADE – Advanced Revision CAD Environment) [Stork et al. 97],
- Les applications CAO qui utilisent les possibilités offertes par Internet [van den Berg 99] comme nous l'avons mentionné plus haut. (par exemple : ProjectPoint™ et Buzzsaw Standard™ [Autodesk 05]). En pratique, l'architecture collaborative qui est appliquée aujourd'hui est une combinaison de ce qui est cité ci-dessus du fait que les distributed workspaces sont accessibles désormais grâce au réseau ou aux VPNs (Virtual Private Networks).

Il existe de nombreux projets de recherche sur le thème de la conception Collaborative. Nous en décrivons brièvement un certain nombre.

2.4.3.1. CSCW-FeatureM

Le projet *CSCW-Featuerm* du domaine de la CAO a été publié en 1998 [Stork et al. 98]. L'architecture de ce projet repose sur la réplication et le traitement des données sur les clients. Les modifications se font l'une après l'autre après un accord préalable et le contrôle de cohérence se fait en fin de session en comparant les historiques. Le partage concerne le modèle entier. Il n'y a pas de serveur mais des clients *FeatureL* (basé sur le modeleur géométrique commercial ACISTM). Le composant de collaboration est propriétaire (Figure 2-3).

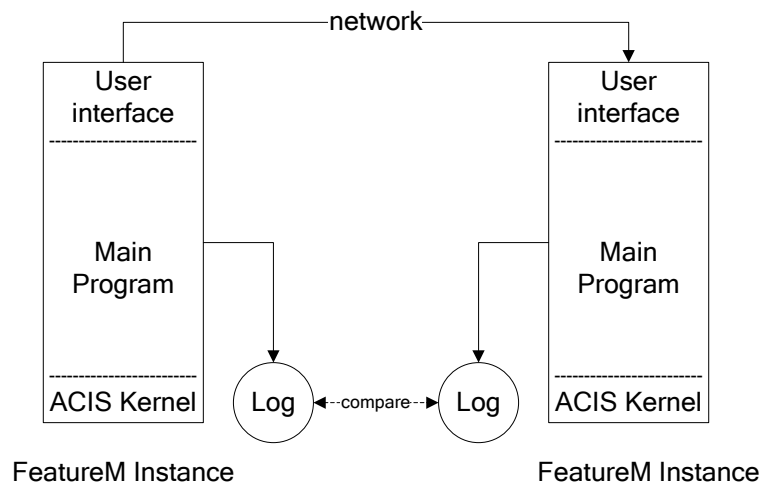


Figure 2-3 L'architecture de CSCW - FeatureM

2.4.3.2. COLLIDE

Le projet *CollIDE* du domaine de la CAO a été publié en 1998, [Nam et al. 98]. Son architecture repose sur la réplication des données sur les clients et les serveurs alors que les traitements se font sur les clients. Chaque concepteur travaille sur sa partie de conception et il/elle peut récupérer celle des autres. Le niveau de partage concerne la pièce d'un assemblage. Le serveur gère les sessions et sert de base de données. Les clients possèdent l'application de CAO (l'application commerciale 'Alias StudioTM') pour éditer les données ainsi qu'un système supplémentaire pour

visualiser la géométrie partagée. Le composant de collaboration est Group Kit [Roseman et al. 92].

COLLIDE est un ensemble d'espaces de travail partagés en 3D auquel peuvent accéder de multiples utilisateurs, mais il n'a pas la capacité de réaliser de création collaborative, d'effacer ou d'opérer la manipulation de la géométrie.

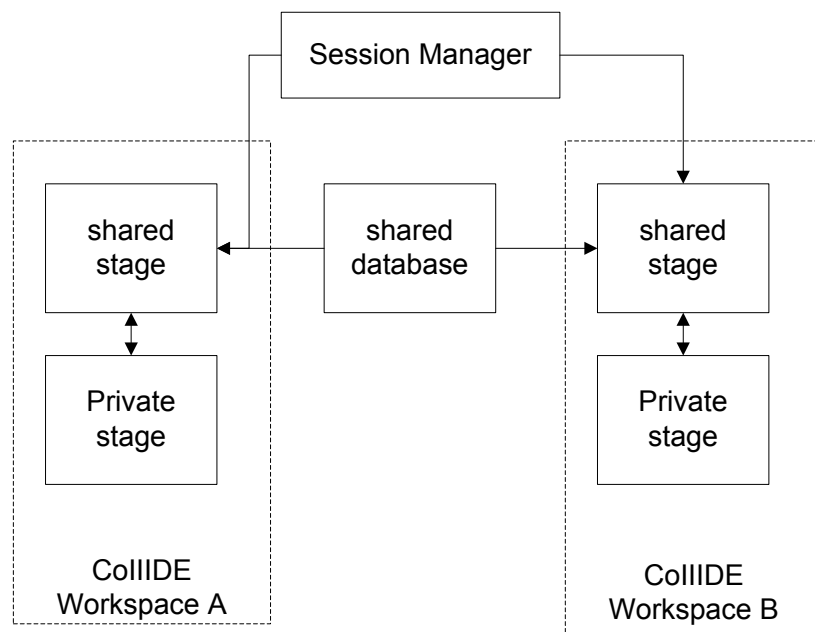


Figure 2-4 L'architecture de *CollIDE*

2.4.3.3. TOBACO

Le projet *TOBACO* du domaine de CAO a été publié en 1997 et 1999. Son architecture repose sur la réplication et le traitement des données sur les clients. Le principe de l'architecture est qu'il faut avoir accès au modèle pour pouvoir y apporter des opérations-modifications. Le niveau de partage concerne le modèle entier (Figure 2-5). Le serveur s'occupe de la gestion et de l'historique des sessions de conception. Les clients possèdent l'application CAO propriétaire [Dietrich t al. 97] (basée sur le modeleur géométrique commercial ACISTM ou AutoCADTM) augmenté d'une extension [von Lukas 99]. Le composant de collaboration est *TOBACO* [Dietrich t al. 97].

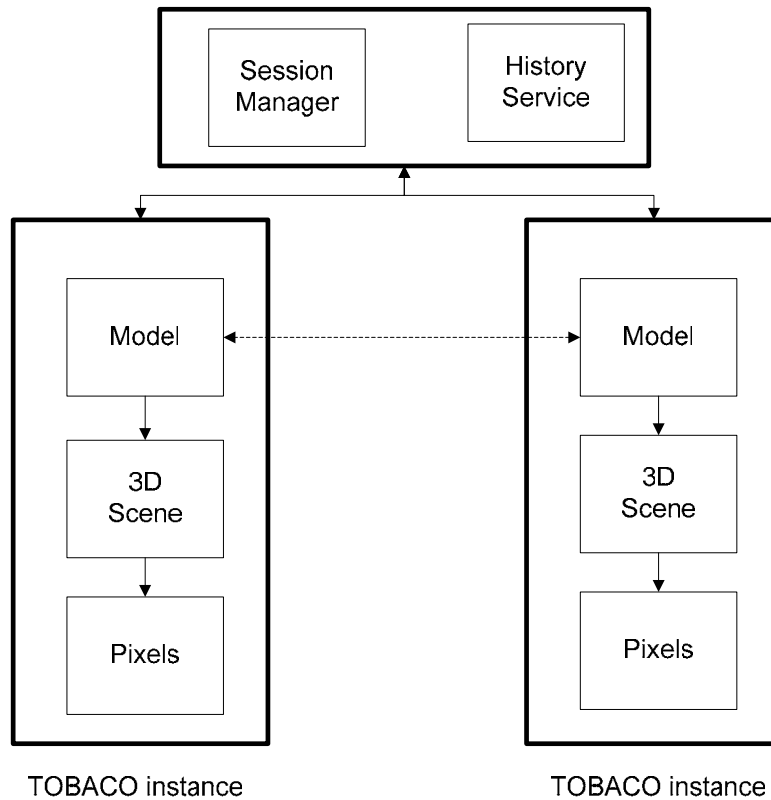


Figure 2-5 L'architecture de *TOBACO*

2.4.3.4. ARCADE

Le projet *ARCADE* (Advanced Revision CAD Environment) du domaine de la CAO a été publié en 1997. Son architecture repose sur la réplication et le traitement des données sur les clients. Les objets ne peuvent être modifiés que par un participant à la fois. Le niveau de partage concerne les objets seuls. Le serveur s'occupe de la gestion des sessions et de la base de données. Les clients possèdent l'application CAO propriétaire (basée sur le modéleur géométrique commerciale ACIS™) (Figure 2-6). Le composant de collaboration est propriétaire [Stork et al. 97].

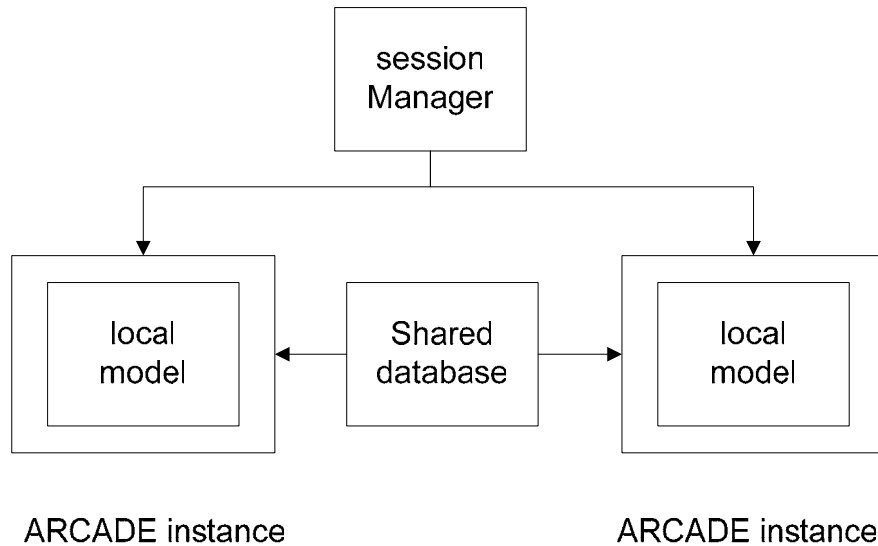


Figure 2-6 L'architecture de *ARCADE*

2.4.3.5. Collaborative Solid Modelling

Le projet '*Collaborative Solid Modelling*' (CSM) du domaine de la CAO a été publié en 1999 [Chan et al. 99]. Son architecture repose sur la réplication et le traitement des données sur les clients ainsi que sur les serveurs. Le concepteur doit travailler sur le modèle de conception pour pouvoir y apporter des opérations-modifications. Les opérations peuvent être bloquées par certains utilisateurs. Le système de '*Collaborative Solid Modelling*' permet la gestion de multiples versions. Le niveau de partage concerne le modèle entier. Le serveur gère les sessions et la géométrie. Les clients gèrent le navigateur Web. Le composant de collaboration est propriétaire (Figure 2-7).

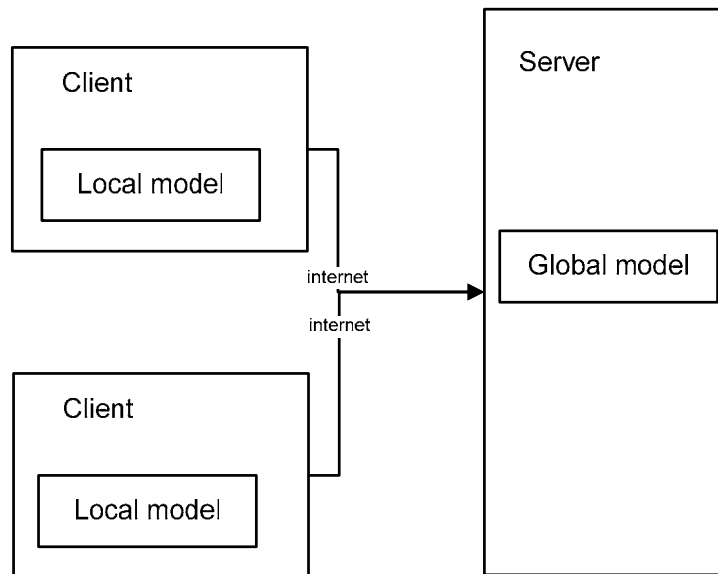
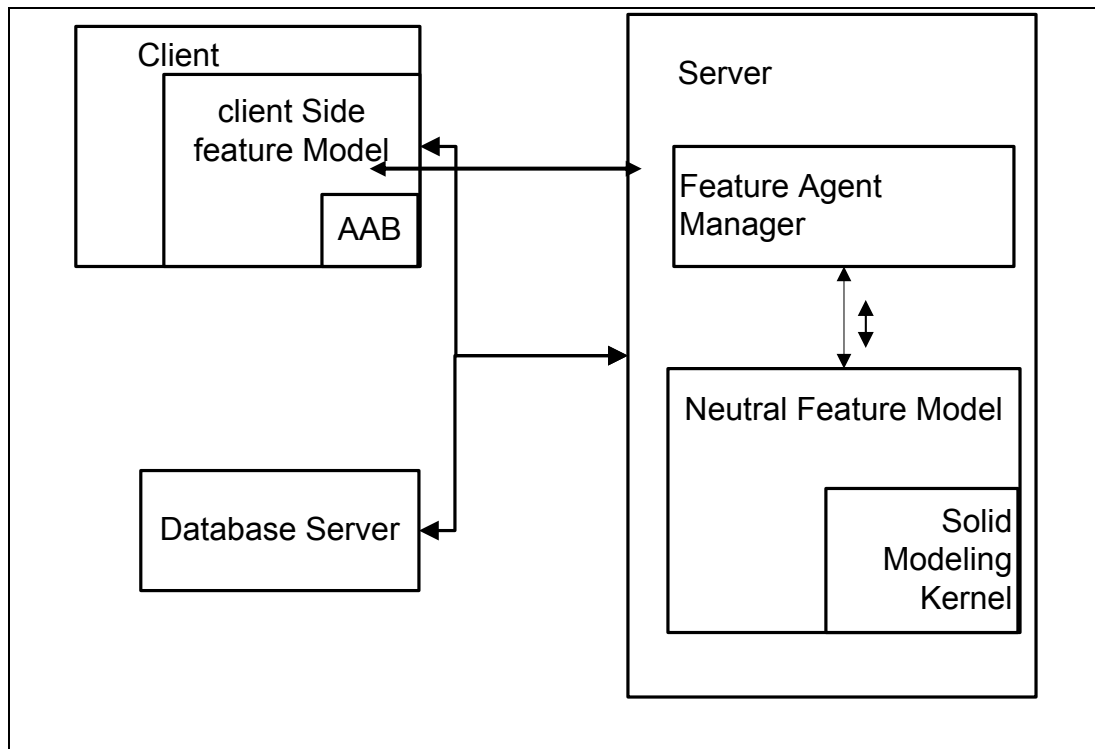


Figure 2-7 L'architecture de CSM

2.4.3.6. NetFeature

Le projet '*NetFeature*' du domaine de la CAO a été publié en 1999 [Lee et al. 99]. Son architecture repose sur la réplication et le traitement des données sur les clients et les serveurs. Ce modèle ne gère pas les conflits d'accès. Le niveau de partage concerne le modèle entier. Les serveurs s'occupent de la gestion de session et de la géométrie basée sur le modèleur géométrique commercial ACISTM. La base de données est située sur un autre serveur. Les clients gèrent le navigateur Web (visualisation grâce à Java3DTM). Le composant de collaboration est propriétaire (Figure 2-8).

Figure 2-8 Le'architecture de *NetFeature*

2.4.3.7. Web Based Collaborative CAAD

Le projet '*Web Based Collaborative CAAD*' du domaine de CAAO (Conception Architecturale Assistée par Ordinateur) a été publié en 1999 [Bajaj et al. 99]. Son architecture repose sur la répliquation des données sur les clients et un serveur alors que le traitement s'effectue sur un ou plusieurs serveurs. Chaque concepteur travaille sur une partie différente du modèle, et il/elle a la possibilité de définir des relations entre différentes parties. Le niveau de partage concerne une partie d'un modèle. Le serveur gère les sessions, les modeleurs géométriques *Plasm* et une base de données *BD2* pour la persistance et la gestion d'accès concurrents. Les clients possèdent une interface vers *Plasm* ainsi qu'un navigateur Web pour la visualisation des modèles (JavaTM et VRML). Le composant de collaboration est *Shastra* [Anupam et al. 94] (Figure 2-9).

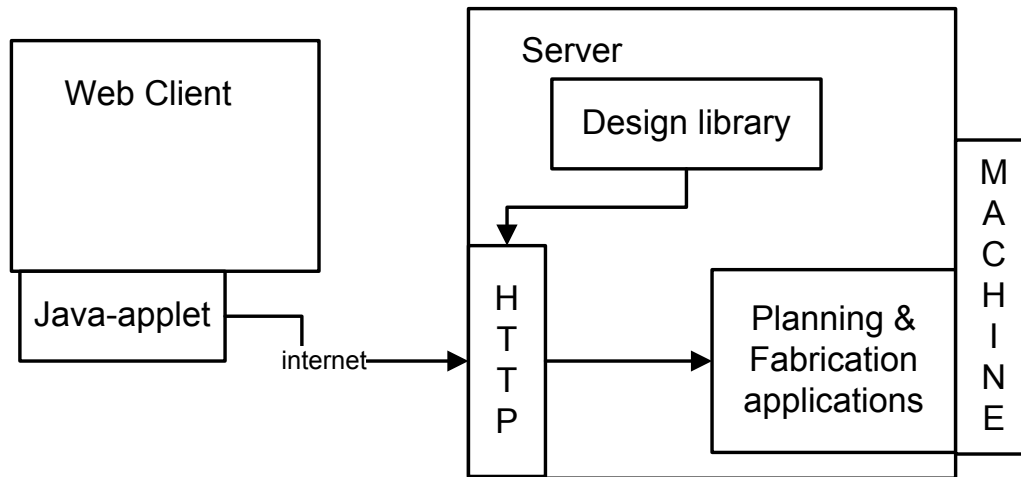


Figure 2-9 L'architecture de *WebCAD3D*

2.4.3.8. WebSpiff

Le projet *WebSpiff* du domaine de la CAO a été publié en 2000 [van den Berg 00a]. Son architecture repose sur la réplique ainsi que le traitement des données sur le serveur. Les opérations sont réalisées dans le produit *Co-Create*. Le niveau de partage concerne les données. Le serveur s'occupe de la gestion des sessions, de la géométrie (*Spiff*, basé sur le modèleur géométrique commerciale ACIS™) et du serveur Web. Les clients possèdent le navigateur Web (visualisation grâce à VRML et Java3D™ ou images fixes). Le composant de collaboration est propriétaire [van den Berg 00b]. WebSPIFF est un modèleur feature-based qui permet la création simultanée, l'effacement et la modification des caractéristiques par de nombreux utilisateurs [Bidarra et al. 01a], [Bidarra et al. 01b].

2.4.3.9. Syco3D

Le projet '*Syco3D*' du domaine de la CAO a été publié en 2001 [Nam et al. 01]. Son architecture repose sur la réplique ainsi que le traitement des données sur les clients. Chaque client travaille sur sa partie des données et peut récupérer celle des autres. Le niveau de partage concerne une pièce d'un assemblage. Les serveurs gèrent les sessions. Les clients possèdent l'application de CAO propriétaire pour éditer les données et une fenêtre supplémentaire pour visualiser les données et les structures partagées. Le composant de collaboration est '*GroupKit*' [Roseman et al. 92].

2.4.3.10. CyberCAD

Le projet 'CyberCAD' du domaine de la CAO a été publié en 2003 [Tay et al. 03]. Le CyberCAD peut être défini comme un logiciel CAO bon marché, amical pour l'utilisateur (user-friendly) et web-interactif, qui permet aux professionnels comme aux profanes de développer des modèles en 3D et de supporter une collaboration synchrone entre des utilisateurs géographiquement dispersés. Il a été développé sous une plateforme Java™, Java™ RMI, Java3D™ et Java™ Media Framework (JMF) directement. Le CyberCAD est devenu un objet orienté, simple, robuste, portable, multi-fils (multithreaded), à plateformes indépendantes, réparti et dynamique, (Figure 2-10).

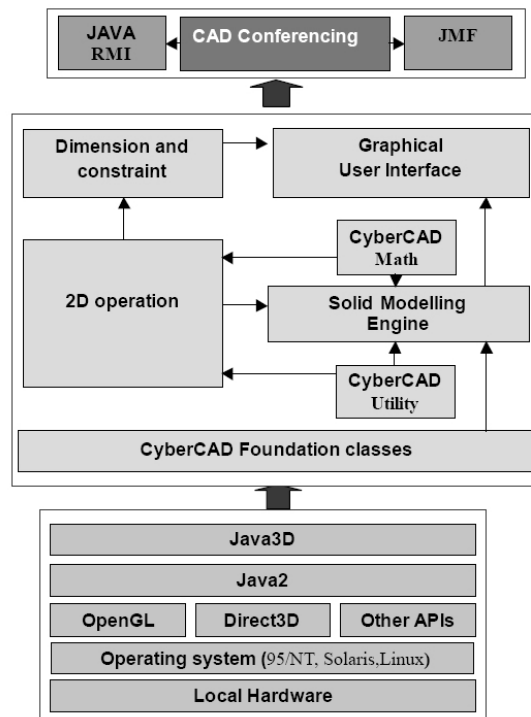


Figure 2-10 L' architecture de CyberCAD

2.4.3.11. Systèmes commerciaux de CAO collaboratifs

Plusieurs systèmes commerciaux sont également disponibles. Webscope™ [Webscope 01] est un système de communication des données d'un produit à temps réel qui permet à plusieurs utilisateurs de voir, annoter et rechercher des modèles et des documents CAO simultanément. Onespace™ Designer [CoCreate 01] permet de

visionner, choisir, éditer et sauvegarder des données de collaboration, mais il ne permet pas de créer une nouvelle géométrie. Au moyen de ce service, deux autres utilisateurs au plus, peuvent être invités à un meeting online. Les utilisateurs du meeting peuvent visionner et relever des modèles en 2D et 3D. Le système peut sauvegarder et partager les modèles des utilisateurs dans une base de données et spécifier qui a la permission de lire et de modifier le travail de conception. Le manager des données peut organiser des archives de projet en 2D ou 3D dans une base de données et aide à retrouver l'historique et la version du document. Le CollabCAD™ [CAD Group 06] est une solution CAO/FAO (Fabrication Assistée par Ordinateur) qui permet à de multiples concepteurs de travailler sur le réseau et d'accéder simultanément à la même conception pour la visionner et la modifier. Il offre aussi la possibilité de communiquer par Audio et Vidéoconférence.

Le CollabCAD™ facilite les capacités traditionnelles CAO/ FAO (Fabrication Assistée par Ordinateur) comme la conception, l'esquisse, la modélisation des surfaces, les caractéristiques de base NC, l'échange des données, l'écriture, et cetera. Le CollabCAD™ est basé sur des technologies ouvertes comme Java™, Java™ 3D, Open Cascade Geometry Kernel, Conferencing Tool, Text-to-speech synthetizer, etc. Le projet Catia™ [Dassault Systèmes 05] permet la conception collaborative instantanée et la construction simultanée informelle de parties ou surfaces utilisable dans tous les domaines. Les concepteurs peuvent partager des caractéristiques pour créer simultanément des parties ou échanger du savoir-faire (know-how) de façon synchrone ou asynchrone. La conception collaborative instantanée est graduable pour des communications de réseau égal à égal ou client/serveur, ce qui représente un avantage par rapport aux solutions de communication standard. Le projet DIVISION™ [PTC 01] sert de support pour la visualisation, la simulation et le réexamen de produits en 3D ou 2D.

Le projet Alibre Design™ [Alibre 05] permet aux utilisateurs de réaliser de manière collaborative une modélisation géométrique, mais un 'modelling Kernel' solide est nécessaire pour tout client capable de réaliser une modélisation. Une session de conception peut-être utilisée pour l'organisation d'une équipe virtuelle afin de concevoir des modèles 2D et 3D simultanément. La sauvegarde des modèles des utilisateurs permet d'accéder à ceux-ci et de les partager en toute sécurité. Ce système peut supporter le partage de messages entre utilisateurs.

2.4.3.12. Conclusion

Les systèmes de conception collaborative vus ci-dessus sont présentés de manière synthétique dans le tableau comparatif ci-joint (Tableau 2-2). La légende expliquant les caractéristiques des projets apparaissant dans le Tableau 2-2 se trouve dans le Tableau 2-3. En général on peut constater notamment que depuis 1999, les projets collaboratifs se concentrent principalement sur la conception collaborative sur le Web avec partitionnement des données entre les participants (clients). Les données et les traitements ont ainsi tendance à passer d'un mode de répllication (client) sur les clients vers un mode plus centralisé (serveur).

Projets Collaboratifs									
Projets	An	Dm	Dn	Tr	Ap	Nv	Sr	Cl	M
<i>ARCADE</i>	1997	C	C	C	V	O	S B	A	G
<i>TOBACO</i>	97-99	C	C	C	V	M	S H	A	G
<i>CollIDE</i>	1998	C	S C	C	P	P	S B	A F	G
<i>CSCW-Feature</i>	1998	C	C	C	-	M	-	A	G
<i>WBC CAAD</i>	1999	A	S C	S	P	M	S G B	W F	G
<i>NetFeature</i>	1999	C	S C	S	-	M	S G B	W	G
<i>CSM</i>	1999	C	S C	S	V	P	S G	W	G
<i>webSpiff</i>	2000	C	S	S	S	D	S G W	W	G
<i>Syco3D</i>	2001	C	C	C	P	P	S	A F	G
<i>CyberCAD</i>	2003	C	S	S	S	M	S G W	W	G

Tableau 2-2 Projets de recherche en conception collaborative

	Légende	
An	Année de publication	
Dm	Domaine	C pour CAO, S pour sculpture M pour modélisation 3D
Dn	Données	S pour sur le serveur C pour réplication sur les clients
Tr	Traitements	S pour sur le serveur C pour sur les clients
Ap	Approche	V pour verrouillage, P pour partitionnement, S pour sérialisation, E pour exécution réversible et – pour pas de gestion des conflits
Nv	Niveau	O pour objets, M pour modèle, A pour application, P pour pièce d'un assemblage D pour données
Sr	Cote serveur	S pour gestion des session, G pour gestion de la géométrie, B pour base de données A pour application de CAO ou autre, H pour historique des session, W pour serveur Web et – pour pas de serveur
Cl	Cote client	A pour application de CAO ou autre, B pour base de données, C pour composant de distribution, L pour client léger, F pour fenêtre supplémentaire W pour navigateur Web
M	Modeleur	G pour modeler géométrique F pour modeler feature-based D pour modeler déclarative
WBC CAAD	Web Based Collaborative CAAD	
CSM	Collaborative Solid Modelling	

Tableau 2-3 Légende du Tableau 2-2

2.5. La modélisation déclarative

Les modèles géométriques, classiques, malgré leurs avantages, sont relativement mal adaptés aux besoins des concepteurs car ils n'apportent pas d'aide à la conception. Ceci est dû à l'impossibilité de donner des descriptions incomplètes ou imprécises ainsi qu'à l'impossibilité de descriptions à plusieurs niveaux de détail.

Pendant la conception avec un modèleur conventionnel, le concepteur commence par avoir une idée pouvant amener à une solution (Figure 2-11). Il/elle la précise ensuite en déterminant l'ensemble des spécifications à respecter. A partir de là, il/elle conçoit mentalement un objet assez précis. De cet objet, le concepteur doit déduire la succession d'opérations élémentaires permises par le modèleur et nécessaires à la conception de l'objet. Durant une phase interactive, il construit petit à petit l'objet en suivant la planification des opérations élémentaires. Il reste encore à faire subir à l'objet obtenu un ensemble de tests de façon à vérifier sa validité. Des tests négatifs signifient que l'objet mental conçu par l'utilisateur ne répond pas correctement aux spécifications. Aussi, doit-il le corriger avant de mettre à jour interactivement l'objet modélisé. Ces opérations sont répétées jusqu'à ce que l'objet satisfasse l'ensemble des tests de validité (boucle "tests négatifs" Figure 2-11). Face à l'objet terminé, il est possible que le concepteur souhaite modifier les spécifications pour mieux répondre au problème qui lui est posé. Cela correspond au processus de conception (voir la boucle "processus de conception" Figure 2-11). Il doit réitérer l'ensemble des phases de la boucle "tests négatifs" de façon à obtenir un objet conforme aux nouvelles spécifications [Colin et al. 98].

Plus spécifiquement, les problèmes des modèleurs conventionnels (géométriques) ont été identifiés [Plemenos 95], [Sellinger 95], [Liege et al 97] comme il suit :

- La modélisation géométrique (MG) interactive peut être un processus difficile, ce qui aboutit à des possibilités d'erreurs dans la représentation géométrique.
- Il n'est pas possible de décrire un objet par un moyen imprécis puisque tous les traits de la représentation géométrique doivent être spécifiés précisément et explicitement.

- Il arrive souvent qu'une connaissance complète de la scène soit exigée si l'utilisateur veut modifier une partie.
- Il y a peu d'informations sur les aspects non- géométriques de la scène.

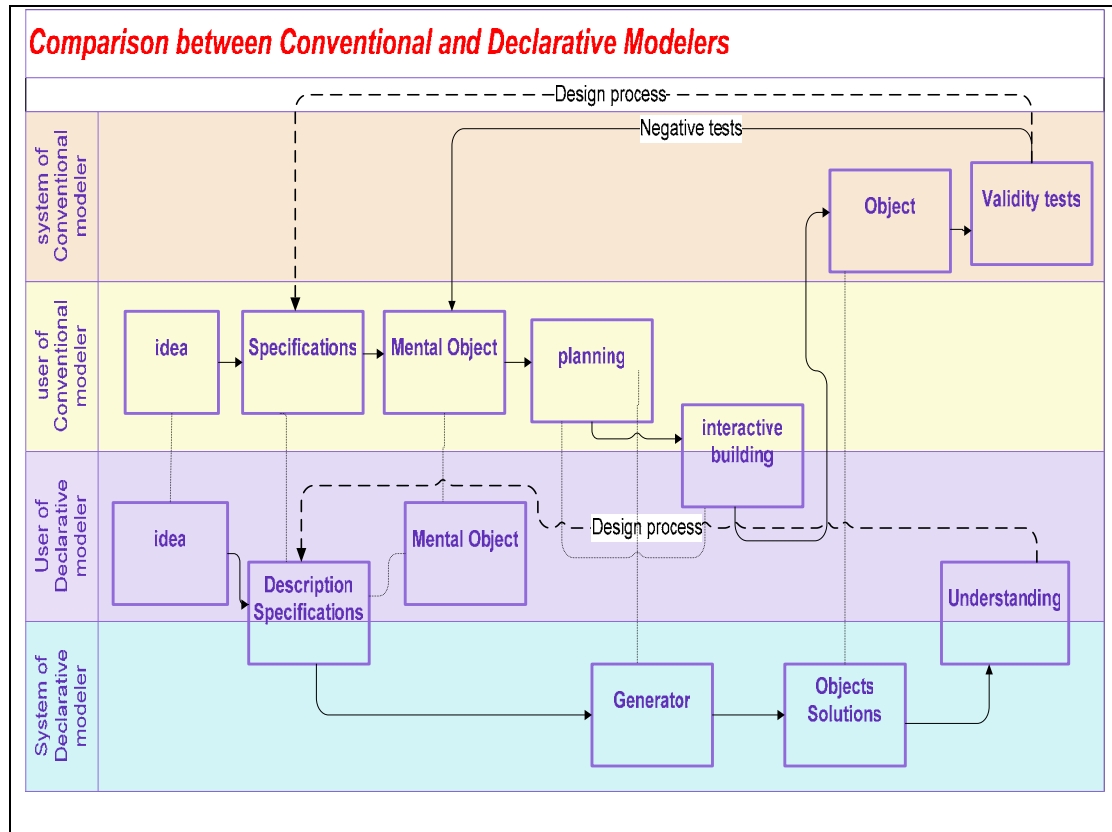


Figure 2-11 Comparaison entre modeleur conventionnel et modeleur déclarative

Dans les années 1980 un nouveau paradigme pour la modélisation géométrique a été développé [Lucas et al 90]. Ce nouveau formalisme est appelé la modélisation déclarative. L'objectif de la modélisation déclarative est de palier les inconvénients de la modélisation géométrique classique en permettant la description d'une scène par ses propriétés qui peuvent être imprécises et incomplètes [Plemenos 91], [Plemenos 95].

Plus précisément, la modélisation déclarative (MD) permet au concepteur de décrire une scène en utilisant ses caractéristiques et ses propriétés et l'ordinateur serait responsable de l'exploration de l'univers des solutions géométriques possibles et de la présentation des scènes qui vérifient la description (Figure 2-11 [Colin et al. 98]).

Après avoir fourni la description de la scène et la génération automatique faite, le concepteur pourrait choisir l'instance géométrique désirée, en utilisant un ensemble d'outils appropriés. Le concepteur serait donc libre de se concentrer seulement sur la tâche de création.

Les inconvénients de la modélisation déclarative sont les suivants :

- La description de scènes complexes est toujours une tâche difficile.
- La génération de scènes exige trop de ressources de l'ordinateur.
- L'exploration d'un grand nombre de scènes est une tâche pénible, donc l'utilisateur doit avoir la possibilité d'avoir une vue sur les traits intéressants.

Ces trois inconvénients correspondent aux trois phases principales de la modélisation déclarative, c'est-à-dire la description, la génération et la prise de connaissance.

2.5.1. Le fonctionnement général d'un modéleur déclaratif

Généralement, le fonctionnement d'un modéleur déclaratif est découpé en trois phases, plus ou moins séquentielles [Lucas et al. 95], [Desmontils 95], [Colin et al. 97] :

- la phase de description;
- la phase de génération;
- la phase de prise de connaissance.

La phase de description est évidemment le point de départ de tout travail avec un modéleur déclaratif. L'utilisateur décrit comment il perçoit la scène. Les moyens de description peuvent être multiples et se compléter. Certains modéleurs déclaratifs utilisent des langages de description plus ou moins proches du langage naturel. Cette description est ensuite traduite en un modèle interne par le modéleur déclaratif qui l'utilise pour engendrer les scènes correspondantes.

La phase de génération correspond à la production des scènes solutions satisfaisant la description. A partir du modèle interne issu de la description, le moteur

de génération va explorer un univers de scènes potentielles pour y trouver les scènes solutions c'est-à-dire celles vérifiant toutes les propriétés exprimées par l'utilisateur. Cette phase est véritablement le cœur du modelleur. Les capacités du moteur de génération caractérisent complètement le modelleur déclaratif. Son efficacité est souvent liée à la rapidité de traitement qu'il peut avoir sur ces données. Et enfin, la souplesse du modelleur dépend de sa capacité à intégrer de nouveaux paramètres de la description alors qu'il est déjà en cours de génération.

La phase de prise de connaissance : les scènes solutions sont proposées à l'utilisateur. L'utilisateur peut vérifier que les propriétés qu'il a utilisées conduisent bien à des scènes satisfaisantes. Cette prise de connaissance peut être faite sous forme textuelle. La prise de connaissance peut être également visuelle : les scènes solutions sont traduites à partir des données manipulées de façon interne par le modelleur, en scènes géométriques. Ces scènes géométriques sont ensuite affichées à l'aide d'un algorithme de visualisation. Le modelleur déclaratif peut également fournir des informations supplémentaires (statistiques sur les scènes engendrées), ou améliorer l'aspect visuel. [Colin 97], [Kamada et al 88], [Plemenos 91], [Plemenos et al 96], [Mounier 98a].

Suivant le processus déclaratif, le concepteur décrit intuitivement son idée par des propriétés (phase de description) tandis que plusieurs solutions sont produites qui satisfont la description d'entrée (phase de génération). Ces solutions sont converties en des modèles géométriques et sont présentées au concepteur (phase de prise de connaissance) via des outils graphiques (sous phase de visualisation) et éventuellement, elles peuvent être évaluées (sous phase d'évaluation) selon son anticipation premièrement déclarée. L'utilisateur peut sélectionner une solution ou modifier les propriétés d'entrée et relancer le processus. Il peut continuer ainsi jusqu'à ce qu'il obtienne la scène solution souhaitée.

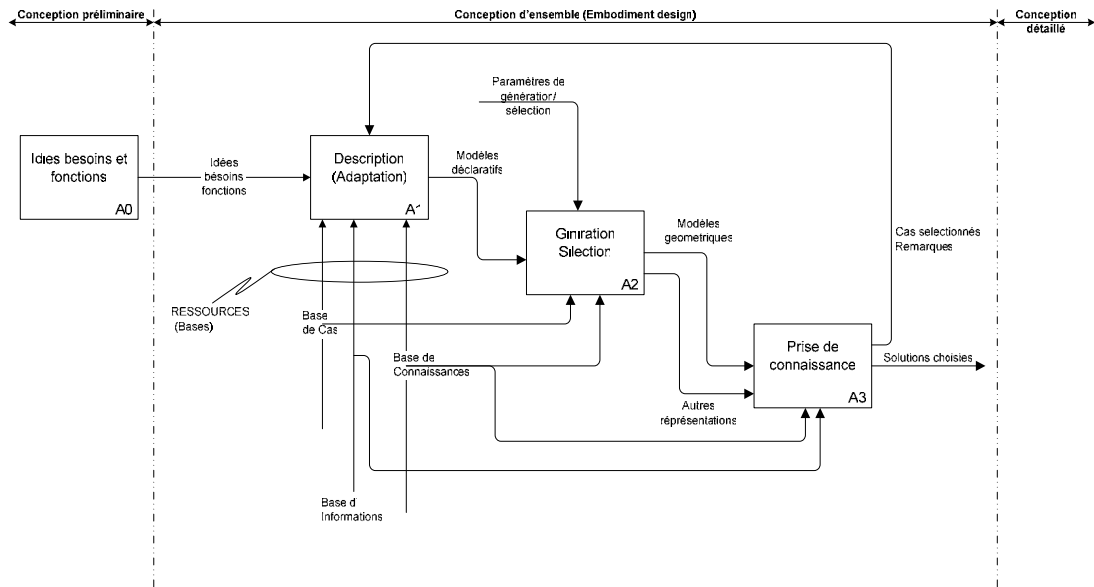


Figure 2-12 Modélisation détaillée du processus de conception déclarative

La Figure 2-12 montre en détail le schéma du processus de conception déclarative et les informations d'entrée et de sortie de chaque phase.

Pendant le cycle de conception déclarative, le modèle du produit (la scène) passe par certains niveaux de transformation: il commence comme des idées et restrictions mentales, il est traduit en modèle déclaratif (haut niveau d'abstraction), il est transformé en modèle géométrique qui peut être manipulé par des systèmes de visualisation de CAO. Plusieurs modeleurs déclaratifs ont été développés :

- *Polyformes*. Un modeleur qui concerne la construction de solides polyédriques à partir de combinaisons de polyèdres [Martin e al. 88].
- *CCAD (Cooperative Computer Aided Design)*. Le projet de conception collaborative assistée par ordinateur (CCAO) combine les possibilités d'activités de conception manuelles et automatiques [Kochhar 94]. Dans ce projet, l'utilisateur exprime des décisions de conception initiale sous la forme d'une conception partielle et définit une série de propriétés que la conception finale doit avoir. Ensuite, le système génère des développements alternatifs partiels de la conception initiale sous la forme d'un langage. Il structure les résultats dans un cadre spatial dans lequel l'utilisateur va explorer les différentes alternatives.

Celui-ci va sélectionner la meilleure conception partielle, la retoucher manuellement et ensuite recourir au développement automatique ultérieur. Ce processus continu jusqu'à ce que la conception soit complète.

- *SpatioFormes*. Ce projet permet la description et la génération de scènes tridimensionnelles modélisées par des matrices de voxels [Poulet 94]. *SpatioFormes* travaille à partir d'un arbre d'exploration pour proposer les scènes résultats.
 - *FiloFormes*. Est un modèleur déclaratif de configurations de segments de droites pouvant être utilisé dans la génération de tableaux de fils [Pajot-Duval 94]. L'utilisateur décrit l'allure générale du tableau et le modèleur propose les différentes configurations de segments de droites possibles
- *MegaFormes*. Il modélise des monuments mégalithiques et les visualise de manière virtuelle [Poulet et al. 96].
- *DES²MON*. Un modèleur dont l'objectif est d'assister à la conception de scènes tridimensionnelles en ce qui concerne le dimensionnement et de placement d'objets dans un univers fini. Le modèleur fournit au concepteur une bibliothèque d'objets décrivant leurs principales caractéristiques et de contraintes spatiales (pour gérer le positionnement relatif des objets), énoncées dans un langage de haut niveau. Ce langage permet l'expression de la description des scènes en langage naturel, par exemple "*le chaise est à côté de la table*" [Kwaiter 98].
- *UrbaFormes*. Ce modèleur propose d'établir de manière déclarative un itinéraire permettant de découvrir les aspects urbains d'une ville donnée [Mounier 98b]. Ces parcours peuvent être obtenus de façon dynamique.
- *BatiMan* qui se concentre sur la construction de bâtiments décomposables en un ensemble fini d'éléments en introduisant des méthodes d'apprentissage [Champiaux 98].
- *Multiformes* [Plemenos 95] [Bonnet 99] [Ruchaud 01] modélise des scènes complexes (par exemple des habitations) décrites selon la technique de la décomposition hiérarchique. Suivant cette technique, afin de décrire une scène, on la décompose en pièces qui sont récursivement décrites (jusqu'à un certain niveau de détail).

- *Conception urbaine*. Ce projet [Liège 96] met en évidence les notions d'éléments urbains qui peuvent être combinés afin de composer des figures urbaines plus complexes, [Gaildrat 03].

Ces modeleurs se sont concentrés sur la phase préliminaire de la conception des scènes, de ce fait, caractérisant les logiciels développés en tant que systèmes de conception déclarative assistée par l'ordinateur. Les modeleurs déclaratifs sont classés en deux catégories : les génériques, qui sont indépendants du domaine d'application et ceux qui sont dédiés à un domaine spécifique de connaissance.

2.5.1.1. La modélisation déclarative par décomposition hiérarchique

La modélisation déclarative par décomposition hiérarchique a été introduite dans le but de fournir un cadre plus structuré pour la modélisation déclarative [Plemenos 91]. Son approche se base sur la notion de structure dans une scène. Nous pouvons considérer qu'il existe deux catégories de scènes :

- Les scènes faciles à décrire. Une scène est facile à décrire si ses caractéristiques peuvent être exprimées dans le modeleur à l'aide d'un petit ensemble de propriétés et d'éléments de base.
- Les scènes difficiles à décrire. Une scène est difficile à décrire si ses caractéristiques ne peuvent être décrites simplement à l'aide des propriétés proposées par le modeleur. Dans ce cas, on décompose la scène en un ensemble de sous- scènes, chacune des sous-scènes pouvant être décomposée suivant le même critère de complexité. La description d'une scène complexe prend donc un aspect arborescent.

La modélisation déclarative par décomposition hiérarchique a pris corps dans un projet dénommé *MultiFormes* [Plemenos 91]. La structure générale de MultiFormes rappelle celle d'un système expert : la description du modèle est une combinaison des *faits* (sous- scènes) et des *règles* qui régissent les relations/propriétés des sous-scènes. Un *solveur de contraintes* est utilisé par le *moteur de génération de scènes* pour trouver les scènes-solutions à sélectionner et afficher celles qui sont pertinentes. La description d'une scène est introduite et convertie dans un langage formel du type Prolog. La description convertie est appelée *modèle interne* ou

Représentation Déclarative Interne de la scène - RDI (Internal Déclarative Représentation – IDR).

Dans la version actuelle de MultiFormes, les scènes- solutions se trouvent par résolution de contraintes. Suivant [Bonnefoi 99], le système de contraintes est celui des contraintes sur un domaine fini (CLP(FD) Constraint Logic Programming in Finite Domain) et ces contraintes sont formulées sous une forme dite primitive afin d'en uniformiser le traitement. Les méthodes de résolution suivent le modèle de satisfaction de contraintes (CSP – Constraint Satisfaction Problem) et s'appuient sur les techniques de consistance d'arc et utilisation de différentes heuristiques. Dans la phase de génération de scènes, la notion de parallélisme est introduite : La génération est parallélisée de manière transparente pour l'utilisateur.

2.5.1.2. Décomposition hiérarchique dans MultiFormes

MultiFormes propose essentiellement trois type de propriétés :

- Les propriétés de taille inter – dimensions. Ces propriétés portent sur les dimensions d'un seul élément de la scène. Elles en indiquent les rapports. Exemple : "PlusHautQueLarge", "PlusHautQueProfond" etc.
- Les propriétés de forme. Ces propriétés spécifient l'allure générale d'un élément ou d'un groupe d'éléments si elles sont appliquées à un élément composé. Ces propriétés ne sont pas prises en compte par la résolution, elles ne sont appliquées qu'à l'obtention d'une solution pour en améliorer l'aspect. Exemple : "HautArrondi", "HautCreux" etc.
- Les propriétés de placement ou de position. Les propriétés de placement travaillent sur un ou plusieurs éléments de la scène. Quand elles travaillent sur plusieurs éléments ces éléments doivent être les enfants d'un même autre élément de la décomposition hiérarchique. Exemple : "CollerAGauche", "PoseDessus" etc.

La description hiérarchique permet la représentation de propriétés de la scène (relations sur et entre les scènes- composants) sur différents niveaux de détail. Il y a une certaine classification de propriétés utilisées : une scène est décrite par des propriétés de position (parmi les autres composants, par exemple 'à gauche de', etc.), de forme (arrondi, creux etc.), et de taille ("plus haut que large" etc.). Sur l'arbre de

décomposition d'une scène, les propriétés de forme et de taille décrivent la situation locale du nœud mais aussi, certaines propriétés sont héritées des scènes-parents. Les propriétés de position expriment des relations entre les sous scènes.

Dans la Figure 2-13, la scène 'Habitation' est analysée : elle est décomposée en 2 sous- scènes : La 'Maison' et le 'Garage'. La sous-scène 'Maison', à son tour, en étant complexe, est décomposée en les espaces habitables de la maison.

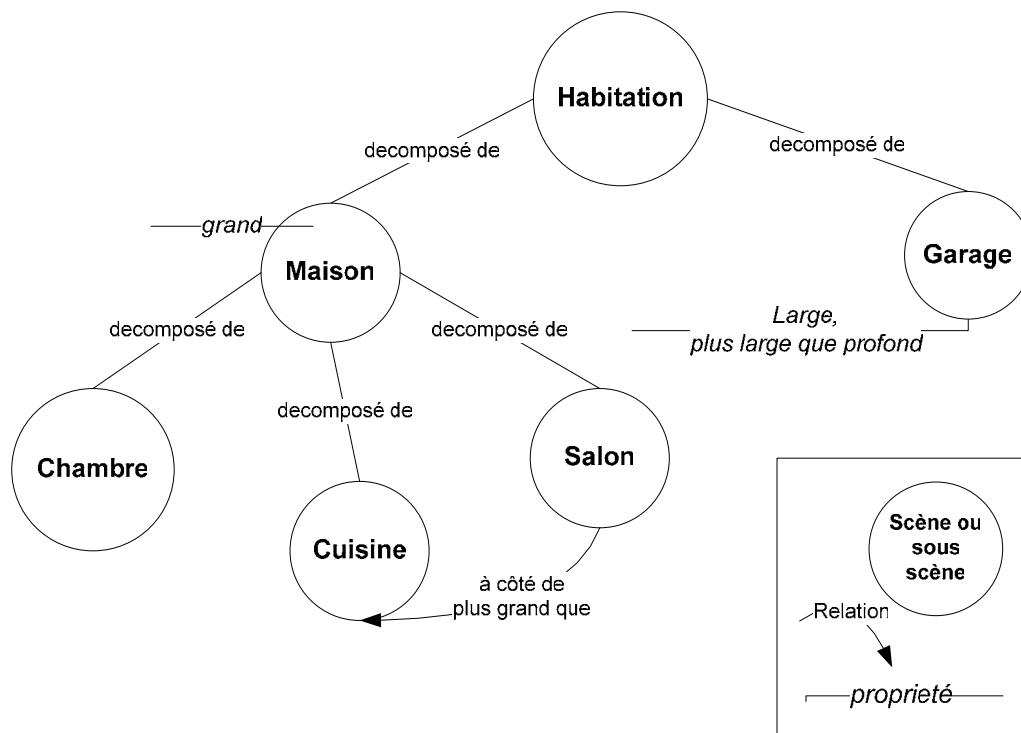


Figure 2-13 La décomposition hiérarchique de la description d'une scène

Les propriétés et les relations qui figurent (par exemple, 'Grand', 'Large', 'Plus Large que Profond' ou 'à Côté de', 'Plus Grand que') se traduisent en règles exprimées sous forme de prédicats de la logique du 1er ordre (par exemple $Grand(Maison)$, $A_côté_de(Cuisine, Salon)$). Ces prédicats constituent le modèle interne de la scène qui est utilisé à la phase suivante, la génération de solutions.

2.5.2. MULTICAD

MultiCAD n'est pas seulement un modéleur déclaratif mais il est un cadre d'architecture de logiciel pour le développement des systèmes multimédia et intelligents d'information dédiés à la Conception Déclarative Assistée par l'Ordinateur – CDAO, [MIAOULIS 02]. Plus précisément, MultiCAD propose une plate-forme des systèmes d'informations basés sur la modélisation déclarative par la décomposition hiérarchique -MDDH des scènes.

Les spécifications d'une telle plate-forme sont définies au sein d'un projet de recherche soutenu par le laboratoire MSI (Méthodes et structure Informatiques) de l'université de Limoges avec l'équipe de recherche « Intelligent Information Systems Engineering Research Team» du département d'Informatique de l'Institut d'Education Technologique) d'Athènes [Miaoulis 02].

2.5.2.1. L'architecture MultiCAD

MultiCAD propose une architecture multicouche qui comporte les couches principales suivantes

- *La couche d'interface* qui entoure des fonctions telles que la visualisation intelligente des modèles et des documents de scène, la création et édition des modèles et la description, la formulation de la demande (formulations de SQL - Standard Query Language), SQL spatial ou recherche des textes libres), la navigation et lecture rapide des bases de données acquisition et édition des différents types de connaissance et l'information, application et interaction commande.
- *La couche de traitement* comporte des fonctions telles que de génération ou de compréhension des différents niveaux des modèles, convertissant les différents types du même niveau du modèle (par exemple de modèle abstrait de la scène en modèle géométrique).
- *La couche de gestion de l'information et de la connaissance* qui emploie des outils pour la structuration, la gestion, la recherche et l'exploitation de la base de données que MultiCAD incorpore.

Ces couches sont projetées aux trois phases du cycle déclaratif de conception comme cela est montré par la Figure 2-14. Afin d'être plus précis, les mécanismes proposés à chaque couche traitent les tâches qui ont lieu à chaque phase du processus déclaratif. Dans la section qui suit, nous expliquons en détail ces tâches, adaptées au cadre que MultiCAD établit.

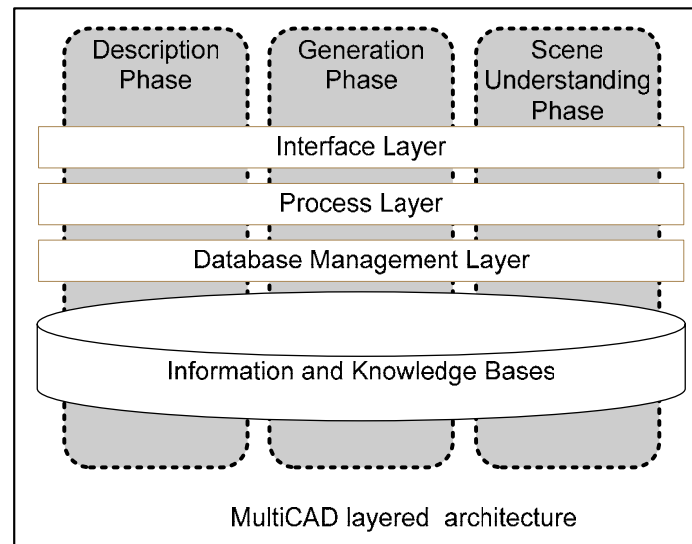


Figure 2-14 L'architecture de MultiCAD projetée auprès du cycle déclaratif

2.5.2.1.1. Phase de Description

Le concepteur décrit à un haut niveau d'abstraction la scène désirée en définissant les entités (objets) qui la décomposent, leurs propriétés et leurs relations. En fait, les propriétés et les relations des objets définissent les contraintes de la scène

La description de la scène est représentée des plusieurs manières :

- Représentation sous la forme d'un arbre. La technique de la décomposition hiérarchique renforce la notion de descendance qui est exprimée par un arbre dont les nœuds sont les objets. A chaque nœud, sa propre description est assignée et entre les nœuds existent des relations. Respectant la nature de l'arbre, chaque nœud non feuille est relié par défaut avec la relation « *est décomposé de* » avec ses nœuds enfants. La notion d'héritage est aussi respectée, donc la description de

chaque nœud père influence ses nœuds enfants. De cette façon, nous acceptons que l'arbre formé soit hiérarchique, aussi.

- Représentation textuelle, exprimée dans le langage formel défini par le modelleur MultiFormes. Il existe aussi un mécanisme qui permet la conversion de cette représentation en celle de l'arbre hiérarchique (pour garder la conformité des informations concernant la description des scènes) et vice versa.
- Représentation dans une base de données relationnelle objet. Les éléments divers de la description d'une scène sont mémorisés dans la base sous la forme d'enregistrements. La structure de la base consiste en des tableaux pour garder les informations concernant respectivement la scène, ses objets, les propriétés des objets et leurs relations.

2.5.2.1.2. Phase de génération

Le moteur de génération produit des solutions alternatives de modèles qui satisfont les contraintes posées par la description de départ. Cette phase commence par l'évaluation des propriétés déclaratives du modèle abstrait pour produire des solutions bien concrètes (c'est-à-dire, des modèles géométriques). Selon la méthode de génération utilisée, les scènes sans cohérence avec les contraintes de départ peuvent être éliminées soit à chaque nouvelle production soit à la fin de la production de l'ensemble des solutions.

Parmi les différents systèmes prototypes basés sur l'architecture de MultiCAD, des techniques variées ont été utilisées pendant cette phase de la modélisation déclarative : le modèle de satisfaction de contraintes (Constraint Solving Problem) [Bonnefoi 99], [Ruchaud 01], [Fribault 03], des algorithmes génétiques [Vassilas et al 02], des réseaux de neurones [Plemenos et al 02], algorithmes génétiques multi-objectifs [Makris 05].

Les avantages et les désavantages de chaque méthode utilisée ne sont pas analysés dans ce mémoire parce que notre proposition est indépendante de la méthode de génération utilisée.

2.5.2.1.3. Phase de prise de connaissances

Les solutions générées (sous la forme des modèles géométriques) sont visualisées en deux ou trois dimensions et sauvegardées respectivement dans des fichiers de format correspondant. Il existe divers formats de visualisation testés, par exemple des bitmaps [Miaoulis 02], VRML [Vassilas et al 02], DXF (fichier AutoCAD) [Makris et al 03].

Après la visualisation et selon la technique de génération utilisée, l'évaluation des solutions se met en place. Ici aussi, suivant la méthode d'évaluation appliquée, l'évaluation est caractérisée comme automatique (faite par le logiciel), manuelle (faite par le concepteur, selon ses propres critères) ou semi automatique (le logiciel propose des scores pour certaines solutions et le concepteur évalue celles qui restent ou même il peut changer celles qui sont déjà évaluées) et comme unitaire (évaluation d'une seule solution) ou collective (évaluation de toutes les solutions visualisées, en même temps). Un exemple d'évaluation des solutions est présenté par [Vassilas et al. 02] qui utilise la technique des algorithmes génétiques afin de produire des ensembles des solutions (appelés générations). L'évaluation est collective et manuelle. Un exemple d'évaluation automatique des solutions est présenté par Makris [Makris 05].

2.5.2.1.4. La gestion des informations et des connaissances

L'architecture MultiCAD insiste sur la gestion des informations et des connaissances dans un système de modélisation déclaratif pour en faire un système multimédia et intelligent d'information consacré à la Conception Déclarative Assistée par l'Ordinateur (CDAO). Les informations et les connaissances sont mémorisées dans la base de données relationnelle objet qui est composée, plus précisément, de quatre bases logiques :

- La *base de Projets*, qui mémorise des informations générales concernant le projet en cours du client, comme le nom du client, de dates de création du projet, des données financières etc. Un projet peut se référer à plusieurs scènes pour justifier les vrais besoins du concepteur à des conditions réelles. A ce point, nous devons clarifier la notion de la décomposition. Un projet peut être traduit en une seule scène ou un ensemble de scènes. La décision est prise par le concepteur et dépend de son point de vue de la notion de décomposition : un projet peut être vu comme

une seule scène qui est décomposée en plusieurs sous-scènes ou il peut être vu comme des scènes indépendantes. En fait, par rapport à ces besoins d'analyse des données, le concepteur décide du niveau d'abstraction de la décomposition. La différence entre les deux cas est la dépendance ou non des scènes, traduite comme l'influence du positionnement ou de dimensionnement d'une scène à l'autre.

- La *base de Scènes* qui contient des informations générales de chaque scène (nom de la scène, sa description textuelle, dates, noms des concepteurs, etc.), la description des objets composant la scène, leurs propriétés et leurs relations.
- La *bibliothèque Multimédia* est une base qui mémorise les modèles géométriques (tridimensionnels), des images ou des fichiers multimédia (par exemple des vidéos) associés à la scène.
- La *base de Connaissances* concerne la typologie des objets, des propriétés et des relations utilisées dans la description de la scène. La typologie concerne un certain domaine d'application, donc, dans la base de connaissances, il peut y exister différents types du même élément de la description abstraite (par exemple, en architecture l'objet 'maison' signifie un ensemble d'espaces habitables mais pour la mécanique, il signifie un ensemble de murs et colonnes).
- Les différents types d'objets correspondent à des objets élémentaires, à des objets primitifs géométriques. En ce qui concerne les types de propriétés, les aspects suivants sont pris en compte : la forme des objets (aspect morphologique), leur propre position (aspect spatial).
- Comme de types de relations nous pouvons distinguer des relations de classification (classe d'inclusion etc.), de métonymie, temporelles, spatiales, relations d'influence, relations de dépendance, de corrélation, et cetera [Miaoulis 02].

Le modèle conceptuel de la base de données est basé sur le modèle relationnel objet étendu [Miaoulis 02]. Suivant ce modèle, la théorie du modèle relationnel objet est étendue afin d'inclure des notions de la programmation orienté objet mais appliqué à une base de données relationnelle, comme la notion d'agrégation, d'héritage, de polymorphisme et cetera.

Dans ce cadre, la description de la scène est vue comme un ensemble d'objets qui ont des propriétés et sont reliés à des relations. Donc, la base de scène comprend quatre tableaux principaux pour mémoriser la scène, les objets qui la composent, leurs propriétés et leurs relations définissant un modèle conceptuel respectif appelé le méta-modèle de la Scène.

De la base de scènes, la typologie des éléments de la scène (par rapport au domaine de connaissances) est extraite qui constitue la connaissance concernant les scènes. Donc, à un niveau d'abstraction plus élevé, la base de connaissances contient une série de types d'objets, de types de propriétés et de types de relations auxquels les éléments de la base de scènes sont projetés. Le modèle conceptuel de la représentation des connaissances comprend trois tableaux respectifs. En plus, un tableau est préservé pour la mémorisation du domaine de connaissances.

La Figure 2-15 montre le méta-modèle de la scène et le niveau de modélisation de la connaissance des descriptions des scènes mémorisées.

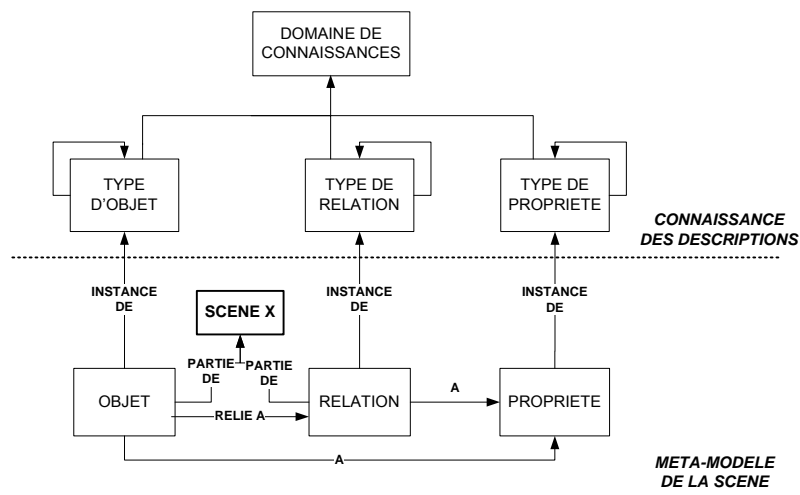


Figure 2-15 Le méta-modèle de la scène et la connaissance des descriptions dans MultiCAD

Enfin, la Figure 2-16 présente de manière analytique l'architecture de MultiCAD.

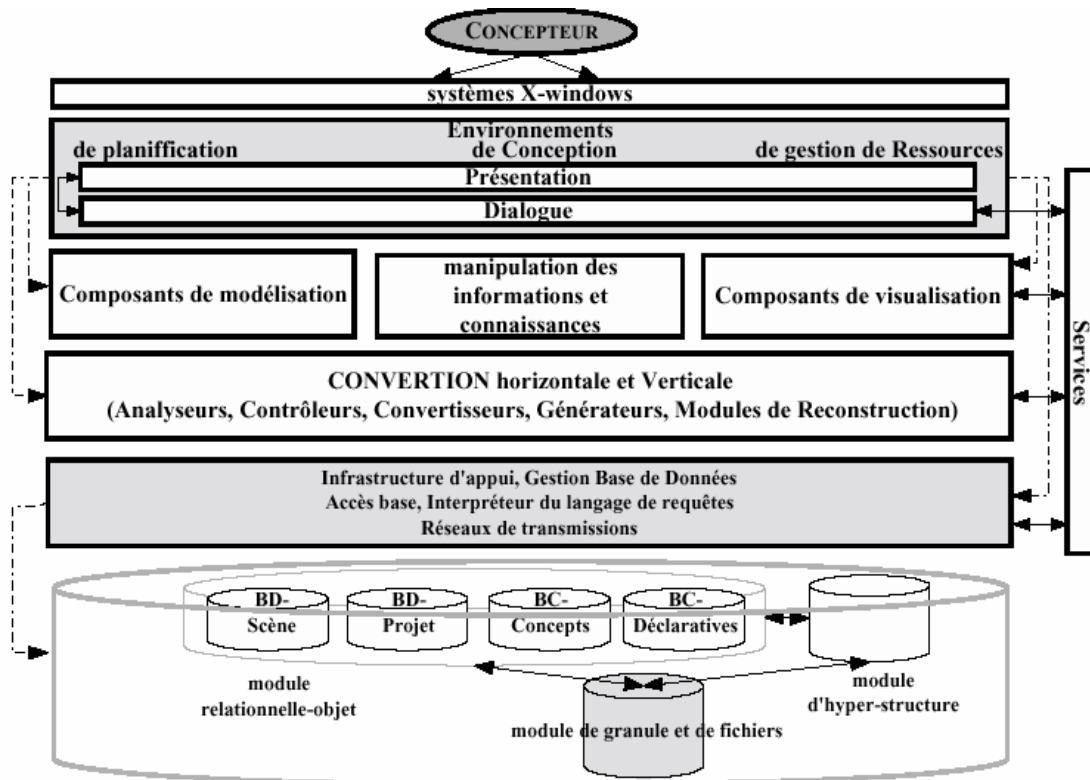


Figure 2-16 L'architecture de MultiCAD vue d'ensemble

2.5.2.2. Le cadre DKABM (DKABM Framework)

Durant la phase de description, une scène pourrait être incomplète, ou contenir des paramètres qui déclencheraient des conflits, ou simplement elle pourrait être exprimée de nombreuses manières. Pour éviter un tel inconvénient, il faut adopter un cadre pour fournir un prototype (template) pendant la description de la scène. Parallèlement, pour les besoins de cette thèse, il nous faut un cadre capable de combiner à la fois la représentation de l'information utilisée dans une modélisation de scène déclarative et dans une modélisation de scène conceptuelle. Nous adoptons le Declarative Knowledge Framework for Architecture-oriented Building Modelling (DKABM) qui incorpore la conception conceptuelle des scènes [Makris et al 03] [Ravani et al 03]. Ce cadre offre la typologie des entités, propriétés et relations, utilisée afin d'identifier les relations sémantiques nécessaires pour définir les modèles d'architecture orientée vers le bâtiment. De plus des contraintes architecturales ont été déterminées pour valider la cohérence et la sémantique des bâtiments représentés.

L'adoption du DKABM produit des descriptions architecturales de bâtiments cohérentes. Le cadre DKABM satisfait aux exigences suivantes :

- Intégrer les connaissances de conception architecturale.
- Permettre des descriptions de bâtiments sémantiquement significatives.
- Définir un cadre d'information de représentation des bâtiments dans une structure de données appropriée pour produire des bâtiments cohérents (du point de vue architectural).

2.5.2.2.1. Analyse du cadre

Le DKABM contient toutes les informations nécessaires pour décrire de manière consistante et complexe un bâtiment au point de vue architectural. Un bâtiment peut être décrit de la façon suivante : Il est formé d'espaces organisés en étages. Chaque espace ou groupe d'espaces remplit certaines fonctions. Ces espaces sont liés les uns aux autres. Avec le cadre la description d'un bâtiment est organisée en catégories d'objets qui partagent les mêmes propriétés et relations. L'information est organisée hiérarchiquement formant une composition en arbre influencée par la Modélisation Déclarative par Décomposition Hiérarchique. Tous les nœuds sont des entités considérées comme des catégories d'objets ayant des propriétés, des relations internes et externes. Le nœud principal de l'arbre est le *Bâtiment* contenant une description abstraite de la construction. L'information générale s'attache à la description du bâtiment, décrivant des détails sur le site et le projet de construction. La racine est le Site qui est sensé être en accord avec la législation du code de la Construction, avec l'orientation naturelle ou autres propriétés fonctionnelles similaires, [Makris 05].

Le nœud *Bâtiment* contient aussi des données générales comme des informations sur le client et le concepteur, des études de faisabilité, des détails de versioning, et cetera. Un *Bâtiment* peut être décomposé en *Etages de Bâtiment* et en systèmes de Toiture. Un *Etage de Bâtiment* rassemble des informations sur les *Espaces*. L'espace comprend le schéma selon les fonctions des espaces : public, privé, technique et de circulation. En général un espace de bâtiment est déterminé comme l'ensemble de toutes les catégories d'espaces ci-dessus. C'est pourquoi les relations

entre ceux-ci et l'entité *Espace* est un assemblage. Dans la Figure 2-17, nous montrons un ensemble d'entités avec leurs relations.

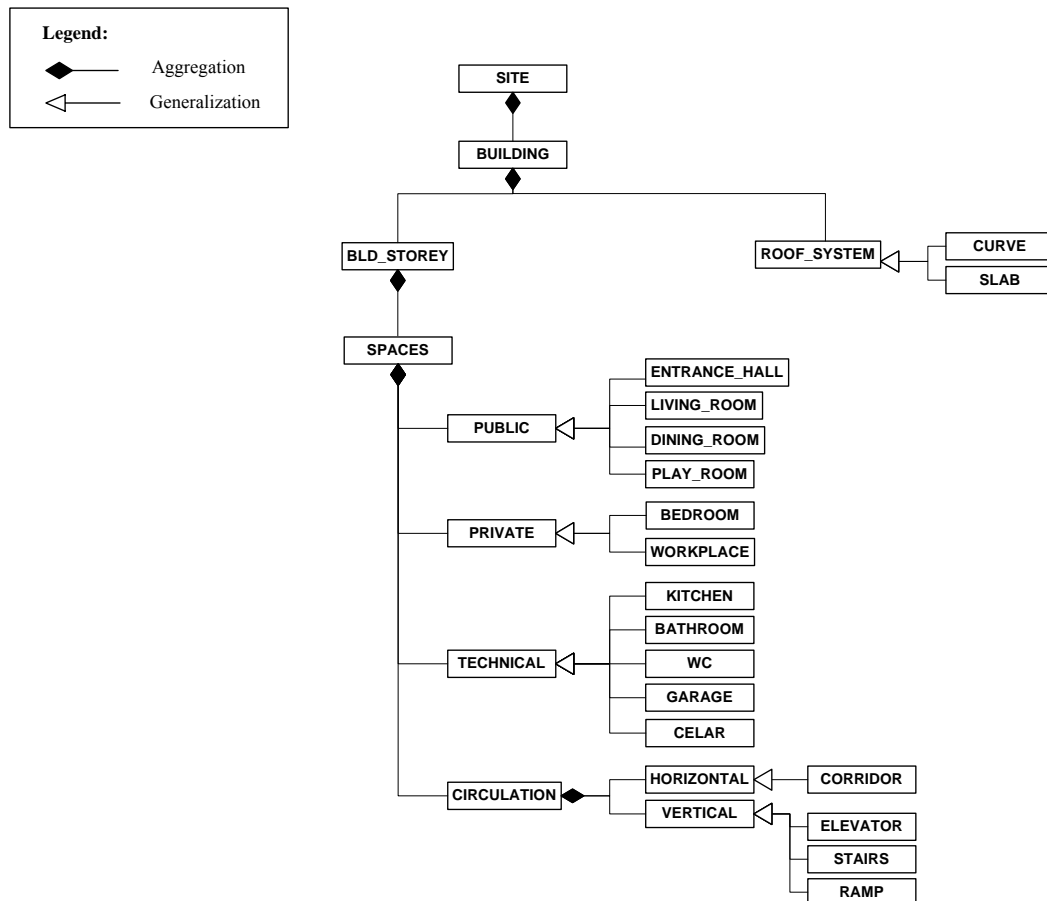


Figure 2-17 DKABM, ensemble d'entités avec leurs relations

De plus le DKABM contient des types spécifiques de relations et de propriétés pour décrire les bâtiments, [Makris 05] :

- Les propriétés descriptives et structurelles, comme le nom de l'entité, sa forme, ses dimensions, et cetera.
- Les informations qui décrivent la fonction de l'entité. Par exemple : une *Cuisine* est utilisée pour préparer et manger la nourriture.
- Les propriétés déclaratives qui sont des unités lexicales associées aux valeurs de l'attribut. Les attributs sont évalués dans un intervalle spécifique de valeurs. Par

exemple : le *Séjour est Grand* signifie que l'attribut *Surface* varie entre 25 m2 et 50 m2.

- Les relations internes qui sont des propositions logiques qui se réfèrent aux propriétés déclaratives. Par exemple : *Le Séjour est plus Long que Large*.
- Les relations de *généralisation* qui définissent les exemples des entités. Par exemple : *Le Séjour est une Pièce publique*.
- Les relations d'*assemblage* entre une entité et les entités qui la composent. Par exemple : *Le Séjour est une partie de l'espace du Bâtiment*.
- Les *associations* (spatiales, topologiques) entre les entités. Par exemple : *Le Séjour est adjacent à la Cuisine de la côte Sud*.

Dans la Figure 2-18, on présente un exemple de description de bâtiment basé sur le cadre DKABM, comme cela est expliqué ci-dessus.

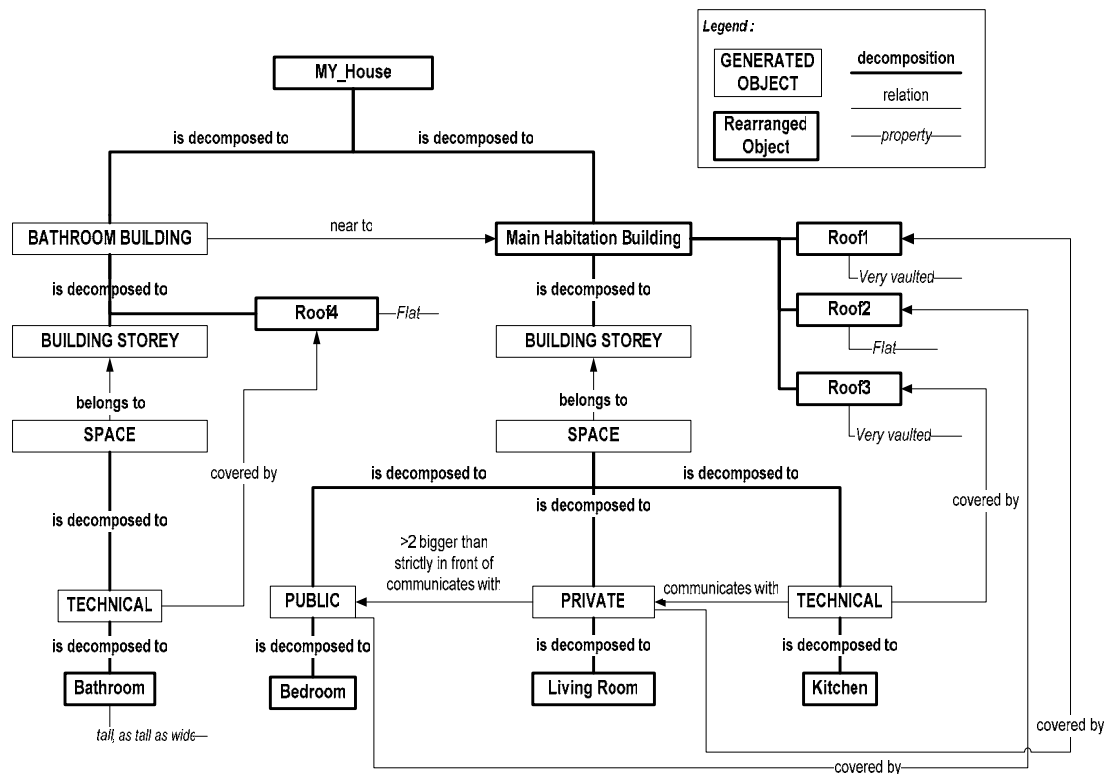


Figure 2-18 Description de bâtiment *MY_House* basé sur le cadre DKABM

2.6. Discussion

Après l'étude et l'évaluation des trois domaines de recherche on a vu qu'il n'existe pas de systèmes de Conception Déclarative Collaborative Assistée par l'Ordinateur dans le domaine du graphisme. Alors, nous allons proposer l'introduction de l'exemple de modélisation déclarative dans l'environnement de la conception collaborative. Le mobile d'une telle proposition est que le système collaboratif déclaratif va faire face à un certain nombre d'inconvénients de la part à la fois des modeleurs géométriques et des modeleurs feature-based à l'intérieur de l'environnement collaboratif. Cependant, il va accroître le potentiel dynamique de la modélisation déclarative. C'est la première fois, à notre connaissance, qu'une telle approche est proposée.

Chapter 3

Research Proposal

3.1. Introduction

In this chapter is presented the framework of a design platform for collaborative design of building during the early phase of their design. The proposed platform is the “*Enhanced – Collaborative Declarative Modelling System*” (Enhanced – CDMS). The Enhanced – CDMS system is proposed and designed in order to provide designers with a working environment for the development of large architectural project(s) in a synchronous or asynchronous way and between geographically distant workplaces.

In the first section of the chapter is presented the Kernel – CDMS. This system is based in collaborative design and declarative design. In particular the system is based on the Kernel-CDMS, which is a web-based framework with client-server architecture, which allows multi-users/designers from different locations to participate in the same design process within a declarative design system (MultiCAD). An architectural project can be designed with the aid of the specific system based on the MultiCAD system. The proposed Kernel – CDMS should respond to a number of particular issues. The specific collaborative environment, through which architectural projects are implemented, should provides the possibility in any designer to work in a personal schedule and to deal with projects in different and distant geographic location from him/her. Additionally, it should also provide the designer with the possibility of criticising the design work of his/her colleague(s), adopt new design practices, or even confronting the emergence of a problem within a team .

In the second section of the chapter we present the Enhanced – CDMS system. Its development should provide benefits that arise from a new structure and functions. In particular the Enhanced-CDMS confronts the two following demands. Firstly, role management of the designers (define working hierarchies and working groups

between the designers. Introduce the concept of “*Role*” within a collaborative environment). Secondly, the management of the projects (define specific sub-parts for a project. Define Working-parts for designers’ working groups).

3.2. Kernel – CDMS framework

In the proposed Kernel – CDMS we need to confront a number of particular interrelated issues [Golfinopolulos et al 04].

- The system should support the management of certain information regarding to each designer/client (Information concerned designer’s connections and profiles data). Such management should control any access rights of the designer (s)/client(s).
- During the collaborative scene description phase the collaborative process needs data sharing. Such process it should control the different status of a scene graph.
- The collaborative system should provide access into both the Knowledge bases of MultiCAD, and to bases necessary for the collaborative process.

In this way the main parts of this system are the following:

- *Design clients.*
- *Collaborative server.*

Figure 3-1 presents a scheme of the proposed Kernel – CDMS. This system could supports multiples equivalent designers (clients) into different phases of the declarative modelling process.

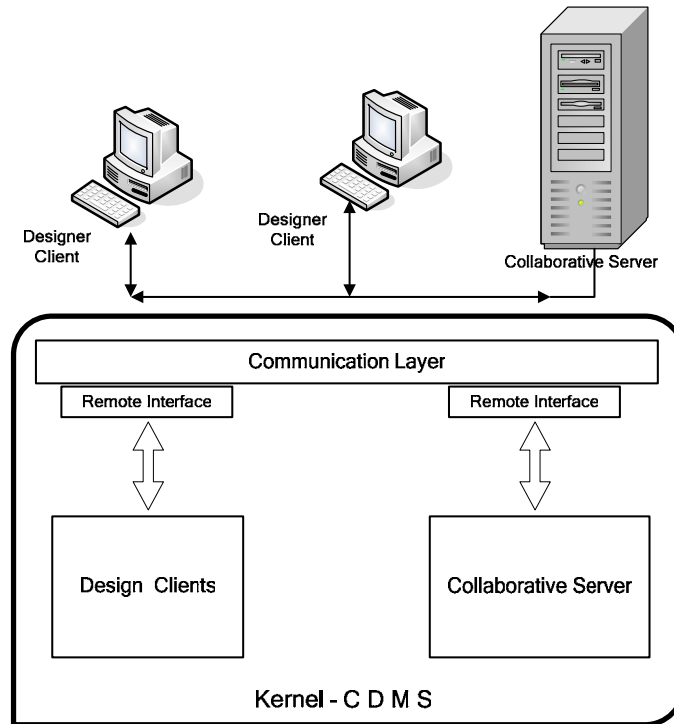


Figure 3-1: Kernel – CDMS scheme

During this process the designers share all relative and appropriate for the design resources (project information, multimedia documents, declarative descriptions, geometric models etc).

3.2.1. Framework analysis

The Kernel – CDMS will be based on a framework which comprises specialised communication and collaborative mechanisms (Figure 3-2). Such mechanisms will be involved with MultiCAD system. The proposed mechanisms are the following [Golfinopolulos et al 04]:

- MultiCAD Designer module.
- Collaborative Declarative module.
- Server facilities module.
- Collaborative base module.

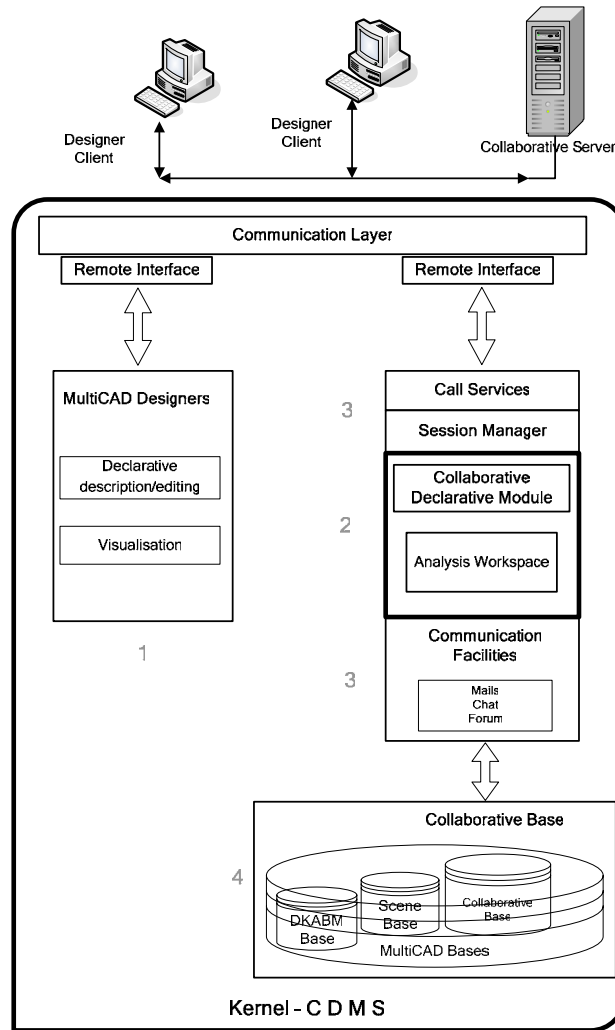


Figure 3-2 Kernel – CDMS framework

3.2.2. Collaborative declarative process

MultiCAD designers

We will apply the collaborative process in the basic declarative conception cycle only during the Scene Description phase (Figure 3-3).

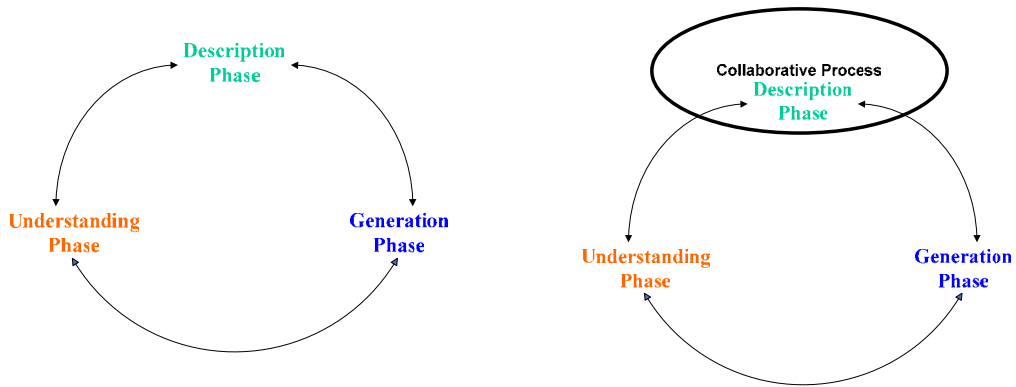


Figure 3-3 Collaborative process in the basic declarative conception cycle

During the scene description phase, designers decompose the project dynamically into sub-parts (Figure 3-4). Next they submit sub-scene description (properties-relations) to the server and finally the system synthesises all description into a complete declarative scene description (Figure 3-4).

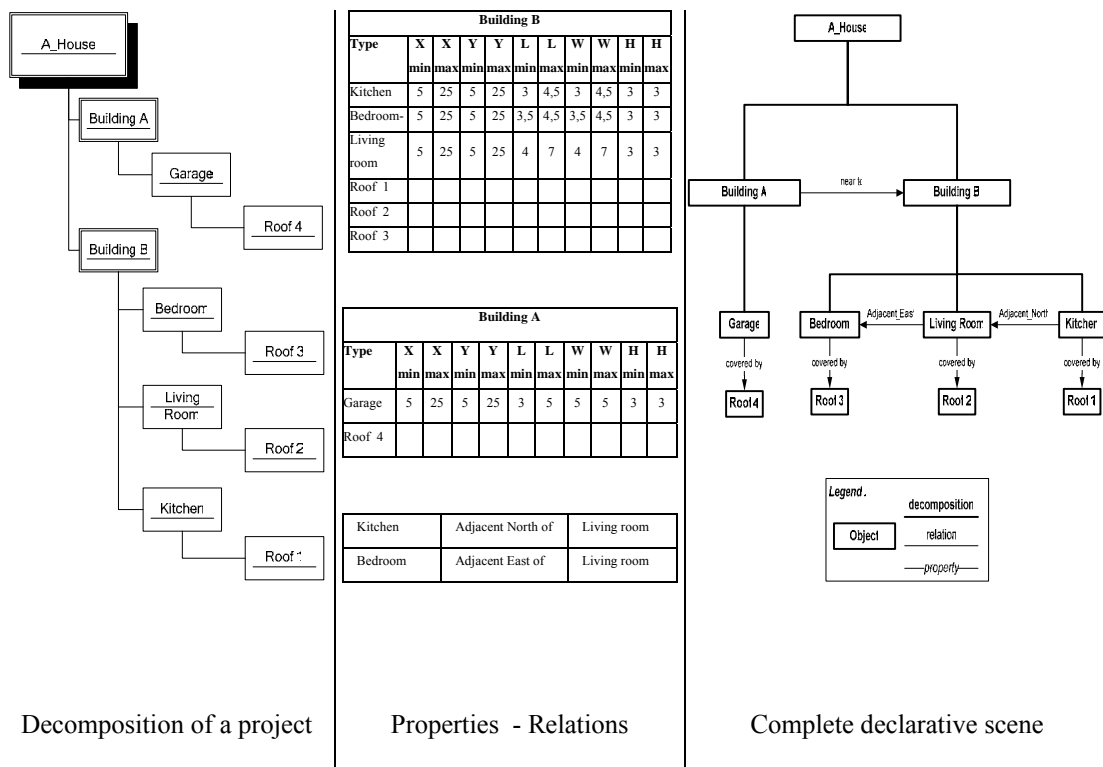


Figure 3-4 Project decomposition

3.2.3. Collaborative Declarative Module (CDM)

The *Collaborative Declarative Module* (CDM) is composed by four interdependent functions (Figure 3-5), [Dragonas et al 05]. The first function concerns the examination of MultiCAD designer's access rights (authorisation – authentication process, et cetera). The second function concerns designers' design actions (manipulations, et cetera). The third function concerns the dynamic generation of new scene graph. This function first collects and then composes all possible designers' manipulations in a final scene graph. The fourth function concerns the submitting of the final scene graph into the appropriate MultiCAD bases.

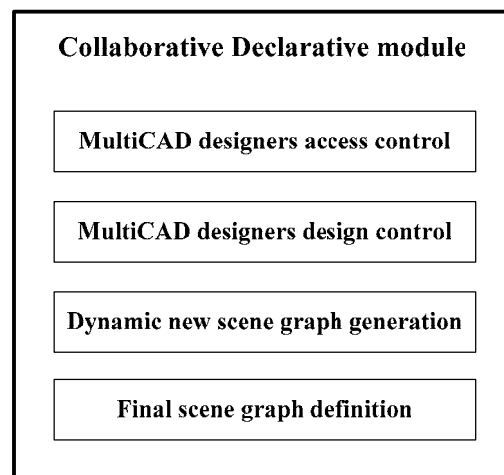


Figure 3-5 *Collaborative Declarative module*

For the needs of the collaborative description of a design project we propose a division of the scene description phase into three new description sub-phases (Figure 3-6):

- *Project Initialisation*
- *Project Assignment*
- *Project Synthesis.*

The above three sub-phases appeared within the second function of the collaborative declarative module.

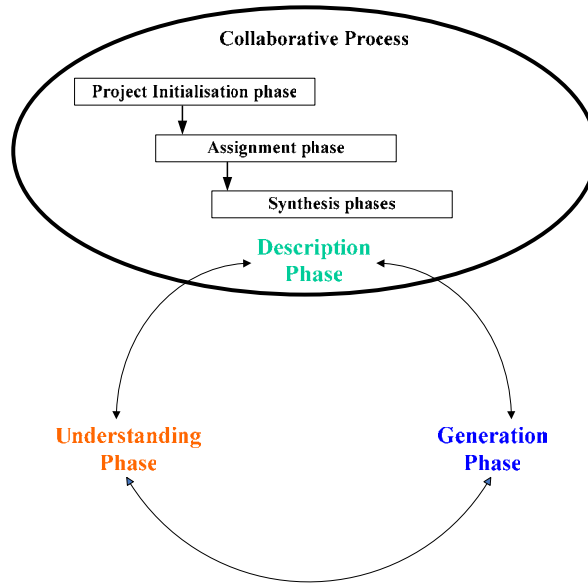


Figure 3-6 Collaborative Description Phases

During the *Project Initialisation* sub-phase, the designer works with the building owner and defines the project by supplying any project information (requirements, specifications, building code, et cetera) Figure 3-4.

The *Assignment* sub-phase gives to designers the opportunity to contribute to the hierarchical decomposition of the scene. The basic assignment cycle is presented in Figure 3-7.

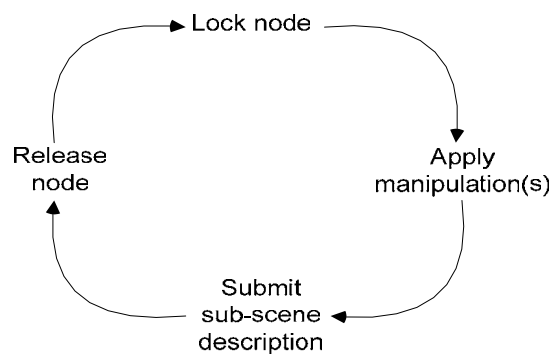


Figure 3-7: The basic cycle of Assignment sub-phase

During the *Synthesis* sub-phase, the final scene description is generated by combining all parts from the participated designers.

The rest two phases of the Declarative cycle remain the same. During the *Generation* phase a set of solutions is produced that is relevant to the final scene declarative description (Chapter 2). Finally, in the scene *Understanding* phase, all produced solutions are visualised (Chapter 2). Afterwards, the final scene description and the valid solutions as well, are stored in a database.

The project model is represented on the collaborative server in the *Analysis Workspace* (Figure 3-2) by a scene graph, which consists of connected nodes (Figure 3-8).

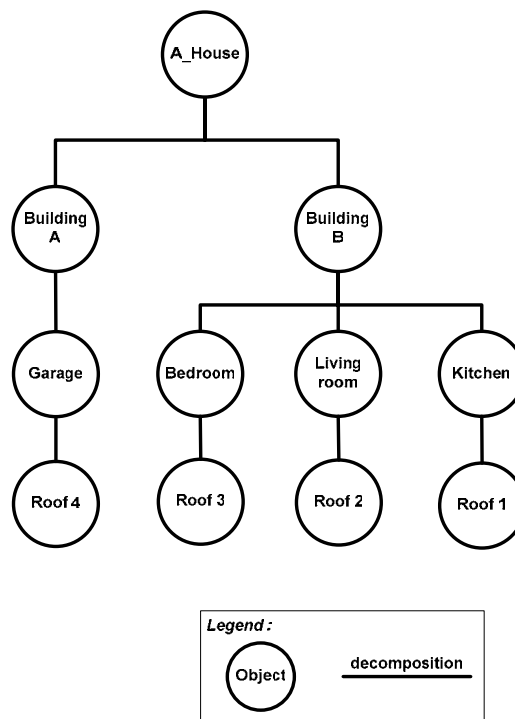


Figure 3-8 Scene graph

For the needs of the collaborative description we define two new status properties for the nodes concerning their manipulation by designers within a scene graph, [Dragonas et al 05].

- *Locked.*

- *Released.*

In principle every node of a scene graph is in *Released* status, i.e. available for selection from the designer(s). When a designer select to work with a specific node then that node alter its status to *Locked*. In Figure 3-9 is presented the scene graph “A_House” when a designer has selected the node “Building A”. In this way the specific node and all its children nodes are *Locked* (grey circle). The remaining nodes retain their initial status that of *Released* (white circle).

Depending on the status of the nodes we define two types of constraints as following, [Dragonas et al 05]:

- *Local constraint.* A constraint that it could be diffused only between all locked nodes (Figure 3-9).
- *Global constraint.* A constraint that it could be diffused between every Released node (Figure 3-9).

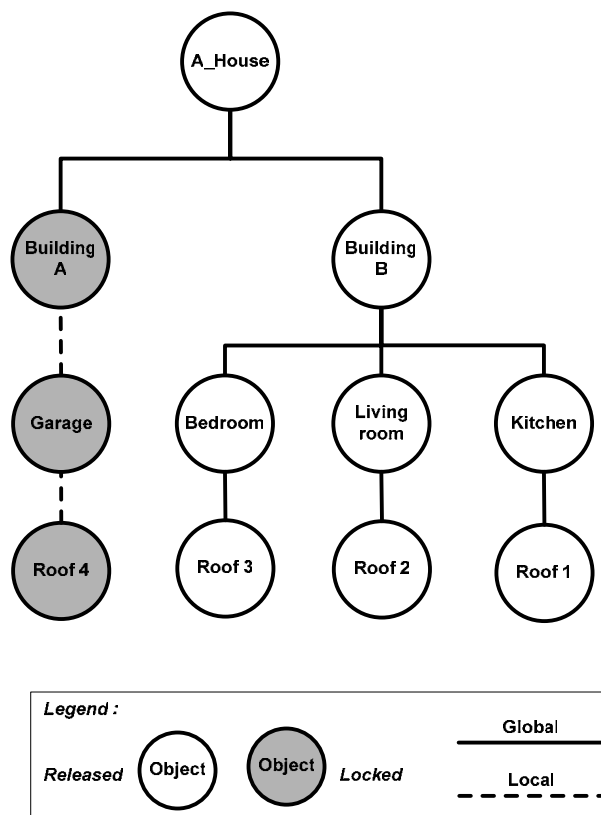


Figure 3-9: Global and Local constraints

The two available operations that a *Designer* can apply to an object (node) of a scene graph are the following:

- “*View*” (Unrestricted operation).
- “*Modify*” (Restricted operation).

The operation “*View*” is provided unrestricted to any authorized designer at any time. The operation “*Modify*” is a restricted operation by the Collaborative mechanism. A designer in order to apply the “*Modify*” operation to an object (node), he/she must “*Register*” himself/herself for it. A restriction mechanism allows or denies this registration dependent on the *State* of this object-(node). Also a registered designer’s actions are limited only to the selected object, and he/she cannot be registered to another object simultaneously. The object can have one of the following states [Dragonas et al 05]:

- *Registered* to a designer.
- *Closed* as a “*child*” node of a tree registered to a designer.
- *Closed* as a “*parent*” node of a scene graph that contains another node registered to a designer.
- *Open*, for possible registration to a designer.

Node state:	Registration:
<i>Registered</i>	✗
<i>Closed</i>	✗
<i>Closed parent</i>	✓
<i>Open</i>	✓
✓: Allowed ✗: Not Allowed	

Table 3-1 Relationship between the state of a node and the Modify operation

However the “*Registration*” to an object (node) is prohibited on certain cases except the case of either an *Open* object (node), or a “*Closed_Parent*” state when the operation is allowed to a designer (Table 3-1).

The general condition of operations that a designer can apply to an object of a scene graph is the following: when a designer registers him/her to an object-(node) he/she defines the state of that object-(node) as “*Registered*”, and additionally the *child* objects (nodes) of the sub-tree that might contain this object they also inherit the “*Closed*” state. For that particular “*Registered*” *father* object all objects that are above it in the same graph they became in the state “*Closed_Parent*” (Figure 3-10, Figure 3-11).

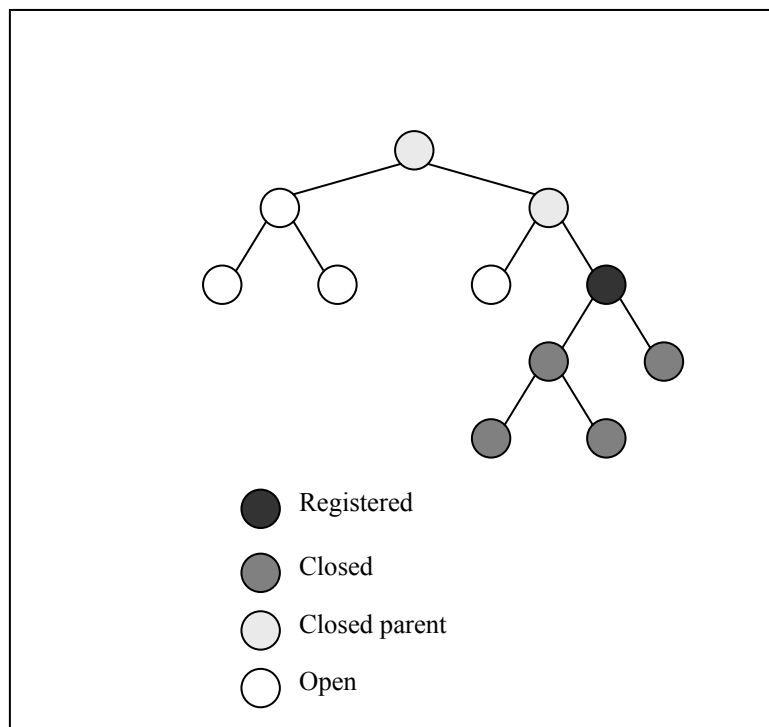


Figure 3-10: A typical situation of a tree with one registered node.

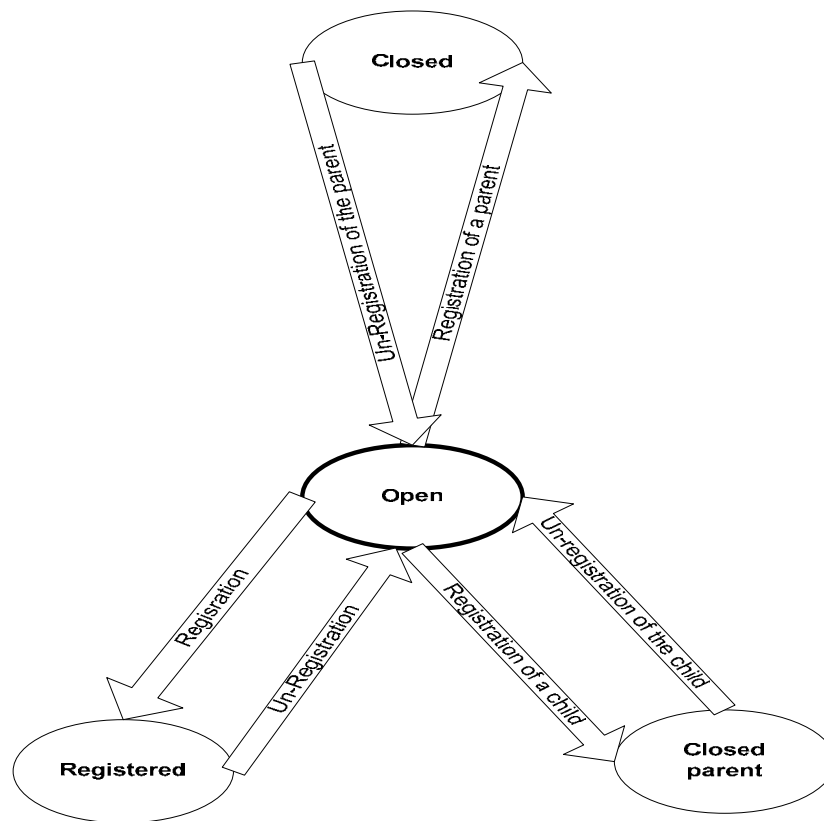


Figure 3-11: Different states of an object

We will define now the “*Modify*” operation as following. When a designer is registered to an object he/she can apply the “*Modify*” operation. The “*Modify*” operation is decomposed in the following sub-operations:

- “*Modify*” object’s Details.
- “*Modify*” object’s Properties.
- “*Modify*” object’s Relations.
- “*Insert*” another object as child to the registered object.
- “*Delete*” one of the child objects of the current registered object.

Each of the above sub-operations is available for the editing of the all objects of the sub-tree of the registered object as well. The only exception is that the sub-operations 4 and 5 can be applied only to registered parent objects. Each sub-operation could have certain restrictions dependent of the state of the object that is

going to be applied to. Specific combinations of operations between / about objects are shown in Table 3-2 and in Table 3-3.

	<i>Registered</i> by another designer	<i>Closed</i> by another designer	<i>Closed</i> as parent by another designer
<i>Modify</i> Details	×	×	×
<i>Modify</i> Properties	×	×	✓
<i>Modify</i> Relations	×	×	✓
<i>Insert</i> another object	×	×	✓
<i>Delete</i> this object	×	×	×
✓: Allowed – ✗: Not Allowed			

Table 3-2 Relationship between the state of a node and the Modify-Insert-Delete operation (for another user)

	<i>Registered</i> to this designer	<i>Closed</i> to this designer
<i>Modify</i> Details	✓	✓
<i>Modify</i> Properties	✓	✓
<i>Modify</i> Relations	✓	✓
<i>Insert</i> another object	✓	×
<i>Delete</i> From this object	✓	×
✓: Allowed - ✗: Not Allowed		

Table 3-3 Relationship between the state of a node and the Modify-Insert-Delete operation (for same user)

In particular the “*Modify Properties*” sub-operation edits the objects properties, including the following operations:

- “*Add*” a new property,
- “*Delete*” a property,

- “Update” an existing property.

The “*Modify Relations*” sub-operation edits the objects relations including the following operations:

- “Add” a new relation,
- “Delete” a relation,
- “Update” an existent relation.

For a registered designer the state of an object (node) at any time affects how its relations with other objects (nodes) can be considered.

As a matter of fact, the relations between a registered *parent* object (node) and its *child* object(s) within its sub-graph are defined as “*Local*”, and the relations with all other objects of the scene graph are defined as “*Global*”.

In this way the designer can utilise the sub-operation “*Modify*” to a relation only if this relation is “*Local*”. In Figure 3-12 we present an example of the relations of a registered father-object, and its sub-tree with its child-objects.

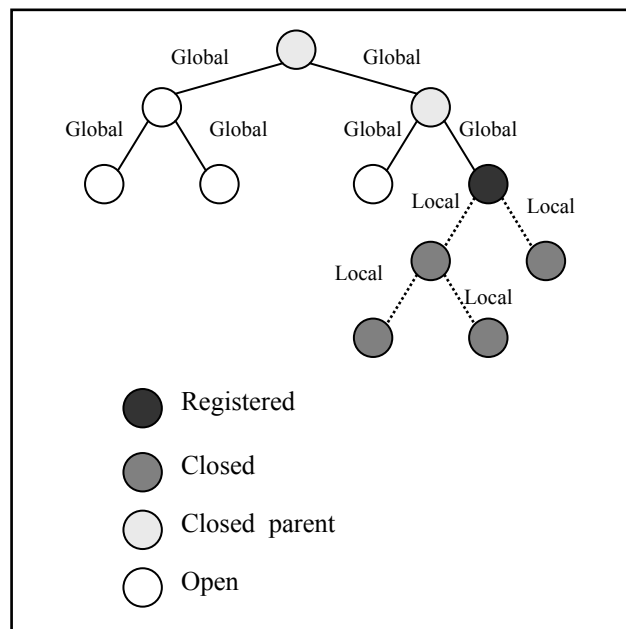


Figure 3-12 Relations for a registered object.

3.2.4. Server facilities

The server facilities provide support to the collaborative Kernel – CDMS in the following way, [Dragonas et al 05]. First they offer a group of Communication services such as mail, chat, and e-forum. Second, the server offers a Session manager, which controls and manipulates every session between the designers and the Kernel – CDMS, (Figure 3-13).

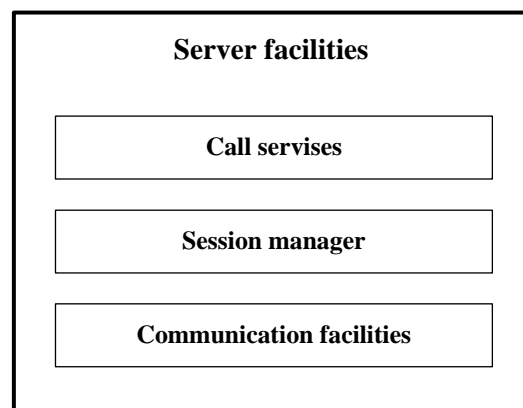


Figure 3-13 Server facilities

3.2.5. Collaborative Base

The collaborative database module acts on two interdependent levels. First it enables the definition of a Collaborative base within the MultiCAD bases repository, [Dragonas et al 05]. Second it provides a set of specific functions in order to connect the MultiCAD designers with the DKABM base, the Scene base the Knowledge base and finally with the Collaborative base (Figure 3-14).

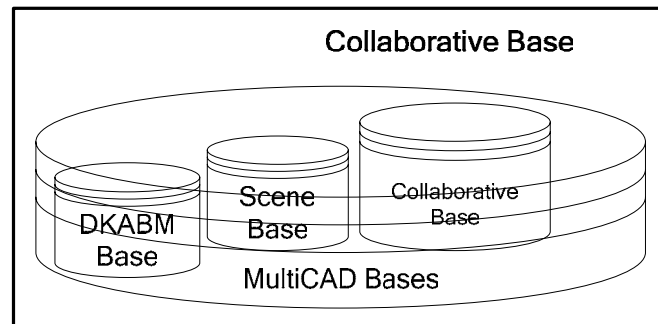


Figure 3-14 Collaborative base

3.2.6. Flowchart

We provide a flowchart in order to show the general flow of the developed methodology. The main group of functions for the Collaborative declarative design are the following:

- *Users' management.* These functions concern the management of the designers which are involved in the design process in general. The functions are *Insert – Delete – Update*, (within the red area in Figure 3-15).
- *Project management.* These functions concern the management of the collaborative design projects. In particular the administrator could either *Create* a new project and/or *Delete* an existent one, (within the green area in Figure 3-15).
- *Project Editing.* These set of functions include all actions that are available for the edit and /or modification of a project description by a designer. In particular the designer could *Modify* and/or *Update* an existent project, (within the blue area in Figure 3-15).

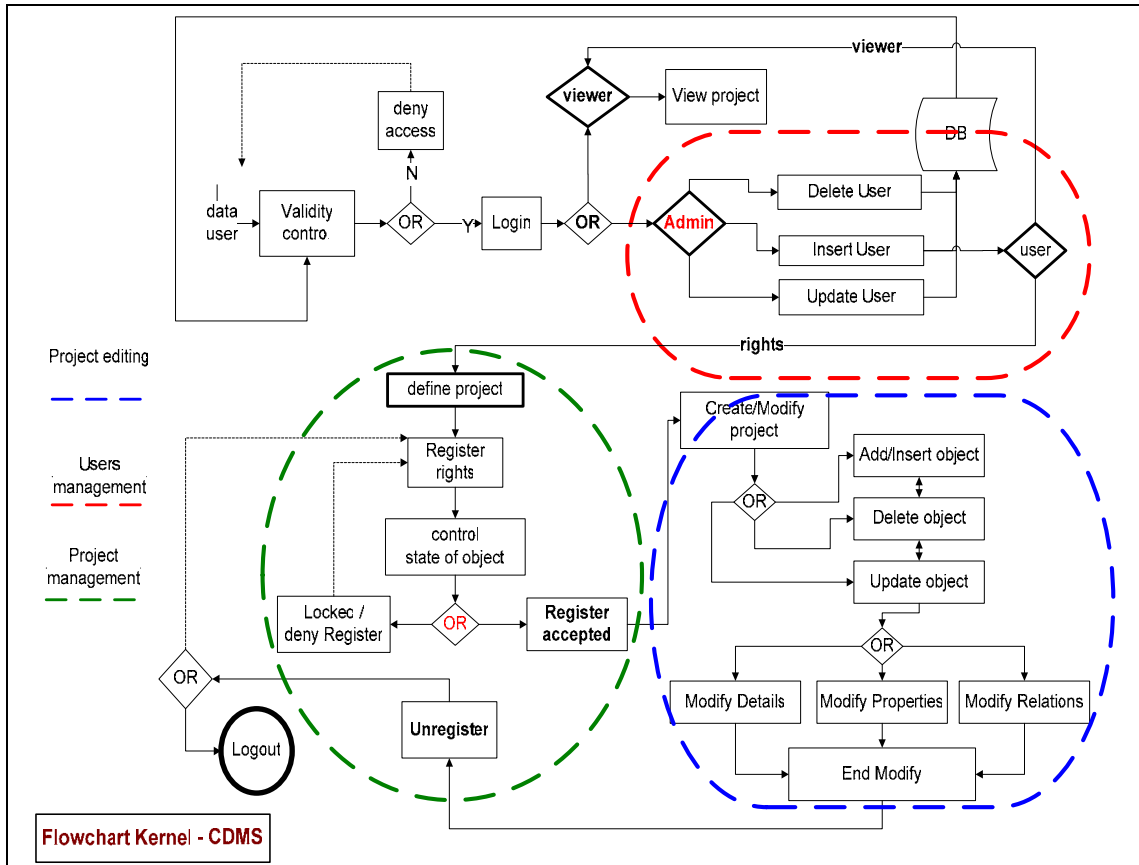


Figure 3-15 Flowchart of Kernel – CDMS

3.3. Enhanced-CDMS

The development of the *Enhanced-CDMS* could provide additional benefits, which they arise from a new structure and functions. In particular the *Enhanced-CDMS* confronts the two following demands. Firstly, role management of the designers (define working hierarchies and working groups between the designers. Introduce the concept of “*Role*” within a collaborative environment). Secondly, the management of the projects (define specific sub-parts for a project. Define Working-parts for designers’ working groups), [Dragonas et al 06].

In this section we present a set of collaborative utilities for project workflow management and design. Such utilities are extensions on the WWW-based system prototype for collaborative design within MultiCAD declarative modelling system. The Kernel – CDMS is the core of the new system. The primary objectives of the current development are the following, [Dragonas et al 06]:

- To enhance the core system (kernel) with administration utilities, for the distributed collaborative environment, and also
- To provide administration support to collaborative declarative design workflow process.

In order to achieve the above objectives the *Enhanced-CDMS* will be based on two new specialised parts, administration part and event recorder part (Figure 3-16). The two parts in the proposed system are the following, [Dragonas et al 06]:

- CDMS Collaborative Administration – CA
- CDMS kernel enriched with Collaborative Event Recorder – CER
- In particular the part *Collaborative Administration* encircles the existing CDMS kernel. In this way the *Collaborative Event Recorder* encircles the *Collaborative Administration* (Figure 3-16).

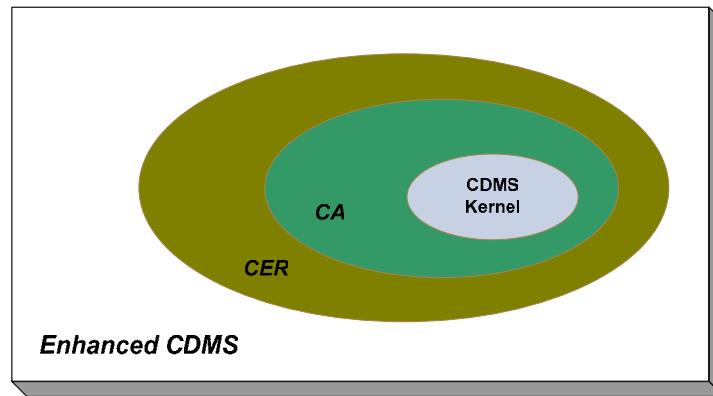


Figure 3-16 *Enhanced-CDMS* scheme

3.3.1. Framework analysis

The *Enhanced-CDMS* will be based on a framework of specialised administration and event recorder mechanisms. Such mechanisms will be involved with MultiCAD system. The proposed mechanisms are the following, [Dragonas et al 06]:

- Collaborative Administration Mechanism – CAM,
- Collaborative Event Recorder Mechanism – CERM.

The logic of multiple administrators (MAD) is for the first time introduced in a collaborative distributed environment. In the *Enhanced-CDMS* we encapsulate that logic. In order to maintain a restricted and controlled access to the administrative information, we develop a collaborative administration mechanism consisting of the two following units (Figure 3-17):

- Kernel Administration Unit (KAU),
- Workflow Administration Unit (WAU).

The functionality of the *Collaborative Event Recorder Mechanism* concerns the following functions:

- “*Record*” all the events that occur from the Designers of the Kernel-CDMS,
- “*Categorize*”, the events

- “Save” events to the “*Collaborative Base*”, in order for this information to be available to other parts of the system.

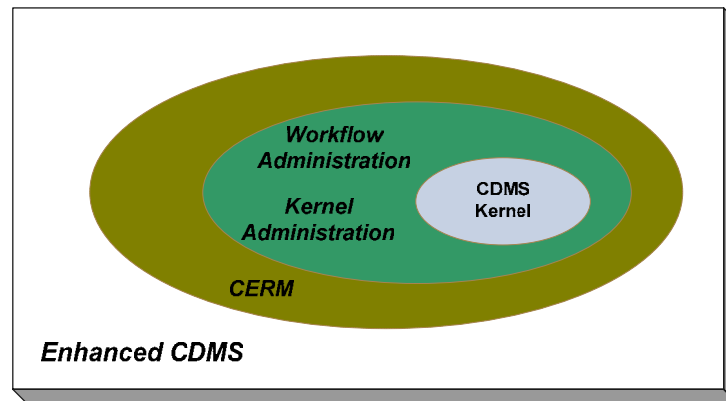


Figure 3-17 *Enhanced-CDMS* framework

3.3.1.1. KAU Entities' General Description

The *Enhanced-CDMS* is developed on top of the Kernel-CDMS. In order to function properly we enrich the basic entities and functions of the *Kernel Administration Unit*. For the scopes of this thesis we define four new basic entities. The entities that used finally within the *Enhanced-CDMS* are the following:

- *Designer*,
- *Team*,
- *Project*,
- *Conceptual Scene graph*.

We should underline that the entities “*Designer*” and “*Conceptual Scene graph*” are the same entities as used by the Kernel-CDMS, which are enriched with additional attributes as required by the functions of the *Enhanced-CDMS*.

In the same manner the entities “*Designer*”, “*Team*”, and “*Project*” include an attribute which defines each one of the following entities: *Designer_Log*, *Team_Log*,

and *Project_Log* respectively. In particular, these entities are the structural entities of the *Collaborative Event Recorder Mechanism*.

Designer. This entity includes all the possible designers and maintains the following attribute (status) which specifies the role *Designer* as one of:

Viewer, (a *Designer* that has only the right to view the *Conceptual Scene graphs* of the current *Projects*)

Team Member, (a *Designer* that belongs to a *Team*, at least one, and he/she has rights to alter the *Conceptual Scene graph* of the *Project* of his/her *Team* only)

Team Leader, (a *Designer* that belongs to a *Team*, having additional rights to that of a *Team Member*, such as:

Add/Edit/Delete Designer to his/her *Team*,

Project Initiation,

Creation of the *Conceptual Scene Graph*,

Set Conceptual Scene graph to *Designer*,

View Team log, Project log)

Administrator (a *Designer* that has access to *Administrator Mechanism Units* only).

Team. It maintains general information such as name, date of creation, et cetera. Which concern *Team Members*, and *Team Leader*. Each *Team* operates each time to a single *Project* only. The *Team Leader* is responsible for the administration of a *Project*.

Project. It maintains general information such as name, conceptual scene graph identity, et cetera. A *Project* is associated to one *Team* only.

Conceptual Scene graphs. It is the decomposition of the scene which it is based to a DKABM template (Chapter 2). This template contains all objects of that scene, while in this graph a node represents every object and any decomposition relation between objects is represented by a *connection*.

3.3.1.2. WAU Entities' General Description

We define the WAU by specific functions which provide *Team Leaders* with the appropriate tools in order to manage the life cycle of the design process of a *Conceptual Scene Graph* by a *Team Member*. However these tools could manage any events that occur during this process. The life cycle of a design process could be expressed as following (Figure 22):

- *Team Leader* set the basic characteristics of the *Project* (name, location, etc.)
- *Team Leader* selects the *Conceptual Scene graph* structure, of the *Project*, from the available blueprints (from DKABM).
- *Team Leader* sets the current state of the *Project* as the basis version 0.
- *Team Leader* selects the *Team Members* from the available *Users* of the CDMS.
- *Team Leader* sets rights for each *Team Member* on the Conceptual Tree of the *Project*.
- *Team Leader* initiates the “*Development*” phase and sets the maximum time interval for it.
- Each *Team Member* copies a clone from the last stable version of the project, and works on it.
- Before or by the end of the allowed time interval, *Team Members* post their proposals to the *Team Leader*.
- *Team Leader* decides, approves and disapproves some of the proposals.
- The approved proposals are integrated, by the *Team Leader*, to a new stable version of the *Project*.
- If the *Project* version is satisfactory then it is the final outcome of the collaborative design. In the case where the version is not adequate then the life cycle is repeated from the step 6, onwards.

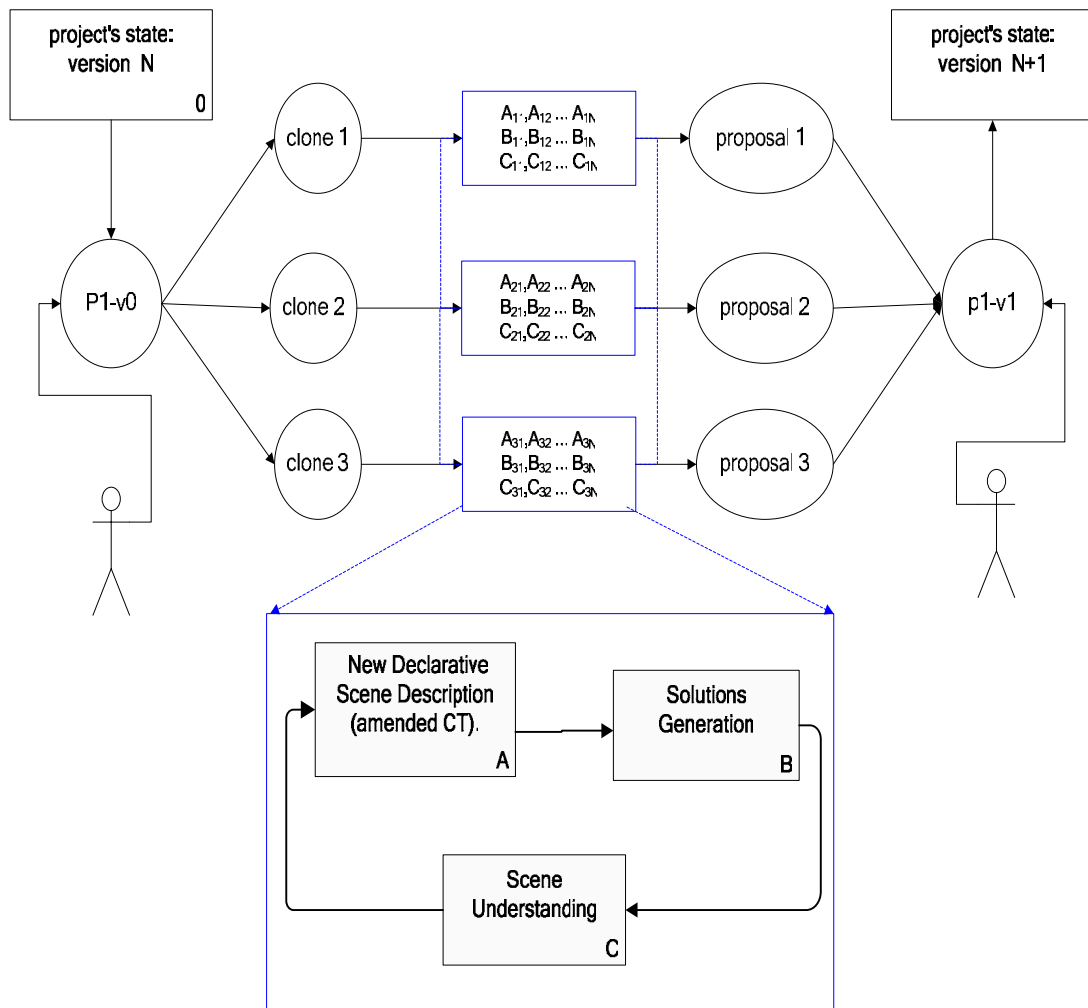


Figure 3-18 Development Process

The step 7 represents the individual design process of a *Team Member* on the *Project*, in order to create a proposal. This is also the declarative conception cycle of three phases and can be analysed as following:

- New Declarative Scene Description (amended Conceptual Tree).
- Solutions Generation.
- Scene Understanding.

During this design process a *Team Member* can submit his *Description* (amended Conceptual Tree) to the Enhanced-CDMS in order to visualise the outcome of his/her work. Then he/she can consider the outcome and either revisit the cycle, or supply the amended Conceptual Tree as his/her proposal. The development process of

a project can be seen in Figure 3-18. In this way within the WAU the main functions that a Team Leader can utilise for this purpose are the following:

- “*Initiation*” of the project (the TL sets the basic properties of a project).
- “*Selection*” of the Conceptual Tree type from a number of available types.
- “*Versioning*” of a project (the TL sets a current project as the latest stable version, in which any further developments could be applied).
- “*Add*” a Viewer as Team Member.
- “*Delete*” a Team Member from the Team.
- “*Set*” the access rights of a Team Member to specific sub Conceptual Tree.
- “*Reject*” a design proposal from a Team Member.
- “*Set*” the next due date of the current proposals.

3.3.1.3. Flowchart

We provide a flowchart of the Enhanced-CDMS system in order to show the general flow of its methodology. The main group of functions for the collaborative declarative design process are the following:

Users’ roles management. These functions concern the management of the roles that involved designers could have. The administrator ascribes particular roles for the designers according a hierarchy with the appropriate access rights within a design process, (within the red area in Figure 3-19). The roles are *Team Leader*, and *Team Member*.

Project workflow management. These functions concern the management of the workflow by the Team Leader of a collaborative design project. In particular the Team Leader could ascribe part of the design sessions to each of the Team Member, and in the end the Team Leader defines the final design synthesis as a result of the Team Member(s) proposals, (within the blue area in Figure 3-19)

Finally the functions about the project editing by the *Team Member(s)* remain in general the same as defined by the Kernel-CDMS. These set of functions include all actions that are available for the edit and /or modification of a project description

by a designer. In particular the designer could *Modify* and/or *Update* an existent project, (within the blue area in Figure 3-19).

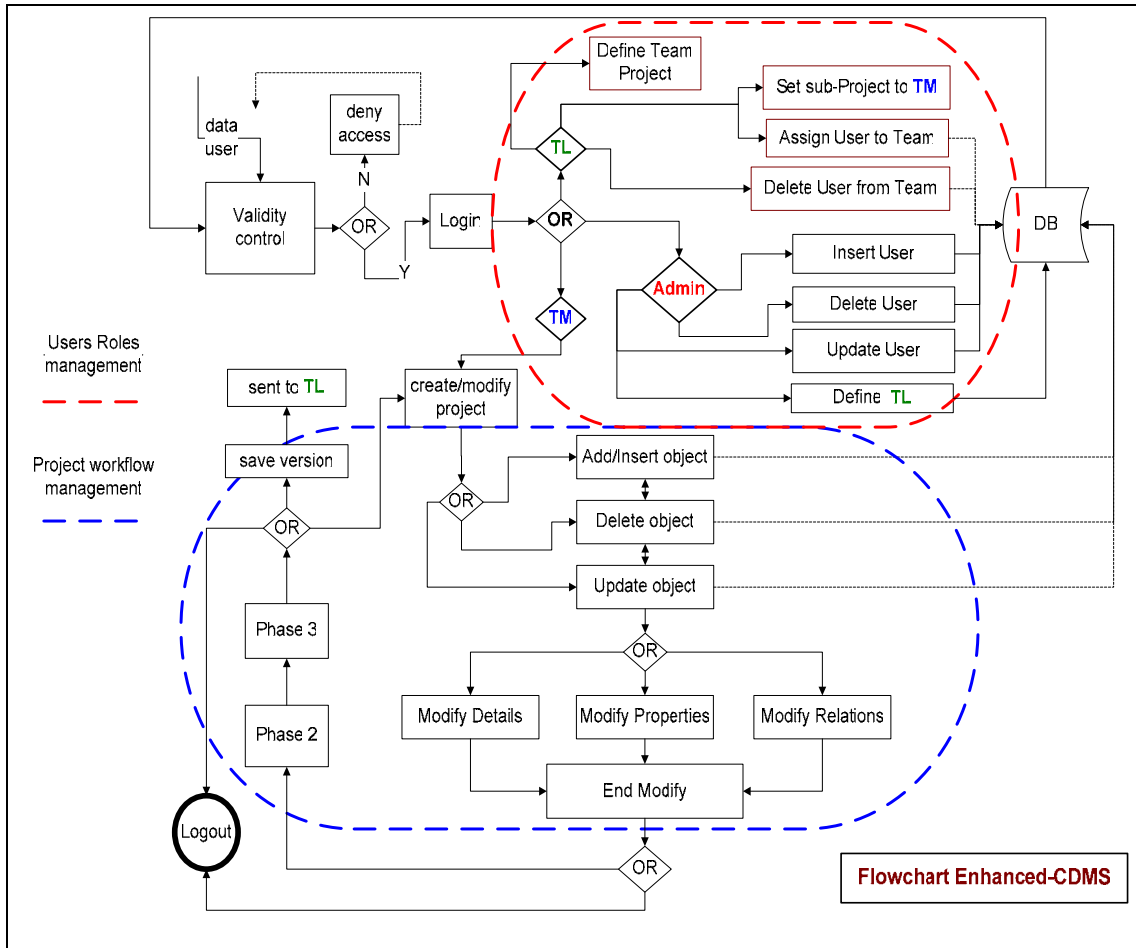


Figure 3-19 Enhanced-CDMS flowchart

Chapter 4

Prototype system implementation

4.1. Introduction

In this chapter we will present the implemented Enhanced-CDMS system components. We will interfere with specific sections of the system framework of MultiCAD as it was presented in Chapter 2. The current chapter is divided in three sections. In the first section we present the implementation of the adaptation of the DKABM within our system. In the second section we present the implementation of the Kernel-CDMS within the MultiCAD system architecture with the use PHP (Personal Home Page – Hypertext Preprocessor) language and database management software (MySQL). The Enhanced-CDMS –CDMS is analysed and moreover the collaborative mechanism that it implements is made clear. In the third section of the chapter we present the implementation of the Enhanced-CDMS system. That system concerns the project workflow management.

4.2. Kernel-CDMS application software

In order to implement the Kernel-CDMS system we made some basic assumptions on issues that concern the operation system, the network platform, and the hardware requirements.

The Kernel-CDMS framework operates in the HTTP (HyperText Transfer Protocol) environment as the communication layer. The Server facilities (services calls, session managers etc) are assured by the operating system. The Collaborative Base and the knowledge resources are included in the MultiCAD Database and are concretized for the needs of this implementation under a set of related tables.

The functionality of the Kernel-CDMS system is based on the overlapping of the database by PHP pages that implement the logic structure and the algorithmic process of the Collaborative declarative mechanism. These pages built a graphic user

interface (GUI) with specific abilities and restrictions. The designers' actions are restricted to actions that are allowed to him/her by the pages at his/her disposal, and he/she cannot act spontaneously. Information concerning the designers' connections and designers' profiles data are all stored in the Collaborative base so as to be controlled by the administrator of the system. The system development is based on open source software technology. The system is very flexible and so very easily transferable among different operation systems, without the need for further conversion or addition, but only these concerning its visual appearance. The demands from the part of the designers are the minimum and the only need for an unobstructed functionality of the designers is a current version of a web browser.

4.2.1. Operating System

In the Kernel-CDMS the operating system is responsible for maintaining the services at both the Client and the Server part for communication and display reasons. In particular, the operating system takes the responsibility to transfer the data from the HTTP protocol to the web browser which operates under it. Also the operating system is responsible to send the proper data requests from the Client part to the Server part through the HTTP protocol. The system as a genuine client-server system needs the operating system on the Client part only to maintain the proper communication between the browser and the HTTP protocol, for display reasons, and the operating system to the server side to maintain the ordinal operation of the system software and the collection of the designers' feedback. In general, the operating system's services are the basic web services that every contemporary operating system provides. Due to the fact that the software is based on open source software the application is cross-based and it is independent from any particular operating system. The system has been installed and tested at both Windows and Unix-based operating systems. Finally the operating system maintains the operation of the database management system (MySQL in the implemented system) which can be installed in the same server as the main server or in a database dedicated server.

4.2.2. Network platform – Hardware requirements

The network platform does not depend to a particular platform, and the system can operate as easily as any other web-based system. The only consideration that

really needs to take into account is the demands of operation. The system has been tested in laboratory situations and not in real life situations. The number of simultaneously users of the system, the amount of data transferred to and from the server for each user, as long as the geographical distribution of the users is important features of the network. However hardware requirements must be thoroughly examined in order to take decisions about installation of the system. The system is very dynamic and can be reformed, even if it is already operational, at any time by the system's administrators, as long as there is a particular well designed planning and the above requirements has been considered.

4.2.3. Kernel-CDMS implementation framework

The functionality of the Kernel-CDMS system is based on the overlapping of the following four components (Figure 4-1):

- First the MultiCAD Designer module, (1),
- Second the Server facilities module, (2).
- Third the Collaborative Declarative module (CDM) through its algorithmic process, (3).
- Fourth the Collaborative Base module, which it is defined by PHP pages that implement its logic structure, (4)

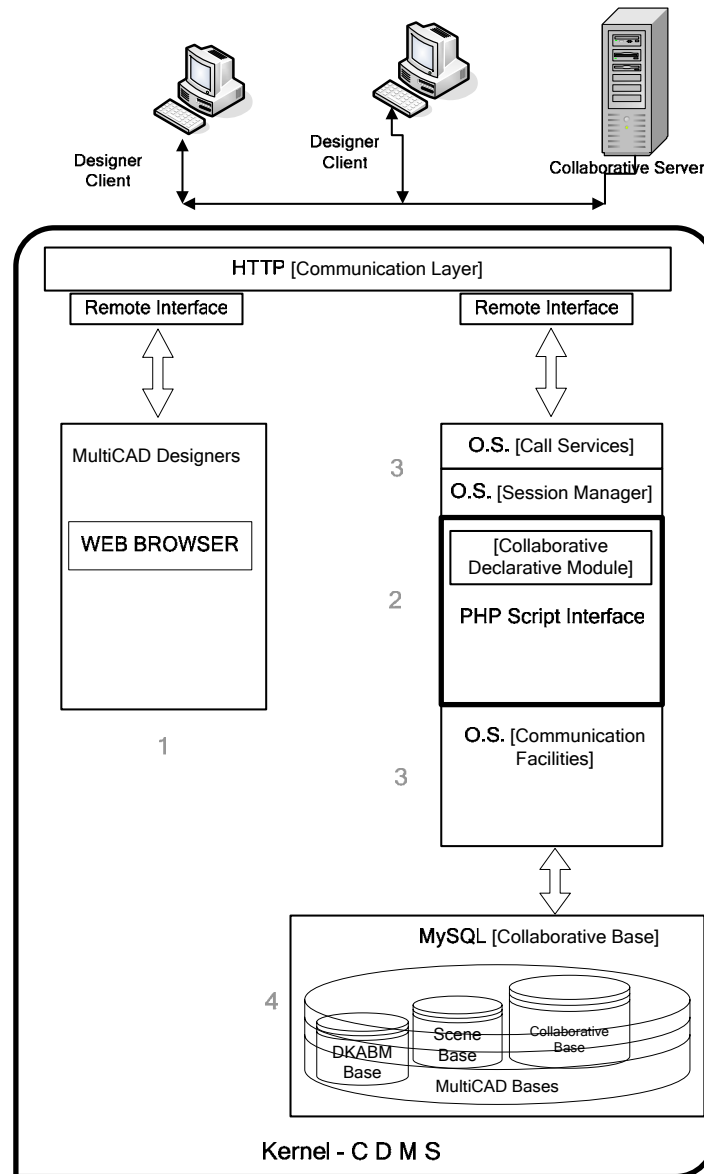


Figure 4-1 Kernel-CDMS system framework

The PHP pages built a graphic user interface (GUI) with specific abilities and restrictions. The designers' actions are restricted to actions that are allowed to him/her by the pages at his/her disposal, and he/she cannot act spontaneously. Information concerned the designers' connections and designers' profiles data are all stored in the Collaborative database so as to be controlled by the administrator of the system.

The implementation of the Kernel-CDMS will be based on the MultiCAD multi-layered architecture.

Figure 4-2 presents a system framework of the Collaborative declarative design within the MultiCAD environment.

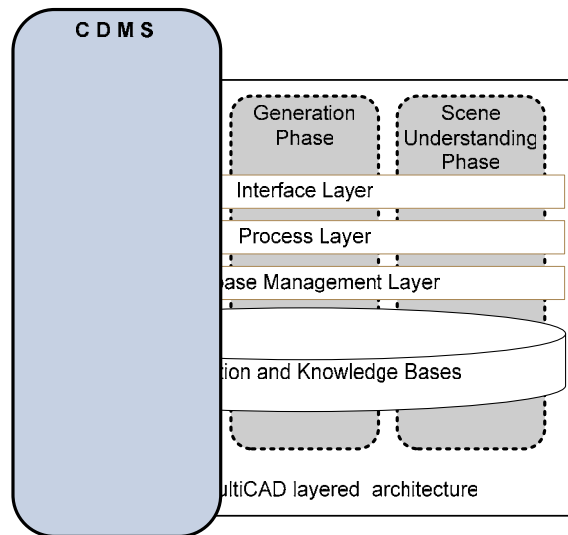


Figure 4-2 CDMS within the MultiCAD environment

4.2.4. Kernel-CDMS modules

In this subsection we will present in details the implementation of the basic modules of the Kernel-CDMS. First the Collaborative Declarative module (CDM), second the Collaborative Base module (CBM), and third the MultiCAD Designer module (MDM).

4.2.4.1. Collaborative Declarative module

The Collaborative Declarative module consists of three layers of restrictions. These layers support the management of the Collaborative database. At any time, any designer who tries to access the collaborative database is forced to encounter all these layers and be checked against each one of them independently. In the case of a successful pass, through each one of them, the designer is allowed to have access to a declarative scene description. These layers are the following (Figure 4-3):

- *Authorisation layer.*

- Collaborative mechanism layer.
- Collaborative base layer.

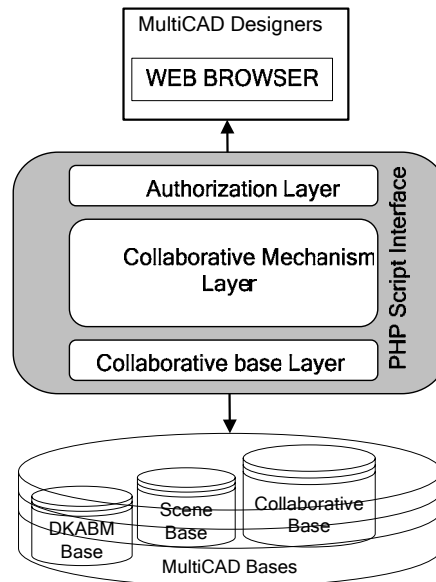


Figure 4-3 Restriction layers

4.2.4.2. Autorisation Layer

The Authorisation Layer incorporates an Authorisation process which enables the connection of a designer with the Collaborative mechanism layer (Figure 4-4). The Authorisation layer manages certain information regarding to each designer, such as: name, location, access rights, username, password et cetera. This information is contained in the Collaborative base. The implementation of the Authorisation Layer is based on certain PHP pages. The Layer enables either the authentication process of a possible designer or the rejection of a non-valid one, by comparing the username-password input by the designers, and the relative information that the collaborative database retains.

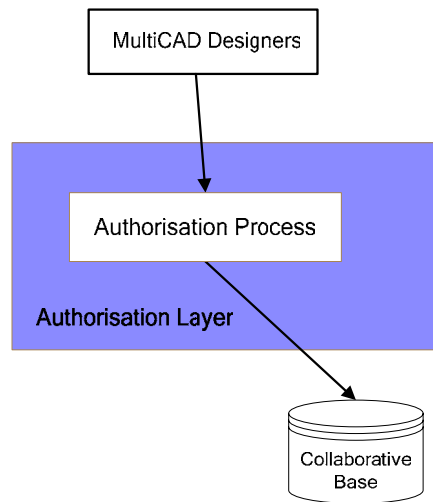


Figure 4-4 Authorisation Layer

All designers' information is expressed in a specific table (Table 4-1). This table is retained in the Collaborative Base, and it has the following schema:

id	int(6)
login_name	varchar(8)
password	encrypted
name	varchar(30)
surname	varchar(30)
location	varchar(50)
last_login	varchar (30)
access	varchar(10)
PRIMARY KEY	('Id')

Table 4-1 *Designer* information table

In the case of a successful authentication of the designer, his/her data (username-password-access) are passed to the next layer (Collaborative Mechanism Layer).Based on the 'access' field value of the above table, certain function buttons are activated and certain rights to projects are loaded. Each successful login is recorded for administration purposes in a table (Table 4-2) with the following schema:

id	bigint(20)
login-datetime	varchar(40)
logout-datetime	varchar(40)
user	varchar(8)
ip	varchar(15)
PRIMARY KEY	(Id)

Table 4-2 *Login* table

4.2.4.3. Collaborative Mechanism Layer

The Collaborative Mechanism layer concerns all possible designer activities during collaborative declarative design. This layer enables the connection between MultiCAD designers and all appropriate information from and within the Collaborative Base (Figure 4-5).

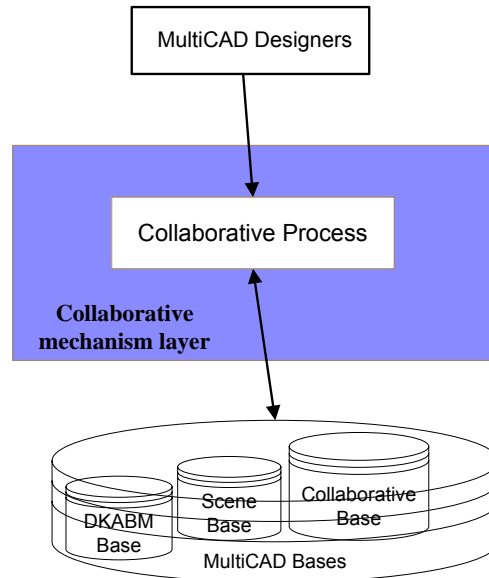


Figure 4-5 Collaborative Mechanism Layer

The Collaborative Base retains the architectural projects as scenes, and every scene description (blueprint) is retained as a DKABM template. Each DKABM is saved as a table. We define the table's schema with the form of Table 4-3:

record_id	bigint(20)
father_id	smallint(6)
child_id	smallint(6)
PRIMARY KEY	(`record_id`)

Table 4-3 *DKABM* table

For every node (object) there exist as many records as its children (object). This table can be constructed with the use of a specific function button, which is available to a designer only. In this way the Kernel-CDMS environment can contain

multiple DKABMs templates. However, a project is related only to one DKABM template.

Another table is defined in order to retain the *actual* objects that take part in a scene at any time. The table has the schema as presented in Table 4-4 .

Object_ID	int(11)
Obj_Name	varchar(50)
Obj_Descr	varchar(50)
Object_Type_ID	smallint(6)
Obj_Type_House	varchar(50)
PRIMARY KEY	(`Object_ID`)

Table 4-4 *Actual* objects

Given the fact that every node (object) is related with a father node (object) and/or a child node (object) the two above tables have the following property. The '*Object_Type_ID*' record is related to the '*father_id*' and '*child_id*' records of the DKABM table.

An authenticated designer can load up a page where he/she can view current project(s). The following algorithm shows the decomposition of the scene to a scene graph based to a DKABM. This Conceptual scene graph tree contains all objects and relations of that scene. A node represents every object and any decomposition relation to one or more children is represented by a connection. This algorithm is the graphic user interface (GUI) that the designer uses in order to select a scene object to operate on. The Collaborative mechanism enables the automatic construction of the scene graph with the use of a specific recursive function (Figure 4-6).


```

function      makethetree      ($node_pos,      $objects_array)      {
$children_array = read_the_children($node_pos);

    if ($children_array == 0) {

        add_terminal_node($node_pos);

        return;

    } else {

        foreach ($children_array) {

            $father = $node_pos;

            $child = next_child($children_array);

            add_a_father($father,$child);

            makethetree ($child, $objects_array);

        }

    }

}

```

Figure 4-6: *Conceptual Scene graph tree* construction algorithm

For the proper operation of the above function it is necessary to have an array containing all already created objects. The array is given as the second parameter `$objects_array`. This array contains the objects of the scene graph. The above function creates the conceptual scene graph by adding a node for each *father* object and then executes itself for each of the *children* of that object in order to complete the conceptual scene graph. This repetitive cycle ends when a *father* object is found without any *children* objects. The above function is based also on different functions which they create the visual entity of the scene graph. The above function is implemented within the *Server* part of the Kernel-CDMS and it is based on the PHP script interface.

The two available operations that a designer can apply to an object of a scene graph are “*View*” (Unrestricted operation), and “*Modify*” (Restricted operation). The operation “*View*” is provided unrestricted to any authorized designer at any time. The operation “*Modify*” is a restricted operation by the Collaborative mechanism. A designer in order to apply the “*Modify*” operation to an object (node), he/she must “*Register*” himself/herself for it. A restriction mechanism allows or denies this registration dependent on the *State* of this object-(node). Also a registered designer’s actions are limited only to the selected object, and he/she cannot be registered to another object simultaneously. The object can have one of the following states [Dragonas et al 05]: “*Registered*” to a designer. “*Closed*” as a “*child*” node of a tree registered to a designer. “*Closed*” as a “*parent*” node of a scene graph that contains another node registered to a designer. “*Open*”, for possible registration to a designer. In order for the above designer activities and object states we define two different tables containing this above information within the Collaborative Base. First, the table ‘*Analysis*’ contains information about which designer is using which object, and it has the following schema (Table 4-5):

id	mediumint(9)
user	varchar(8)
state	varchar(20)
object	int(11)
PRIMARY KEY	(‘id’)

Table 4-5 *Analysis* table

Second the table, “*Closed_objects*” contains all the closed objects, and it has the schema (Table 4-6).

id	mediumint(9)
object	int(11)
cls_state	varchar(10)
PRIMARY KEY	(`id`)

Table 4-6 *Closed_Objects* table

In the following figure (Figure 4-7) we present the alternative operations of the Node Registration algorithm.

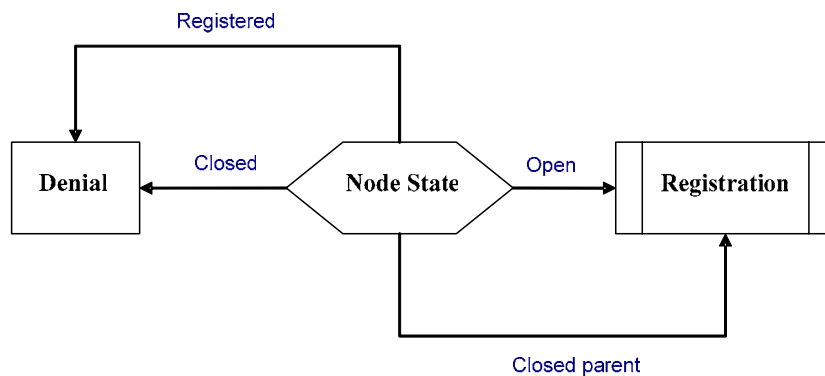


Figure 4-7 *Node Registration* algorithm

All these different states of objects are saved in the specific table called “*Analysis*”, along with information about the designer who affects the specific objects, (Table 4-5).

All information that concerns all operation, object, and other data involved in a sub-operation is sent to the Collaborative Base layer for further execution, (Figure 4-9). It is necessary to mention here that any critical information of objects (properties, details, et cetera) that take part in a particular project is contained in specific tables. These tables are defined as “*Project-Scene*” base, as we will show later.

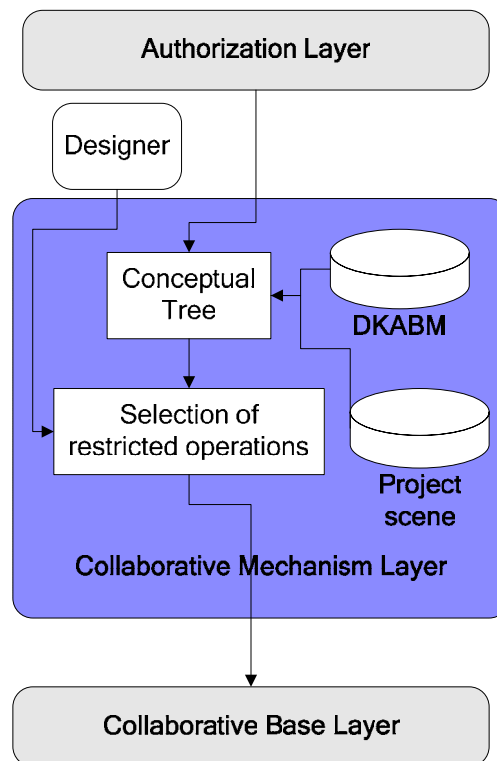


Figure 4-8 Collaborative mechanism layer

Considering all earlier analysis of the Collaborative mechanism the Figure 4-8 provides all related information. The *Collaborative Mechanism* is implemented through a series of PHP pages. These pages contain the defined algorithms, communicate with the *Collaborative Base*. Finally it affects the graphic user interface restricting operations by hiding specific command buttons that correspond to specific operations. Also the pages enable the parametric operations that take input from the designer within certain input fields and information regarding an operation that is passed to the *Collaborative Base Layer*.

4.2.4.4. Collaborative Base layer

In order to enable a necessary link between the *Collaborative Declarative module* and the *Collaborative Base module* we define the *Collaborative Base Layer*. This layer is based in an SQL environment and it is implemented with the use of PHP pages. The process within this layer is the following: information from the

Collaborative Mechanism layer constructs certain SQL statements, which are then triggered by the mechanism. Finally the SQL layer updates the *Collaborative Base*, (Figure 4-9).

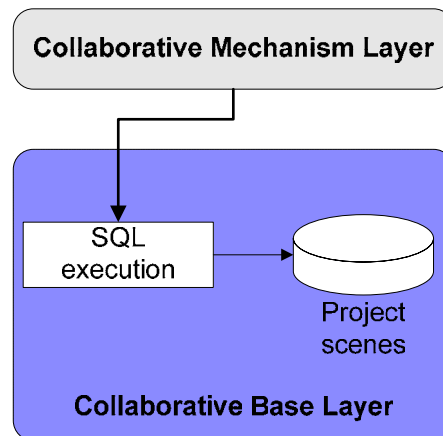


Figure 4-9 Collaborative Base Layer

4.2.4.5. Collaborative Base module

The *Collaborative Base component* incorporates two parts. The first part concerns the *Collaborative Base layer*; the second part concerns the *Collaborative Base*. The *Collaborative base layer* enables the necessary link between the *Collaborative Declarative module* and the *Collaborative Base*. We define this layer as being a common layer between two modules, the *Collaborative Declarative module* and the *Collaborative Base module*. However its functionality has already been defined in the earlier sub-section.

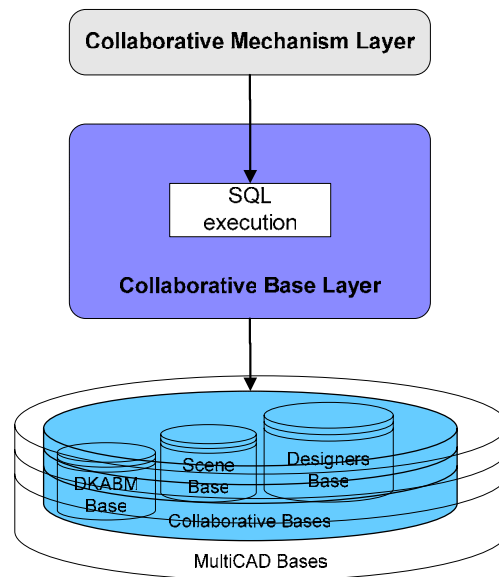


Figure 4-10 Collaborative base module

The *Collaborative Base* is based its functionality on the MultiCAD bases (Knowledge base, Scene base, et cetera). In this way it defines all appropriate bases for the Kernel-CDMS environment (DKABM base, Scene base, Designer base, et cetera), (Figure 4-10).

4.2.5. MultiCAD Designer module

The MultiCAD designer module is implemented with PHP script pages. These pages are finally we define an appropriate graphic user interface (GUI). The MultiCAD designer's module consists of three main layers:

- *General Information* layer
- *Main Workspace* layer
- *General Usage commands* layer.

However these three layers are finally defining the MultiCAD designer graphic user interface (GUI) (Figure 4-11).

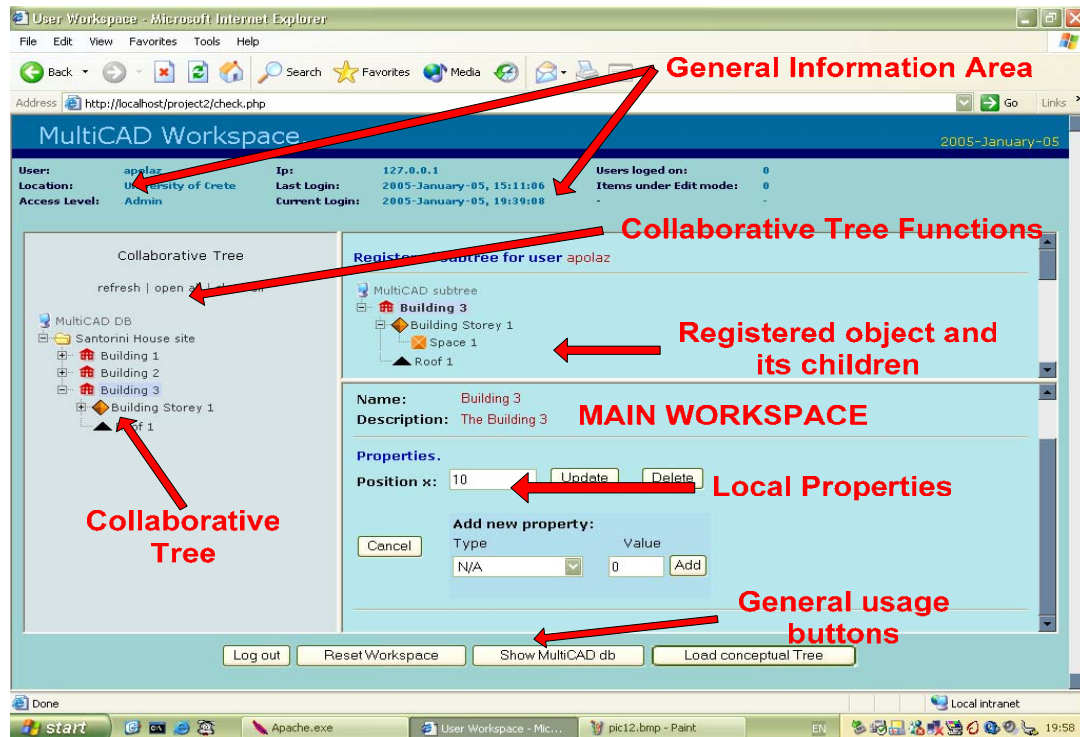


Figure 4-11 MultiCAD designer's browser

The *General Information* layer contains information regarding the designer identity (username, access level, et cetera), (Figure 4-11).

The *Main Workspace* layer is implemented in PHP pages which they enable the designer to use all provided facilities (links to other pages, command buttons, et cetera) in order to use properly the Kernel-CDMS system. In general, these pages are part of the PHP Script Interface that implements the Kernel-CDMS environment, (Figure 4-11).

The *General Usage commands* layer is implemented in PHP pages which they enable the designer to use all provided facilities in the form of specific command buttons within the graphic user interface (GUI). These buttons are command buttons that control the Main Workspace area of the graphic user interface, (Figure 4-11). The graphic user interface is available to designers through a typical web browser.

Finally designers they have at their disposal a graphic user interface in order to view the resulted geometric model(s), after the completion of the generation phase of the declarative conceptual cycle, (Figure 4-12).

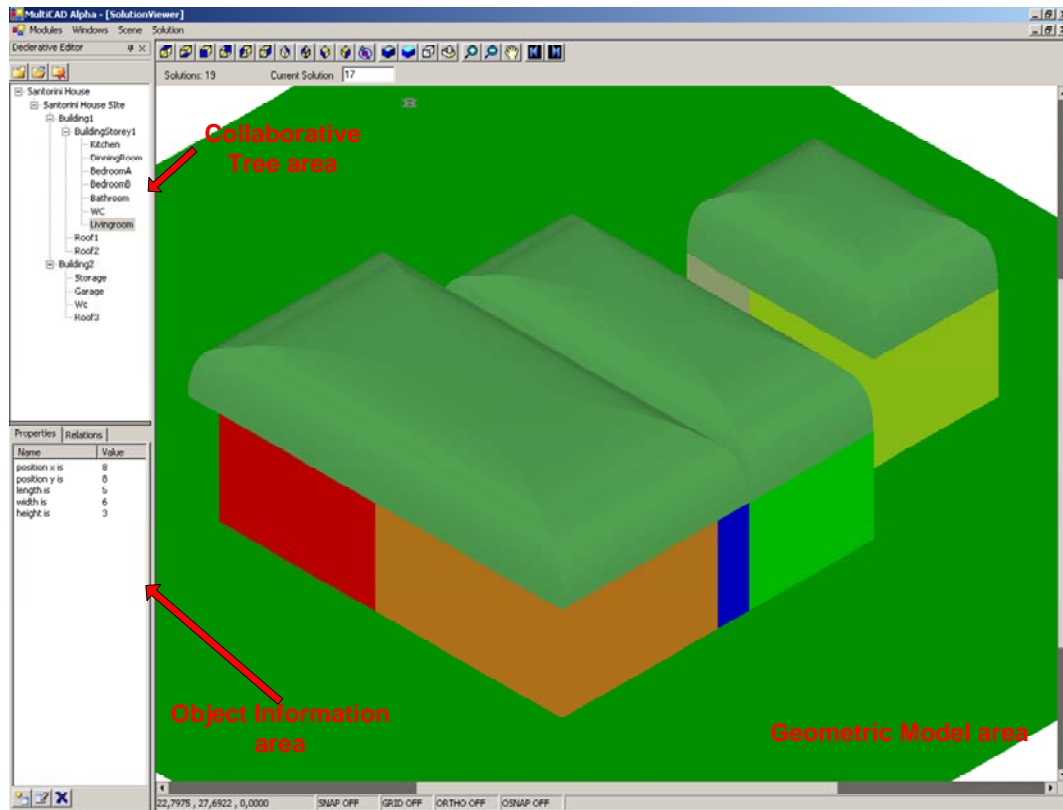


Figure 4-12 Graphic visualisation interface

4.3. Enhanced-CDMS system implementation

We present the development of the *Enhanced-CDMS*. This system operates in HTTP (communication protocols) and its development is based on open source software technology. In particular, the system software scripting is made of several PHP pages. The operating system maintains the operation of the database system (MySQL in the particular implemented system). The *Enhanced-CDMS* system is now installed on a Linux platform. However, its high flexibility makes it transferable to any pre-configured server, with any operating system, without any further major conversions or additions. The demands from the part of the client are the minimum and the only need for an unobstructed functionality of the client is a contemporary web browser like, Firefox, Opera, etc.

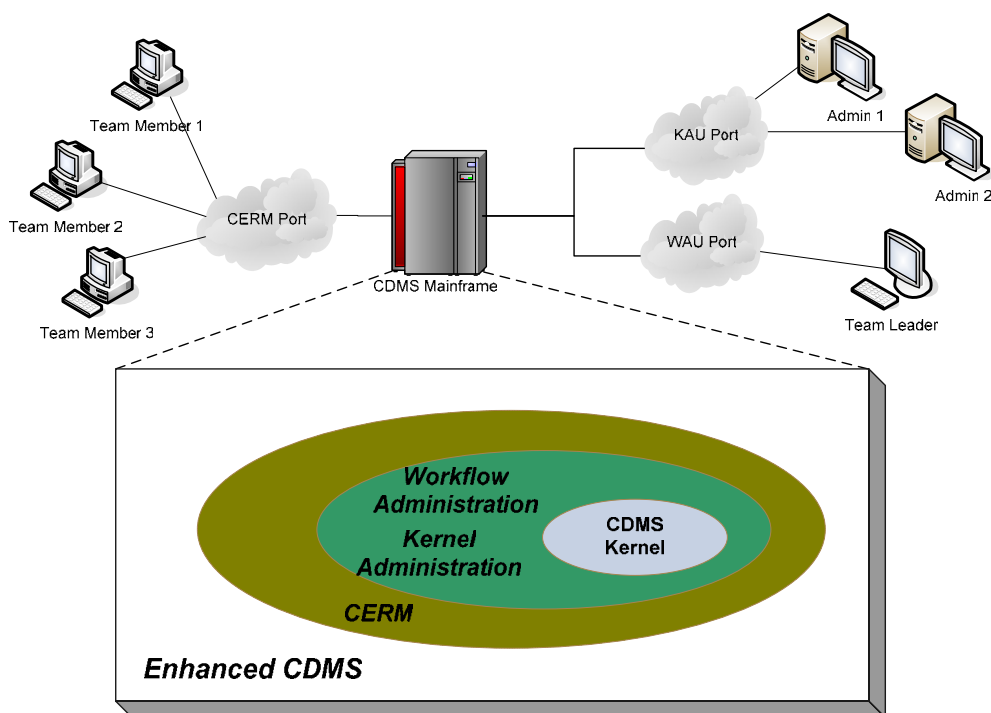


Figure 4-13: System framework

The functionality of the system is based on the overlapping of the Collaborative base by PHP pages that implement the logic structure and the

algorithmic process of the two new mechanisms. In this way these pages built a graphic user interface (GUI) with specific abilities and restrictions. The designer's actions are restricted to actions that are allowed by the PHP pages.

The implementation of the *Enhanced-CDMS* system will be based on the MultiCAD multi-layered architecture. Figure 4-14 presents a system framework of the Collaborative declarative design within the MultiCAD environment.

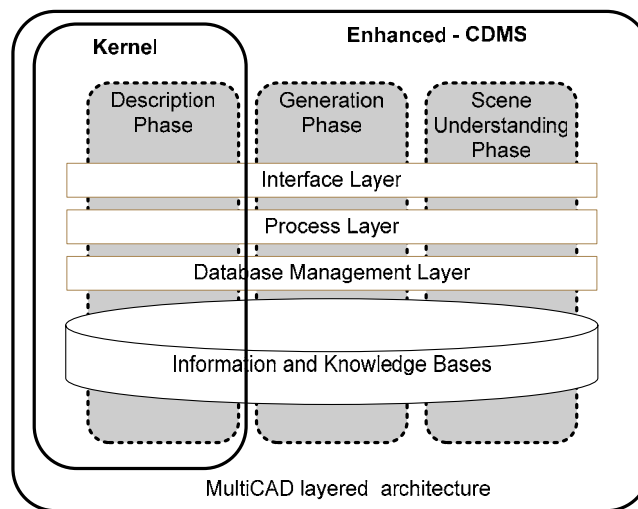


Figure 4-14 Enhanced-CDMS within the MultiCAD environment

4.3.1. Enhanced CDMS modules

The two new parts Collaborative Administration Mechanism (*CAM*) and Collaborative Event Record Mechanism (*CERM*) that define the *Enhanced-CDMS* operate independently in principle. In practice they operate on common information which it is maintained as one entity. Hence these parts are always operating as a pair inside the *Enhanced-CDMS* system.

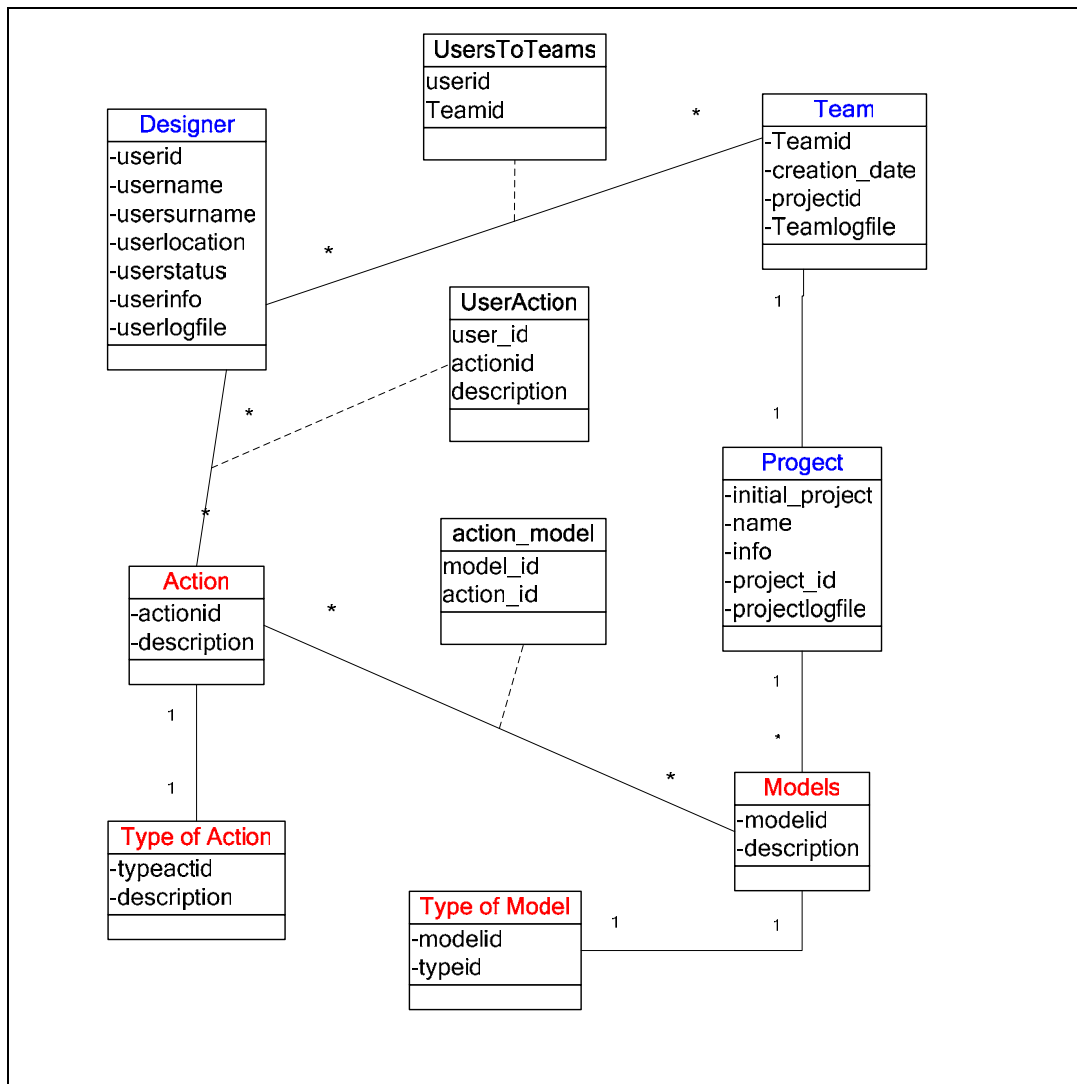


Figure 4-15: Collaborative base schema

The entities used within the Enhanced-CDMS are (*Designer*, *Team*, *Project*, *Conceptual Scene graph*. The entities “*Designer*”, “*Team*”, and “*Project*” include an attribute which defines each one of the following entities: *Designer_Log*, *Team_Log*, and *Project_Log* respectively. In particular, these entities are the structural entities of the Collaborative Event Recorder Mechanism. The DKABM template contains all objects of Conceptual Scene graphs while in this graph a node represents every object and any decomposition relation between objects is represented by a connection. All information relative to DKABMs is maintained by the Collaborative base as in Figure 4-15.

4.3.1.1. Kernel Administration Unit (KAU)

The Administrator entity could manage the previous presented entities (Team Member, Team Leader, et cetera) through the application of this unit. We develop the KAU in the form of PHP pages accessed by authorised Administrators only. The KAU provides a comprehensive and restricted environment to control uniformly and efficiently the Collaborative base which contains the information regarding the entities and their rights.

In order for the Enhanced-CDMS to function properly with the feature of multiple administrators, the KAU should encapsulate a Critical Areas Blockade Algorithm (CABA) (Figures 19-20-21). This algorithm maintains the consistency of the information within the Kernel-CDMS. This CABA operation is based on a special feature that we add to the entities *Designer* and *Team*. In particular, these entities include an attribute “*Entity State*” which can have one of the three possible values: “*Loaded*”, “*Not Loaded*”, and “*Online*” (Figure 4-16).

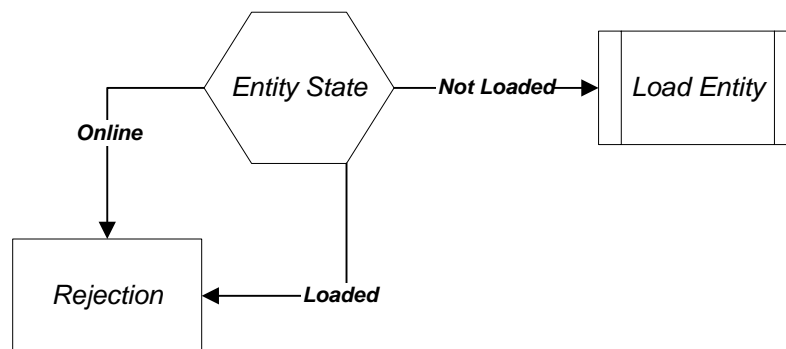


Figure 4-16 CAB Algorithm flowchart

The attribute “*Entity State*” determines if an object of a Conceptual scene graph is ‘*Loaded*’ by another administrator, as the subject of a possible operation, “*Not Loaded*”, or, in the case of a *Designer*, is “*Online*”. The CABA works as in the following pseudocode:

```

precondition: An administrator requested to “Load” a user in order to execute an
operation

while (user.getState != ‘Offline’) {

    if (user.getState == ‘Loaded’)n {

        KAU.deny_operation(administrator);

    } else {

        user.setState(‘Loaded’);

        KAU.execute_operations(administrator);

        User.setState(‘NotLoaded’);

    }

}

```

Figure 4-17 CAB Algorithm operation

The CAB Algorithm controls whether the *Designer* that is going to be loaded is offline or not. While a *Designer* is offline, the algorithm checks if he/she is loaded by another *Administrator*. In case that a *Designer* is already loaded no further operation on him/her is permitted and the algorithm is terminated. In the case where a *Designer* is not loaded, the algorithm loads him/her for the *Administrator* that has posed the request and gives a time-interval for his/her operations. When a *Designer* ends all operations or the interval lapsed, then the *Designer* is altering his/her state to “*Unloaded*” and the CAB algorithm terminates.

The CAB algorithm operates as a FCFS (*First Come First Served*) service to Administrators’ requests, and guarantees a consistent and secured way to control the *KAU* information.

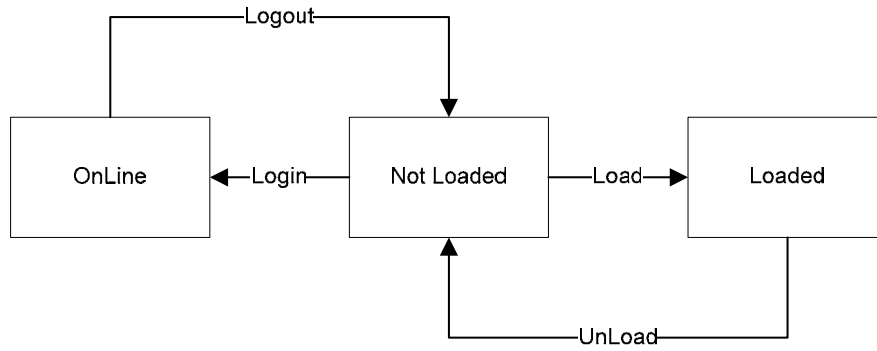


Figure 4-18 Different states of an entity

4.3.1.2. Workflow Administration Unit (WAU)

This unit controls the workflow of the information along with the events triggered by Designers of the CDMS, on other Designers and the Conceptual Scene Graph. This *Unit* is developed as a series of PHP pages which provide a restricted and absolute interface to *Team Leaders*.

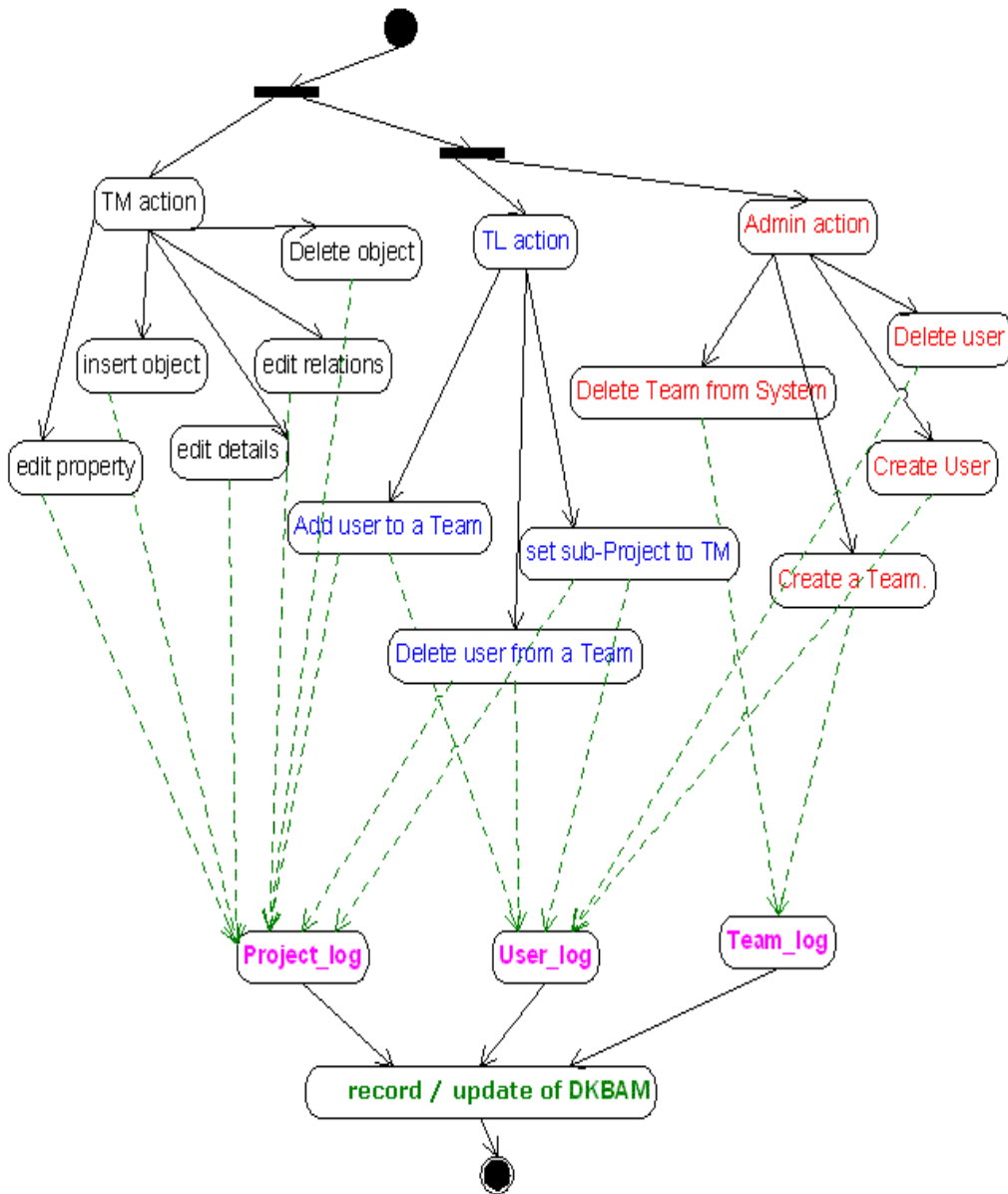


Figure 4-19 CERM operation

4.3.2. Collaborative Event Record Mechanism

This mechanism is a collection of automated functions which concerns the activities of the Team Leader, Team Member and Administrator. The mechanism is implemented in PHP pages, and it operates transparently over all of the events that occur on the CDMS.

Its function is to “*Record*” all the events that occur from the Designers of the Kernel-CDMS, “*Categorize*”, and “*Save*” them to the right database table (*User_log*, *Team_log*, *Project_log*), within the “*Collaborative Base*”, in order for these information to be available to other parts of the *CDMS*. Its operation is described in Figure 4-19. The specific information is available to the users as long as they have rights on it.

Chapter 5

Case Study – System Evaluation

5.1. Introduction

In this chapter we perform a series of case studies for the evaluation and assessment of the two collaborative declarative design system prototypes. The case studies are based on the application of the collaboration paradigm in architectural design process. In this way the plan of the study is organised in two subsections. In particular, in the first subsection we imply within the Kernel-CDMS the first design case. We define a design project which it is developed by two collocated designers.

In the second subsection we define a second case within the Enhanced-CDMS which enables the collaboration of five collocated designers but now with the assignment of specific roles.

5.2. Case study of collaborative activity

The application of the case study involves the collaboration of two designers (idrag, eleni). In particular, they are geographically in distant places, and therefore they are going to connect with the online application <http://quali.teiath.gr/project1>. The designers are going to work on an architectural design project called “Santorini”. The brief of the project concerns the appearance of two buildings “BUILDING1” and “BUILDING2”.

BUILDING1 will have the following objects, (Figure 5-1): Kitchen, Bedroom A, Bedroom B, Living room, Dinning room, Bathroom, WC, Roof1, Roof2. The BUILDING2 will have the following objects (Figure 5-1), Storage, WC, Garage, and Roof3.

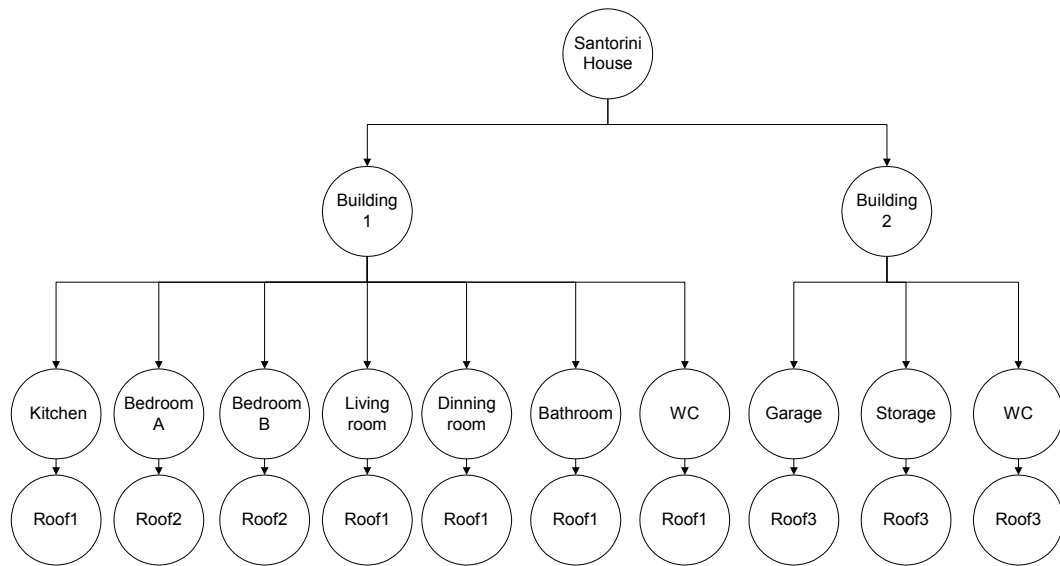


Figure 5-1: Project hierarchical tree.

The project provides specific information considering the spatial relations between the spaces of two buildings. In the case study configuration the Building1 has totally seven spaces and two roofs, (Table 5-1). The Building 2 has three spaces and one roof, (Table 5-2).

Building 1		
<i>Space</i>	<i>Relation</i>	<i>Space</i>
Kitchen	Adjacent North of	Living room
Dinning room	Adjacent West of	Kitchen
Bedroom-A	Adjacent East of	WC
Bedroom-B	Adjacent North of	Bedroom-A
Bathroom	Near West of	Bedroom-B
WC	Near East of	Living room
Roof1	Cover	Kitchen, Living room, Dinning room, Bathroom, WC
Roof2	Cover	Bedroom A, Bedroom B

Table 5-1 Building 1 spatial relations

Building 2		
<i>Space</i>	<i>Relation</i>	<i>Space</i>
Storage	Adjacent North of	Garage
Wc	Adjacent North of	Garage
Roof3	Cover	Storage, Garage, WC

Table 5-2 Building 2 spatial relations

The spaces of both buildings have variable dimensions. The range of their dimensions is introduced by the designer in accordance with the brief. In Table 5-3 we present the ranges for the placement and magnitudes for the seven rooms. The ranges of the magnitudes of spaces that belong to Building 2 are presented in Table 5-4.

Building 1										
Type	X min	X max	Y min	Y max	L min	L max	W min	W max	H min	H max
Kitchen	5	25	5	25	3	4,5	3	4,5	3	3
Bedroom	5	25	5	25	3,5	4,5	3,5	4,5	3	3
Bedroom	5	25	5	25	3	4,2	3	4,2	3	3
Living room	5	25	5	25	4	7	4	7	3	3
Dinning room	5	25	5	25	3,5	5	3,5	5	3	3
Bathroom	5	25	5	25	1,5	2,5	1,5	2,5	3	3
Wc	5	25	5	25	1	1,5	1	1,5	3	3
Roof 1										
Roof 2										

Table 5-3 Objects placement and dimensions

Building 2										
Type	X min	X max	Y min	Y max	L min	L max	W min	W max	H min	H max
Garage	5	25	5	25	3	5	5	5	3	3
Storage	5	25	5	25	3	4	3	4	3	3
Wc	5	25	5	25	1	1,5	1	1,5	3	3
Roof 3										

Table 5-4 Objects placement and dimensions

We will present the specific number of activities that concern with the particular design case. The activities are the following.

1. Designer1 – “eleni” logs in, (Figure 5-2).
2. Designer2 – “idrag” logs in, (Figure 5-2).

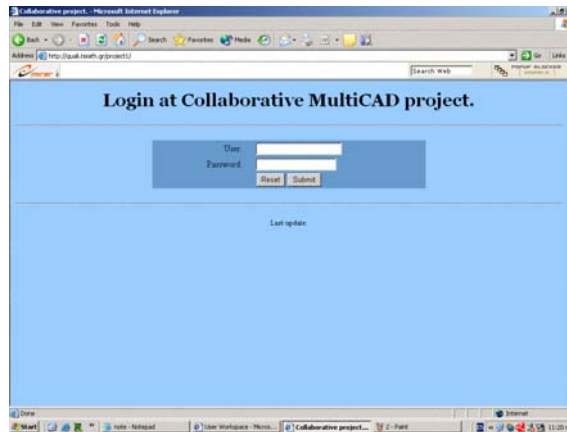


Figure 5-2 Designers login

3. Designer1 registers for edit the project “Santorini House”, (Figure 5-3).

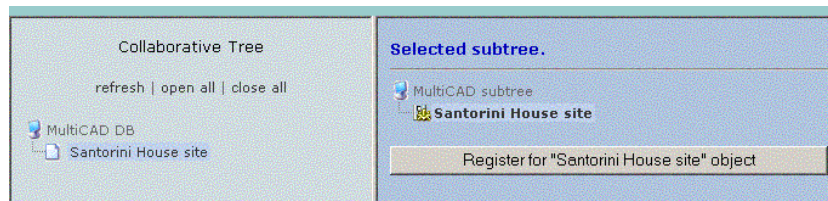


Figure 5-3 Designers registration

4. Designer1 adds the object “Building 1” under the node “Santorini House Site”, (Figure 5-4).

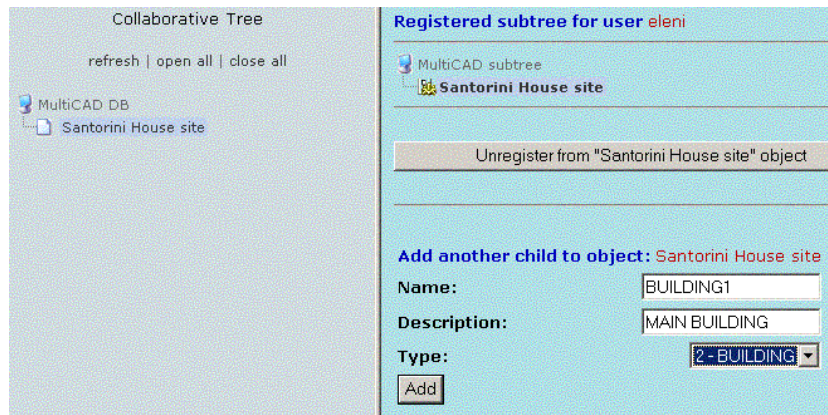


Figure 5-4 Object addition

5. Designer1 adds the object “*Building 2*” under the node “*Santorini House Site*”, (Figure 5-5).

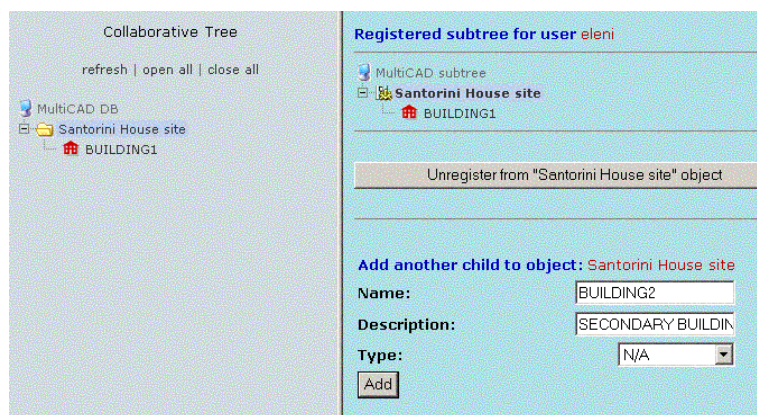


Figure 5-5 Object addition

After the completion of the above actions the Collaborative tree for the project “*Santorini House Site*” is presented in Figure 5-6.

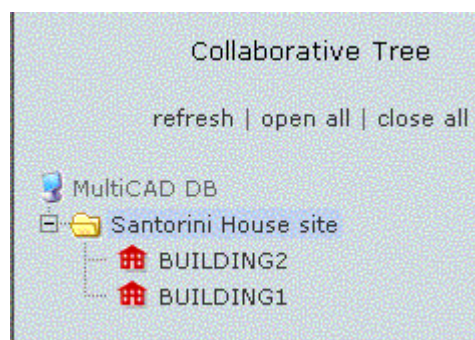


Figure 5-6 Collaborative tree

6. Designer1 gets unregistered from the object “*Santorini House Site*”.
7. Designer1 registers to “*Building1*” and adds another object under it, named “*Building_Storey 1*”, (Figure 5-7).

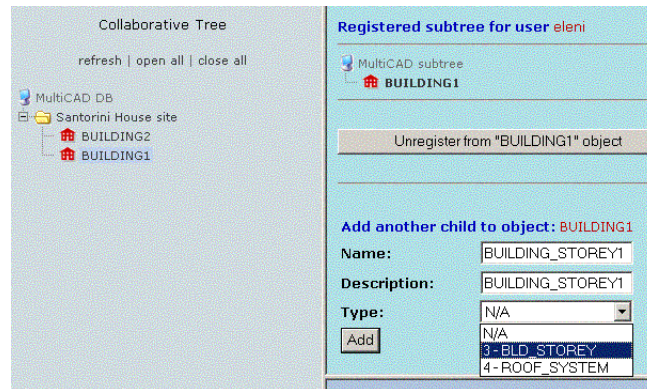


Figure 5-7 Object addition

8. Designer2 is trying to get registered to object “*Building 1*”. The system informs him that he is not allowed to register this object since it is already registered to another designer, (Figure 5-8).

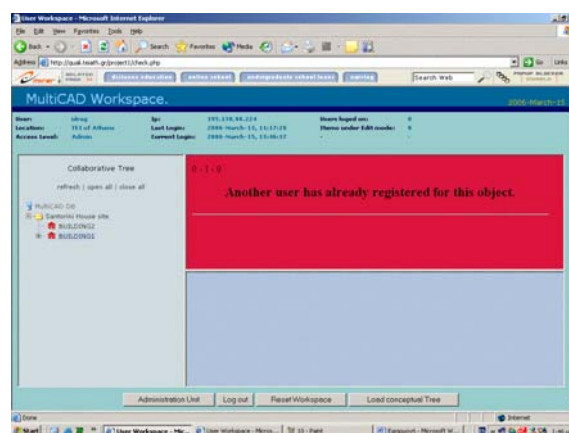


Figure 5-8 System information message

9. Designer1 gets unregistered from the object “*Building 1*”.

10. Designer1 gets registered to object “*Building Storey 1*” and adds 7 objects (of type SPACE), under this node. These objects represent the spaces of this building, and they are the following Kitchen, Bedroom A, Bedroom B, Living Room, Dining Room, and Bathroom, (Figure 5-9).

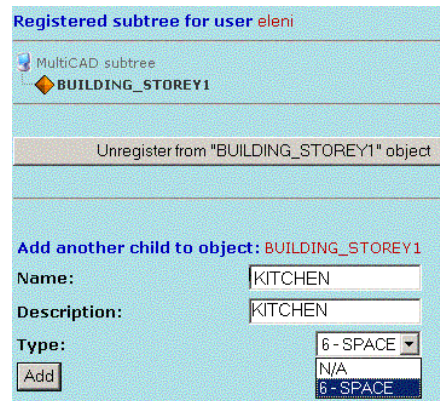


Figure 5-9 Object addition

11. Designer1 gets unregistered from the object “*Building Storey 1*”.

The collaborative Tree of the project is updated and it has the following objects, (Figure 5-10).

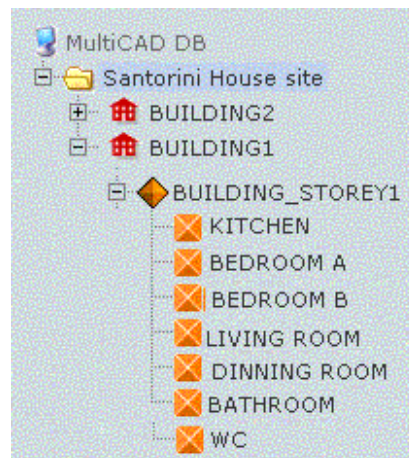


Figure 5-10 Collaborative tree

12. Designer1 gets registered to “*Building 1*” and adds under this node two more objects of type Roof, Roof1 and Roof2, (Figure 5-11).

13. Designer1 gets unregistered.

The Collaborative Tree for the project, it has the following objects, (Figure 5-11).

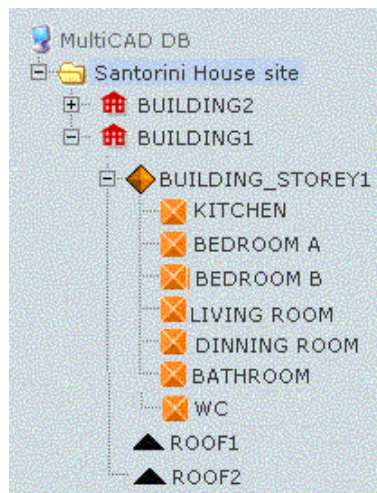


Figure 5-11 Collaborative tree with roof

14. Designer2 gets registered to “*Building 2*”, (Figure 5-12).

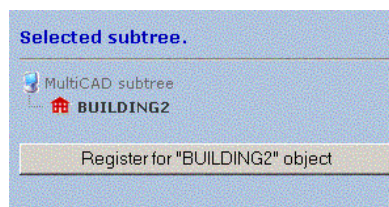


Figure 5-12 Designer registration

15. Designer2 adds another object under the node “*Building2*” named “*Building_Storey2*”, (Figure 5-13).



Figure 5-13 Object addition

16. Designer2 gets unregistered from “*Building2*”.
17. Designer2 gets registered to object “*Building_Storey 2*”.
18. Designer2 adds 3 objects under the node “*Building_Storey 2*” of type SPACE which are, Carage, Storage, Wc.
19. Designer2 gets unregistered from object “*Building_Storey 2*”.
20. Designer2 gets registered to “*Building 2*”.
21. Designer2 adds another object of type ROOF under the node “*Building 2*”. This object is Roof3.

The Collaborative Tree for the project is further enriched and has the following form, (Figure 5-14).

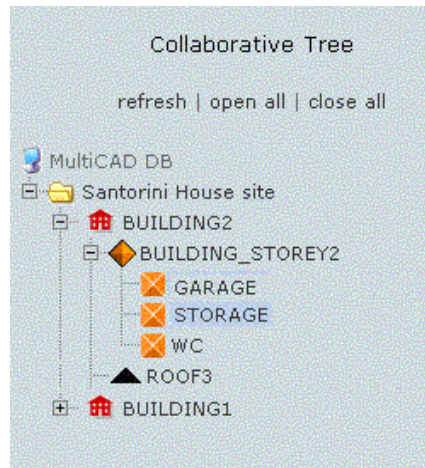


Figure 5-14 Collaborative with Building2

22. Designer1 gets registered to “Building 1”.
23. Designer1 sets relationships between the objects of “Building 1”, (Figure 5-15 - Figure 5-23).

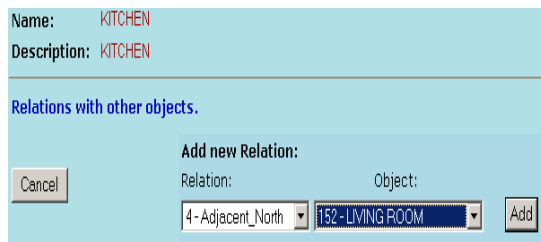


Figure 5-15 Relation addition

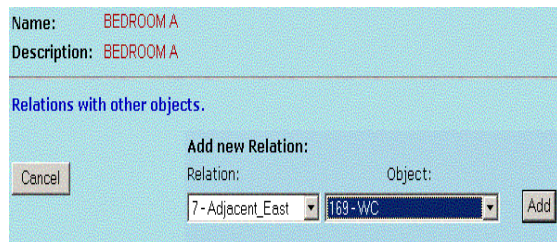


Figure 5-16 Relation addition

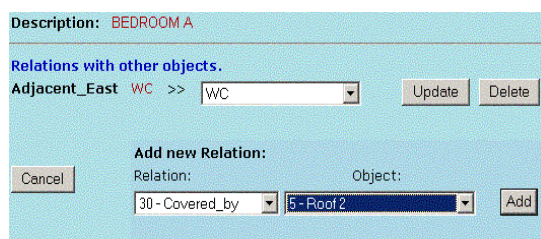


Figure 5-17 Relation addition

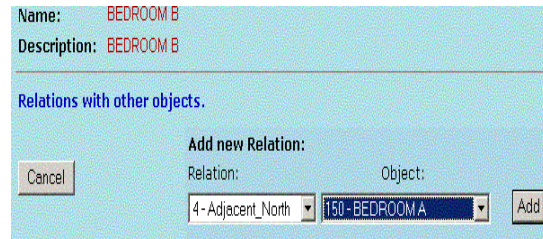


Figure 5-18 Relation addition



Figure 5-19 Relation addition

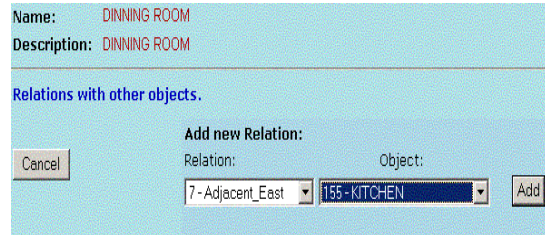


Figure 5-20 Relation addition

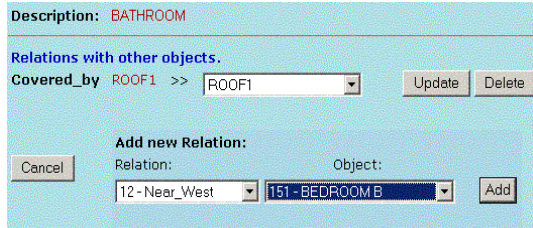


Figure 5-21 Relation addition



Figure 5-22 Relation addition

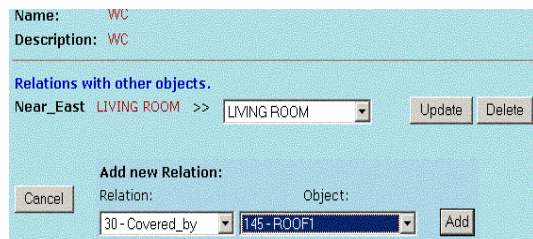


Figure 5-23 Relation addition

24. Since Designer1 finished with setting the relationships of the objects then she sets the values of their properties, (Figure 5-24).

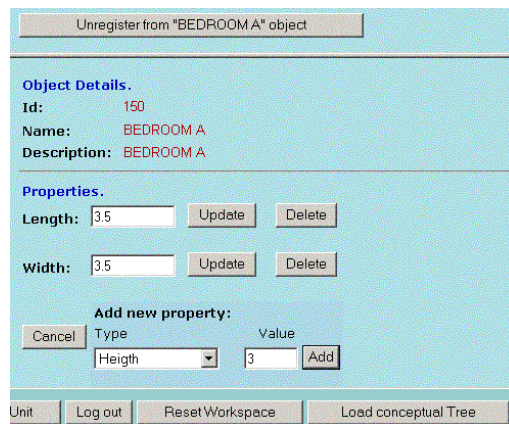


Figure 5-24 Properties' values

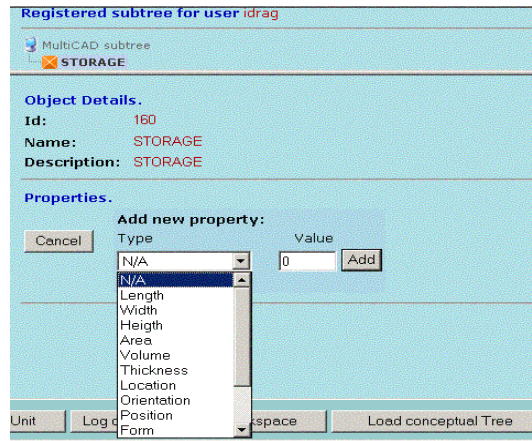


Figure 5-25 Properties' values

25. Designer2 sets the relationships between the objects of Building 2 and the values of their properties in much the same way, (Figure 5-25).

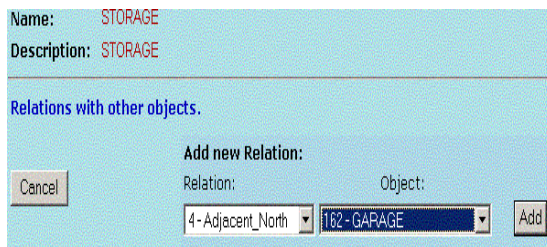


Figure 5-26 Relation addition

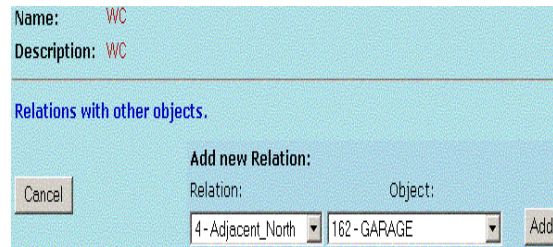


Figure 5-27 Relation addition

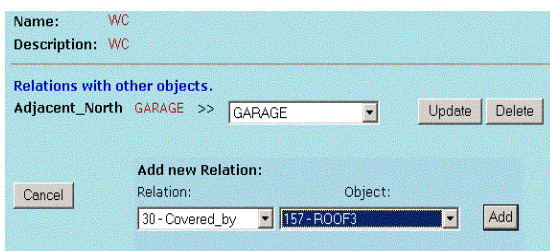


Figure 5-28 Relation addition

26. Since Designer2 finishes the above adjustments on the relations and properties (Figure 5-26 - Figure 5-28) of the objects.

27. Designer1 after the generation phase views the geometric model of the “Santorini House” project description, (Figure 5-29, Figure 5-30).

28. Designer2 after the generation phase views the geometric model of the “Santorini House” project description, (Figure 5-29, Figure 5-30).

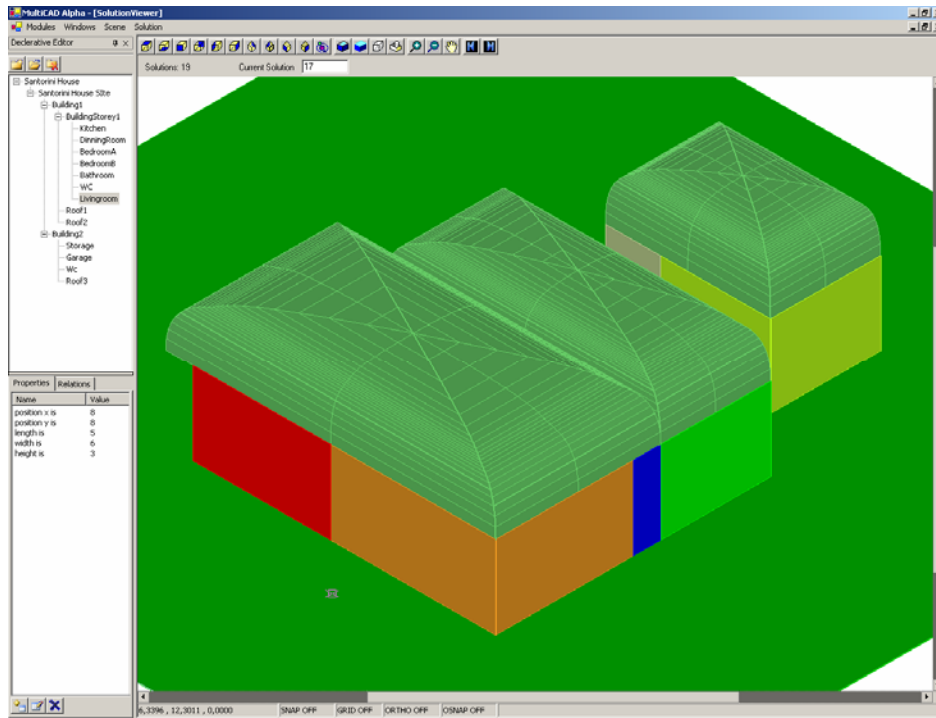


Figure 5-29 “Santorini House” geometric model (spaces and roofs)

(Note: the correspondence between objects and colours is the following:
Living room: orange; dinning room: red; kitchen: blue; bathroom: cyan; wc: blue;
bedroom: green; garage: yellow; storage: grey.)

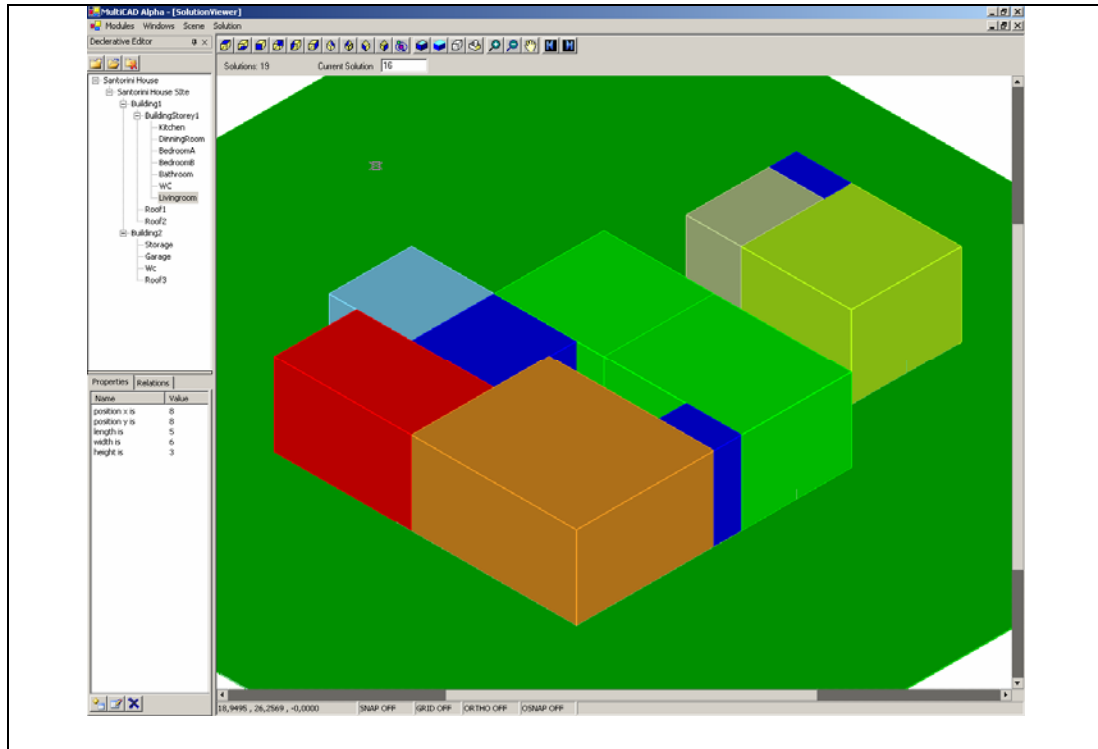


Figure 5-30 “Santorini House” geometric model (spaces)

29. Designer1, she logs out (Figure 5-31).
30. Designer2 he logs out (Figure 5-31).



Figure 5-31 Designer logout

5.3. Case study for Enhanced-CDMS

This case study is introduced in order to assess the usability of the *Enhanced-CDMS* system. This case study considers five geographically distant designers of the system. The case study presented in this subsection concerns cases relating to *Kernel Administrator Unit (KAU)* and *Workflow Administrator Unit (WAU)*.

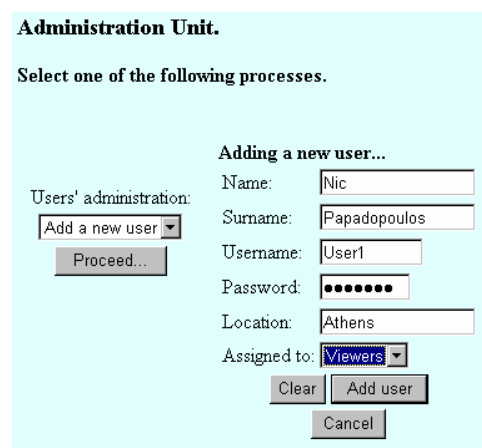
5.3.1. KAU case study

We consider that there are two administrators {Admin1, Admin2} already logged in the system. The user interface of KAU provides them with the necessary utilities in order to:

- “View” the exact current state of the users of the system (loaded, unloaded, online), along with their details.
- “Request” to load a *User / Team* for further process.
- “Create” a *User / Team*.
- “Edit” / “Delete” of a loaded *User / Team*.
- “Release” a loaded *User / Team*.

The activities are introduced with the following order.

1. Admin1 creates User1, (Figure 5-32).



The screenshot shows a web-based interface titled "Administration Unit." Below the title, it says "Select one of the following processes." There are two main sections. On the left, under "Users' administration:", there is a dropdown menu with "Add a new user" selected and a "Proceed..." button. On the right, under "Adding a new user...", there are several input fields: "Name:" with "Nic", "Surname:" with "Papadopoulos", "Username:" with "User1", "Password:" with a masked field of seven dots, "Location:" with "Athens", and "Assigned to:" with a dropdown menu showing "Viewers". At the bottom of this section are three buttons: "Clear", "Add user", and "Cancel".

Figure 5-32 User1 creation

- Admin1 requests to load User1, (Figure 5-33).

The screenshot shows a dialog box titled "Load an existing user...". On the left, there is a section labeled "Users' administration:" with a dropdown menu showing "Edit a user" and a "Proceed..." button. The main area of the dialog contains the following fields and controls:

- Username: A dropdown menu with "User1" selected and a "Load..." button.
- Name: A text input field containing "Nic".
- Surname: A text input field containing "Papadopoulos".
- Password: A text input field with masked characters (dots).
- Location: A text input field containing "Athens".
- Assigned to: A dropdown menu with "Viewers" selected.
- At the bottom, there is a "Save new data for user User1..." button and a "Cancel" button.

Figure 5-33 Existing User1 load in

- Request is accepted – User1 is loaded
- Admin2 requests to load User1.
- Request is rejected, (Figure 5-34).

The screenshot shows a dialog box titled "Administration Unit." with the instruction "Select one of the following processes." Below this, a message states: "User User1 is already loaded by another user. Please try later." On the left, there is a section labeled "Users' administration:" with a dropdown menu showing "Edit a user" and a "Proceed..." button. To the right of the message, there is a "Finish" button.

Figure 5-34 Request rejection

- Admin2 creates User2 (Figure 5-35).
- Admin1 edits User1, (Figure 5-36).
- Admin1 releases User1.

9. Admin2 requests to load User1.
10. Request accepted – User1 loaded
11. Admin2 edits User1 (Figure 5-36).

The screenshot shows a light blue window titled "Administration Unit." Below the title is the instruction "Select one of the following processes." On the left, under "Users' administration:", there is a dropdown menu with "Add a new user" selected and a "Proceed..." button below it. On the right, under "Adding a new user...", there are several input fields: "Name:" with "John", "Surname:" with "Smith", "Username:" with "User2", "Password:" with a masked field of seven dots, "Location:" with "Sydney", and "Assigned to:" with a dropdown menu showing "Viewers". At the bottom right of this section are three buttons: "Clear", "Add user", and "Cancel".

Figure 5-35 User2 creation

The screenshot shows the same "Administration Unit" window. The instruction "Select one of the following processes." is still present. On the left, under "Users' administration:", the dropdown menu now shows "Edit a user" and the "Proceed..." button is below it. On the right, under "Results:", the text "Data for user User1 updated." is displayed. Below this text is a "Finish" button.

Figure 5-36 User1 edit

12. Admin2 defines TL from Viewers (Figure 5-37).
13. Admin2 releases User1.

Administration Unit.
Select one of the following processes.

Users' administration:
Add a new user ▼
Proceed...

Adding a new user...
Name: Nic
Surname: Papadopoulos
Username: User1
Password: ●●●●●●
Location: Athens
Assigned to: TL ▼
Clear Add user
Cancel

Figure 5-37 Team Leader definition

14. TL defines TM1 from Viewers (Figure 5-38).

Administration Unit.
Select one of the following processes.

Users' administration:
Add a new user ▼
Proceed...

Adding a new user...
Name: John
Surname: Smith
Username: User2
Password: ●●●●●●
Location: Sydney
Assigned to: TM ▼
Clear Add user
Cancel

Figure 5-38 Team Member definition

15. TL releases TM1.

We utilise the same example also in the case when administrators request to load in order to define more *Team Member*, or a *Group*. A case where a request can be rejected is when the object (*Viewer*, *Team Leader*, and *Team Member*) to be loaded

is already online. Generally a successful request changes the state (from unloaded to loaded) of an offline object (*Viewer*, *Team Leader*, and *Team Member*), (Figure 5-39).

The state of a loaded object can be restored to its default value (unloaded) for two reasons. First, if the administrator releases the loaded object on his/her demand. Second, when a specified time interval expired since the load of the object (such time interval could be defined internally), (Figure 5-39).

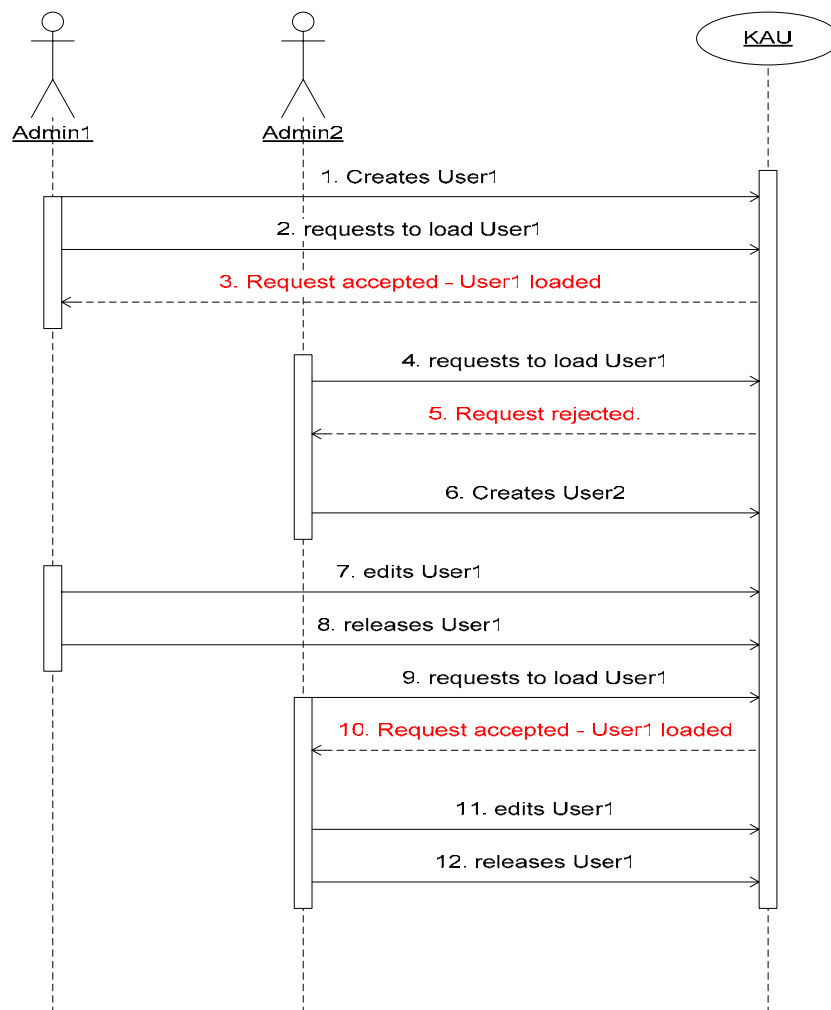


Figure 5-39 KAU example

5.3.2. WAU case study

We have presented during the system implementation (Chapter 3), that the Workflow Administrator Unit provides to Team Leaders the tools to manage the cycle life of a design project (Figure 5-40). In this way within the WAU the Team Leader is provided with all the necessary tools for the following functions:

- Initialization of a project.
- Versioning of the project.
- Administration of the Team.
- Collecting project proposals.

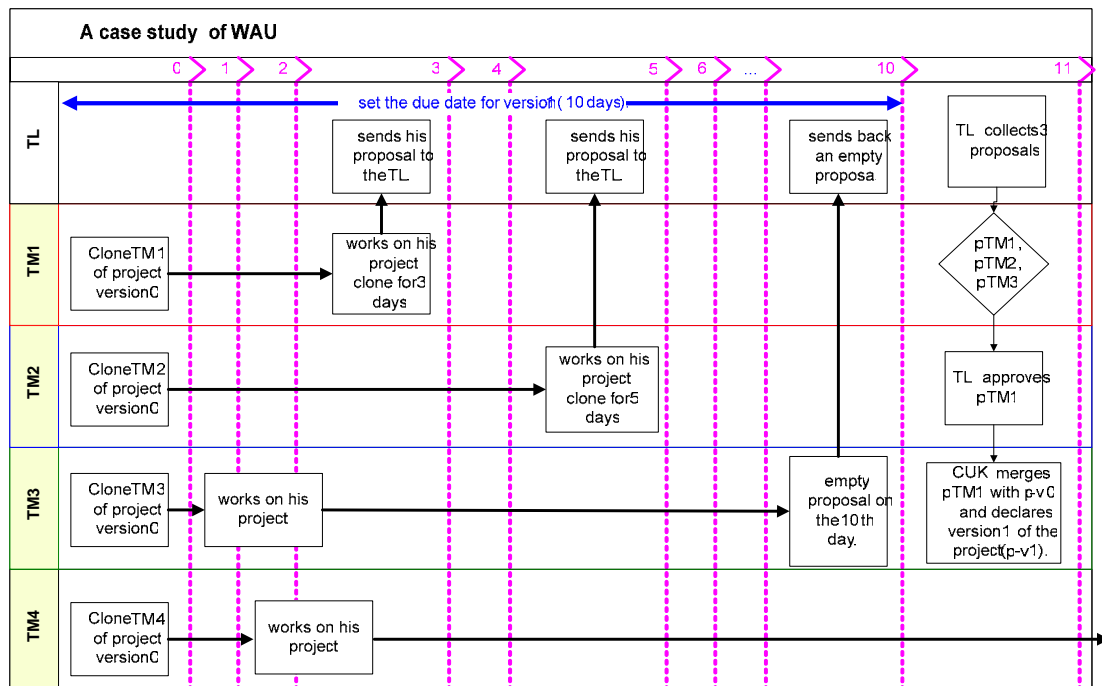


Figure 5-40 WAU case study

We consider as an example the case where there is one “Team” for a “Project A”. The team has a Team Leader (TL1), and four particular Team Members (TM1, TM2, TM3, and TM4). In particular, we consider that the TL1 has already initiated the project, he/she selected the Conceptual Tree, he/she set Version 0 (p-v0) as the

latest stable version of the project, and finally he/she added the Team Members (with access rights) to the Team, (Figure 5-40). A case study is the following:

1. TL1 set the due date for version 1 (in this case is set up to 10 days), (Figure 5-41).

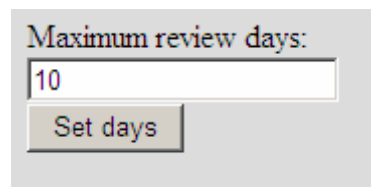


Figure 5-41 Set of due date

2. Team Members clone version “0” of the project.
3. TM1 works on his/her project clone for 3 days and sends his/her proposal to the TL1, (Figure 5-42).

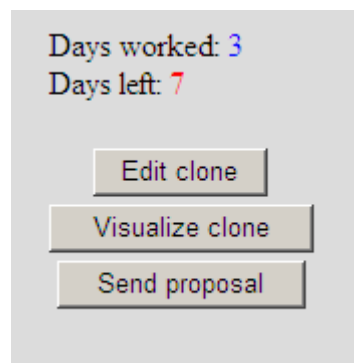


Figure 5-42 Project working period

4. TM2 works on his/her project clone for 5 days and afterwards he/she send his/her proposal to the TL1.
5. TM3 doesn't work on his/her project clone and he/she sends back an empty proposal to the TL1 on the 10th day.
6. TM4 works on his/her project clone but finally, he/she doesn't send any proposal to the TL1.

7. TL1 collects (on the 11th day) three proposals (pTM1, pTM2, pTM3) from the three team members, (Figure 5-43).

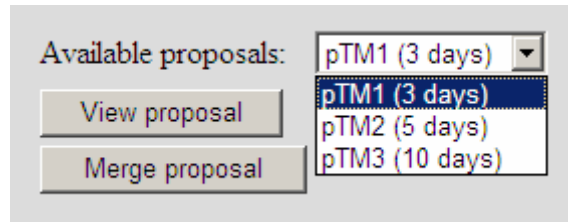


Figure 5-43 Proposals collection

8. TL1 assesses the proposals and approves one, the pTM2 only.

9. TL1 merges pTM2 with p-v0 and declares version 1 of the project (p-v1).

However, this cycle can be repeated for a number of times. If TL1 decides that a version is adequate to his/her requirements, then he/she can proceed as follows:

10. TL1 closes the project at version N.

11. TL1 Delete Team Members from Team.

12. TL1 sends message to administrators for project completion.

A Team Member effort concerns the individual member design process of a project. These efforts consist of the three phases of the declarative design process, which they are analysed as follows:

A. New Declarative Scene Description (amended CT).

B. Solutions Generation.

C. Scene Understanding-Evaluation.

5.4. Case study conclusions

In this chapter we imply two case studies in order to evaluate the capabilities of the two proposed parts of the collaborative declarative design system Enhanced-

CDMS. Both case studies provide useful information regarding the usability of the final complete system.

5.4.1. Kernel – CDMS

The Kernel-CDMS environment provides the designers with all appropriate functionality in order to take part in a distributed collaborative development of an architectural project. The main items of the Kernel-CDMS are the following:

- *Designer*
- *Scene graph.*

The possible actions of a Designer to the Scene graph are the following:

- *Register / Un-register* to a scene graph node
- *Edit / Add / Delete* a Property / Relation / Detail of a scene graph node.

Additionally the Kernel-CDMS takes the following related information from the Collaborative Base such as:

- The type of the *Scene graph*,
- The *Designer* information (access rights manipulations, et cetera) over the *Scene graph*.

The application of a design case enables observations over the final functionality and the initial design principles of the Kernel-CDMS. In particular, we saw a set of advantages and some limitations. The resulted Kernel-CDMS environment provides an efficient and effective collaborative environment within a declarative design system such as MultiCAD. Up to now the context of application of the Kernel-CDMS concerns only a number of participated designers. The design operation is based on logic of “*lock – unlock*” of declarative scenes. Therefore the development of an architectural project can regarded as a semi-serial process. This happens because multiple designers can work on non-correlative sub-trees of the same project but in FCFS (First Come First Served) logic. The Kernel-CDMS is appropriate for projects when from the beginning of the design process exists distinct roles for the designers on specific parts of the architectural design project. The main contribution of the Kernel-CDMS is the synthesis of collaborative design with declarative design.

During the completion of the design cases we observe that some of the above characteristics are sources of possible drawbacks. These are database inconsistency, and time-consuming operations. In particular, some of the limitations that emerge from the Kernel-CDMS are the following:

- The administration of the Designers' rights can be done only from a super-administrator who has access right to the Collaborative base tables, and can be done only directly to them.
- Because of the distributed collaborative environment workspace for a Designer, there is a block algorithm which locks-unlocks different parts of the Conceptual Tree.

However the Kernel-CDMS environment was further improved in order to confront these limitations. Finally it plays an important role within the total Enhanced-CDMS system.

5.4.2. Enhanced-CDMS

The total system Enhanced-CDMS is assessed in order to evaluate its feasibility and functionality. The Enhanced-CDMS firstly, it provides role management of the designers by defining working hierarchies and working groups between the designers. Its basic structure is based on the Kernel-CDMS. Additionally it introduces the concept of “*Role*” within a collaborative environment. Secondly, it provides specific facilities for the management of the projects. In particular it defines specific sub-parts for a project and it defines working-parts for designers' working groups. In this way The Enhanced-CDMS firstly, provides administration utilities, for the distributed collaborative environment. Secondly it provides administration support to the collaborative declarative design workflow process.

The operation of the Enhanced-CDMS is based on a *parallel-cycle* algorithm. The development of an architectural project can be regarded as a parallel process for each designer within a Team. The Team Leader decides the approval of a proposed design, and creates a version of the project based on the designers' proposals. Each version of the project is given to all designers of the Team. Each designer follows a design process (that of the declarative design cycle), on the project and creates his/her

proposal(s). The process of the proposals development is a parallel process for all the designers.

The Enhanced-CDMS is flexible and it is transferable to any preconfigured server, with any operating system, without any further major conversions or additions.

The evaluation has proved that the prototype system functionality runs without any problem. This system is suitable in projects where the requirement of rapid development is an important factor, and the availability of many possible designs is necessary.

Chapitre 6

Discussion-Conclusion

6.1. Discussion

Aujourd'hui, les problèmes liés à la conception sont par nature complexes et nécessitent un travail en collaboration de plusieurs concepteurs. Dans cette thèse, nous présentons le développement d'un environnement de conception déclarative collaborative efficace qui facilite l'échange et le partage d'informations entre les concepteurs au moyen d'Internet. L'architecture et l'approche proposées sont distinctes d'autres systèmes traditionnels de conception en collaboration basés sur le Web, du fait qu'on intègre en plus la modélisation déclarative aux systèmes basés sur le Web pour offrir une plus grande flexibilité au processus de conception. En particulier, nous proposons une architecture et un cadre qui intègre les technologies Web et la modélisation déclarative qui sont adéquates au développement de systèmes de conception collaborative multidisciplinaires. Nous démontrons que l'autonomie de la conception peut être étendue simultanément (ou parallèlement) à des groupes de disciplines différentes (pluridisciplinarité) dans un projet de conception décomposé au moyen de la modélisation déclarative collaborative. On donne à ces groupes « disciplinaires » la liberté de créer leurs conceptions et de formuler et manipuler les propriétés et contraintes des objets associés à chaque concept. Nous présentons les résultats de notre système original en appliquant l'architecture proposée pour la conception de construction de projets en collaboration. Les contributions essentielles de cette recherche incluent :

- Une intégration de la modélisation déclarative par décomposition hiérarchique avec l'Internet et les technologies Web afin de permettre une plus grande flexibilité au cours du processus de conception (description de la conception),
- Un support de collaboration en temps réel de développement du produit (dans notre cas : les bâtiments) qui permette la communication entre des serveurs Web actifs.

Un cadre conceptuel prototype de ce système a été esquissé et décrit de manière détaillée. Le cadre contient les modules fonctionnels requis pour développer le système et toutes les bases de données relatives avec les mécanismes de gestion des connaissances. Ce prototype a permis d'étendre avec succès un système logiciel de conception déclarative monoposte à des applications multiposte.

Il serait intéressant de faire une comparaison de notre système avec les autres systèmes existants. Cependant il est difficile d'évaluer les différents systèmes de collaboration parce que leur succès dépend beaucoup de la technologie qui est utilisée par chacun d'eux ainsi que des différents besoins et styles de travail des participants. Il y a un grand nombre de facteurs potentiellement importants. Même les moyens pour mesurer la qualité de l'interaction du groupe et de la technologie qu'il utilise sont pratiquement insaisissables. Ils incluent des notions concernant l'efficacité des réseaux utilisés et la répartition géographique du groupe. Etant donné qu'il n'est pas facile de comparer technologiquement les différents systèmes de collaboration, nous présentons dans le tableau suivant (Tableau 6-1), une comparaison qualitative de projets non – commerciaux similaires.

Projets Collaboratifs						
Projets	<i>NetFeature</i>	<i>CSM</i>	<i>WebSpiff</i>	<i>Syco3D</i>	<i>CyberCAD</i>	Enhanced-CDMS
Année de publication	1999	1999	2000	2001	2002 - 2005	2006
Domaine	C	C	C	C	C	C
Données	S - C	S - C	S	S	C	S
Traitements	S	S	S	C	S	S
Approche	-	V	S	P	S	V- E
Niveau	M	P	D	P	M	M - D
Cote serveur	S - G - B	S - G	S - G - W	S	S - G - W	S - B - H - W
Cote client	W	W	W	A - F	W	L - W
Modeleur	G	G	G	G	G	D
Types d'Information	G	G	G	G	G	S - G

Tableau 6-1 Comparaison qualitative de projets collaboratifs

Légende	
Domaine	C pour CAO, S pour sculpture, M pour modélisation 3D
Données	S pour sur le serveur, C pour réplification sur les clients
Traitements	S pour sur le serveur, C pour sur les clients
Approche	V pour verrouillage, P pour partitionnement, S pour sérialisation, E pour exécution réversible et – pour pas de gestion des conflits
Niveau	O pour objets, M pour modèle, A pour application, P pour pièce d'un assemblage, D pour données
Cote serveur	S pour gestion des sessions, G pour gestion de la géométrie, B pour base de données, A pour application de CAO ou autre, H pour historique des sessions, W pour serveur Web et – pour pas de serveur
Cote client	A pour application de CAO ou autre, B pour base de données, C pour composant de distribution, L pour client léger, F pour fenêtre supplémentaire, W pour navigateur Web
Modeleur	G pour modeleur géométrique, F pour modeleur base sur la caractéristique (feature-based), D pour modeleur déclaratif
Types d'Information	G pour information géométrique, S pour information sémantique

Tableau 6-2 Légende de Tableau 6-1

6.2. Conclusion

Cette thèse présente un ensemble de proposition qui aboutissent à un système logiciel original, le Enhanced-CDMS qui est indispensable pour supporter la conception déclarative collaborative. Le système de conception qui en résulte présente pour les concepteurs les principaux avantages suivants :

- Prise en compte de la dimension intuitive d'une collectivité de concepteurs. L'intuition est une qualité essentielle du modéleur déclaratif. La modélisation déclarative permet d'approcher la conception d'une scène d'une façon plus intuitive. Un modéleur déclaratif offre des concepts d'un degré supérieur pour la description de la conception. Les concepts sont très proches du langage naturel. Ils appartiennent au domaine d'un modéleur déclaratif spécialisé.
- Inspection automatique des propriétés de la scène par l'ensemble des concepteurs. Les concepteurs fournissent au modéleur déclaratif la description d'un projet. La description possède toutes les propriétés et relations existant entre les objets d'un projet. Ensuite le modéleur manipule toutes les informations, et teste et vérifie tous les critères intégrés à une description.
- Manipulation des aspects non-géométriques de la scène simultanément par l'équipe de concepteurs. La modélisation déclarative permet la description d'un projet à un niveau supérieur. Elle permet l'utilisation de descriptions précises et imprécises et n'est pas basée sur une connaissance complète des modèles géométriques sous-jacents ni sur la capacité de dessins des concepteurs.
- Meilleure adaptation au processus de conception collective, avec la possibilité d'une participation étendue de chaque membre de l'équipe. La modélisation déclarative est bien adaptée au processus de conception. Elle permet d'approcher un objet à partir de différents points de vue. Elle peut unifier de façon parfaite le développement des projets de conception qui répondent aux nombreuses propriétés désirées.

En particulier, la Modélisation Déclarative par Décomposition Hiérarchique (MDDH) adoptée présente et approfondit, dans sa version collaborative un certain nombre d'avantages précis pour le processus de conception :

- Les conceptions sont exprimées de façon progressive par le(s) concepteur(s). Une conception peut être précisée de façon plus ou moins détaillée.
- Il renforce la place de la description (La compétence du concepteur est centrée sur l'explication d'une partie de l'ensemble de la conception sans qu'il s'occupe des autres)
- Il permet la factorisation des propriétés durant la description de la conception.
- Il suscite des relations fortes entre un scène et un sous-scène (Un nœud hérite des contraintes de ses ascendants.)
- Il définit une forte indépendance entre les objets des sous-arbres issus du même nœud de l'arbre de décomposition hiérarchique : les propriétés peuvent être exprimées sur les caractéristiques d'un seul nœud ou sur celles d'un nœud et des nœuds qui en descendent directement.

L'Enhanced-CDMS proposé est un système de conception collaborative classique depuis qu'il profite pleinement des avantages basées sur le Web pour assurer au système stabilité, fiabilité et facilité d'utilisation.

En profitant des utilisations efficaces des technologies Web, ce système est indépendant du système d'exploitation. Les services situés sur Internet offrent au concepteur un moyen efficace de conduire un processus de conception, de définir et manipuler une conception, qu'il le fasse lui-même ou avec d'autres en cas d'activité de conception partagée.

En particulier, le système Enhanced-CDMS proposé est caractérisé par le fait qu'il n'exige pas un modéleur géométrique solide de la part du client. Cette opération est accomplie du côté du serveur de façon à ce que les clients soient les plus faibles possibles. Cela permet une installation facile et l'utilisation du système par des concepteurs qui ont des ressources matériels et logiciels limitées. De cette façon, l'architecture de l'Enhanced-CDMS est un modèle connecté, orienté client-serveur utilisant le protocole TCP/IP. L'outil spécifique est une architecture serveur fort/client faible.

A partir des études de cas, on peut identifier quelques-unes des caractéristiques du prototype Enhanced-CDMS :

- En utilisant une description déclarative les concepteurs ont la possibilité de partager des données à distance et peuvent rapidement et facilement créer et éditer un modèle de conception déclarative.
- Le cycle de conception déclarative qui décrit les trois phases accomplies par la scène en modélisation déclarative, est introduit dans un environnement collaboratif. Dans cet environnement, les concepteurs travaillent comme clients et partagent des données en mode client-serveur, pour contribuer à la description déclarative de la scène.
- Le mécanisme du prototype Enhanced-CDMS : permet aux concepteurs de travailler ensemble sur différentes sous-parties de la même scène.
- Le mécanisme du prototype Enhanced-CDMS offre deux choses à la modélisation déclarative collaborative :
 - des moyens d'administration pour les concepteurs.
 - un support à la procédure d'administration des flux de travail (workflow)
- Par l'application du concept de groupes, le chef d'équipe peut explorer les idées et propositions des différents membres du groupe (concepteurs) non seulement sous la forme d'un modèle statique, mais aussi sous celle d'une simulation dynamique. Cela permettra d'avoir, au cours de la phase de description déclarative, un mécanisme d'évaluation efficace pour vérifier les conceptions des différents membres du groupe.
- Ce système donne aux concepteurs la possibilité de distribuer leurs conceptions sur Internet. Ils peuvent partager des modèles et des données avec leurs partenaires impliqués dans le processus de développement d'un bâtiment, sans qu'il soit nécessaire d'avoir des clients (workstations) chers et un logiciel de CAO.

En résumé, le système de conception déclarative en collaboration peut faciliter le processus de conception en :

- Développant rapidement un modèle de scènes
- Améliorant la compréhension des besoins et des idées pour tous les concepteurs impliqués.

- Facilitant la communication de sorte que l'on passe moins de temps en meeting où les participants sont présents.
- Réduisant le besoin d'investir dans un plus grand nombre de clients (workstations) et logiciel CAO.

6.3. Travaux futurs – Perspectives

Nous considérons que les quelques problèmes présentés par le prototype Enhanced-CDMS doivent être répertoriés et que des efforts d'améliorations sont nécessaires dans l'avenir proche. Ces problèmes concernent l'efficacité de la coordination du projet, la gestion du groupe et des multi-projets et la composition dynamique des flux de travail (workflows).

En général les futurs efforts seront centrés sur les points suivants :

- Améliorations de la gestion des multi-projets et multi-concepteurs dans des environnements complexes.
- Composition dynamique des flux de travail par gestion du profil du groupe et de la collaboration.
- Evaluation du système dans les applications réelles à des projets effectués par des sociétés de conception.
- Moyens intelligents de génération répartie au cours de la conception collaborative.
- Compréhension et évaluation intelligentes des solutions générées par l'ensemble de concepteurs.

Ces deux derniers points complèteront dans l'avenir l'intégration totale d'un mécanisme collaboratif dans le cycle de la conception collaborative (Chapitre 2)

Un autre effort pourra être aussi accompli pour développer davantage l'espace de travail du client, afin d'y incorporer différents modèles collaboratifs supplémentaires, comme des descriptions formelles basées sur le langage. De plus, le prototype Enhanced-CDMS ne garde aucun enregistrement des bases de données concernant les actions des concepteurs et ne peut ainsi les dépister. De ce fait, il est

impossible de corriger les erreurs possibles. Il faudra également développer un système pour couvrir et soutenir efficacement l'activité d'intégration du modèle final pour la phase de description déclarative de la scène, en tenant compte des préférences des co-concepteurs et de la politique globale du projet.

Enfin, une étape suivante de ce travail de recherche inclut une évaluation du système Enhanced-CDMS dans les applications à des projets effectués par des sociétés de conception.

Index of Abbreviation

AEC	Architecture, Engineering and Construction
ASP	Application Service Provider
BRep	Boundary Representation
CA	Collaborative Administration
CAAD	Computer-Aided Architectural Design
CAAO	Conception Architecturale Assistée par Ordinateur
CABA	Critical Areas Blockade Algorithm
CAD	Computer-Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAM	Collaborative Administration Mechanism
CAO	Conception Assistée par Ordinateur
CBM	Collaborative Base module
CDAO	Conception Déclarative Assistée par l'Ordinateur
CDM	Collaborative Declarative Module
CDM	Collaborative Declarative module
CDMS	Collaborative Declarative Modelling System
CDP	Collaborative Product Development
CER	Collaborative Event Recorder
CERM	Collaborative Event Recorder Mechanism
CGI	Common Gateway Interface
CLP(FD)	Constraint Logic Programming in Finite Domain
CSG	Constructive Solid Geometry
CSG	Constructive Solid Geometry
CSM	Collaborative Solid Modelling
CSP	Constraint Satisfaction Problem
CSP	Constraint Satisfaction Problem
DKABM	Declarative Knowledge for Architecture Building Modelling
DM	Declarative Modelling
DMHD	Declarative Modelling by Hierarchical Decomposition
DXF	Data eXchange Format

Index of Abbreviations

EE-R	Extended Entity-Relationship
FAO	Fabrication Assistée par Ordinateur
FBM	Feature-Based Modelling
FBM	Feature-based modellers
FCFS	First Come First Served
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	HyperText Transfer Protocol
IAO	Ingénierie Assistée par Ordinateur
IDR	Internal Declarative Representation
IMD	Internal Model Description
ISO	International Standards Organization
IT	Information technology
KAU	Kernel Administration Unit
MD	Modélisation Déclarative
MDDH	Modélisation Déclarative par Décomposition Hiérarchique
MDM	MultiCAD Designer module
MG	Modélisation Géométrique
NDBM	Normalized Declarative Building Model
PDM	Product Data Management Systems
PHP	Personal Home Page – Hypertext Preprocessor
SQL	Standard Query Language
STEP	STandard for the Exchange of Product model data
UML	Unified Modelling Language
VPNs	Virtual Private Networks
VRML	Virtual Reality Modelling Language
WAU	Workflow Administration Unit
WBC CAAD	Web Based Collaborative CAAD
WWW	World Wide Web
XML	eXtensible Markup Language

Bibliography

- [Alibre 05] Alibre Inc., Alibre Design™ Software, Richardson TX, USA. www.alibre.com, 2005
- [Anupam et al. 94] Anupam V. et Bajaj C., “SHASTRA – An Architecture for Development of Collaboration Applications”, International Journal of Intelligent and Cooperative Information Systems, p. 155-172, 1994
- [Autodesk 05] Autodesk Inc. www.autodesk.com
- [Bajaj et al. 99] Bajaj C., Cutchin S., Morgia C., Paoluzzi A. et Pascucci V., “Web Based Collaborative CAAD” (poster), Fifth ACM Symposium on Solid Modeling and Applications, Ann Arbor, Michigan, USA, p. 326-327, 1999
- [Bidarra et al. 00] Bidarra R., Bronsvoort W.,F., “Semantic feature modelling”, Computer-Aided Design 32, 201–225, 2000
- [Bidarra et al. 01a] Bidarra, R., Van Den Berg, E., and Bronsvoort, W. F., “Interactive Facilities for Collaborative Feature Modeling on the Web”, Proc. of the Tenth Portuguese Conference on Computer Graphics, Lisbon, Portugal, pp. 43–52, 2001
- [Bidarra et al. 01b] Bidarra, R., Van Den Berg, E., and Bronsvoort, W. F., “Web-based Collaborative Feature Modeling,” Proc. of '01-Sixth Symposium on Solid Modeling and Applications, pp. 319–320, 2001
- [Bidarra et al. 02a] Bidarra R, Kranendonk N, Noort A, Bronsvoort W.,F. “A collaborative framework for integrated part and assembly modelling”. J Comput Inf Sci Eng; 2(4):256–64, 2002
- [Bidarra et al. 02b] Bidarra R, van den Berg E, Bronsvoort W.F. “Collaborative feature modeling systems. J Comput Inf

Bibliography

- Sci Eng; 2(3), p.192–8, 2002
- [Bonnefoi 99] Bonnefoi P. F., “Techniques de satisfaction de contraintes pour la modélisation déclarative. Application à la génération concurrente de scènes”. PhD thesis, Limoges, France, 1999
- [Case et al. 96] Case M.P., Lu S.C.Y., “Discourse model for collaborative design”. *Computer Aided Design* 28, 333–45. 1996
- [Champiaux 98] Champiaux L., “Introduction de techniques d'apprentissage en modélisation déclarative”. PhD thesis, Ecole des Mines de Nantes, Nantes France, 1998
- [Chan et al. 99] Chan S., et Wong M., “Collaborative Solid Modelling on the WWW”, *Proceedings of the 1999 ACM Symposium on Applied Computing*, San Antonio, Texas, USA, p. 598-602, 1999
- [Chen et al. 04] Chen L, Song Z.J., Feng L. “Internet-based real-time collaborative assembly modeling via an e-assembly system: status and promise”. *Computer Aided Design*; 36:835–47, 2004
- [CoCreate 01] CoCreate Software Inc., , OneSpace Designer™ Software, Fort Collins, CO. <http://www.cocreate.com>, 2001
- [CAD Group 06] CAD Group, National Informatics Centre, Department of IT, Ministry of Communications and Information and Technology, India. www.collabcad.com, 2006
- [Colin 97] Colin C., Desmontils E., Martin J. Y., Mounier J. P., “Working modes with declarative modeller”, *Compugraphics'97*, 1997
- [Colin et al. 98] Colin C., Desmontils E., Martin J.-Y., Mounier J.-P., “Working modes with a declarative modeler” *Computer Networks and ISDN Systems* Volume 30, Issues 20-21 , Pages 1875-1886, 1998
- [Cross 96] Cross N., Christiaans H., Dorst K., (eds.), “Analyzing

Bibliography

- Design activity”, J. Wiley & Sons, 1996
- [Cross et al 95] Cross N., Cross A.C., “Observations of teamwork and social processes in design”, *Design Studies*, Volume 16, p.143–170. 1995
- [Desmontils 95] Desmontils E. “Les modeleurs déclaratifs”. Rapport de recherche, Université de Nantes, 1995
- [Dassault Systemes 05] Dassault Systemes, www.3ds.com/corporate , 2005
- [Dietrich t al. 97] Dietrich U., von Lukas U., et Morche I., « Cooperative modeling with TOBACO », *Proceedings of the TeamCAD '97 Workshop on Collaborative Design*, Atlanta, USA, p. 115-122, 1997
- [Dragonas et al 05] Dragonas J., Makris D., Lazaridis A., Miaoulis G., Plemenos D. “Implementation of Collaborative environment in MultiCAD Declarative Modelling System“ 8th International Conference 3IA 2005, Limoges France, 2005.
- [Dragonas et al 06] Dragonas J., Lazaridis A., Makris D., Miaoulis G., Plemenos D. “Towards Project Coordination Environment for Collaborative Declarative Scene Modeling” 9th International Conference 3IA 2006, Limoges, France, 2006
- [Eastman 99] Eastman C. M., “Building Product Models: computer environments supporting design and construction”, CRC Press, Florida, USA, 1999
- [Fribault 03] Fribault P., “Modélisation Declarative d’Espaces Habitable”, (in French), PhD thesis, University of Limoges, France 2003
- [Fuha et al 05] Fuha J.Y.H., Lib W.D. “Advances in collaborative CAD: the-state-of-the art”, *Computer-Aided Design*, Volume 37 p.571–581, 2005
- [Gaildrat 03] Gaildrat V., “Modélisation déclarative d'environnements virtuels : Création de scènes et de

Bibliography

- formes complexes par l'énoncé de propriétés et l'emploi d'interactions gestuelles”. Habilitation à diriger des recherches, Université Paul Sabatier, Toulouse, janvier 2003
- [Gero et al 98] Gero, J. S., McNeill, T., “An approach to the analysis of design protocols”, *Design Studies* Volume 19 Issue 1, p. 21-61, 1998
- [Golfinopolulos et al 04] Golfinopoulos V, Dragonas J, Miaoulis G Plemenos D., “Declarative design in collaborative environment”, 7th International Conference 3IA' 2004, Limoges (France), May 2004
- [ISO 92] ISO TC184/SC4/WG4 N34 P5, “Guidelines for the Development and Approval of STEP Application Protocols”, 1992
- [ImpactXoft 05] www.impactxoft.com, 2005
- [Kamada et all 88] Kamada T., Kawai S., A Simple Method for Computing General Position in Displaying Three-dimensional Objects, *Computer Vision, Graphics and Imaging Processing*, 41, 1988
- [Kochhar 94] Kochhar, S., “CCAD: A Paradigm for Human-Computer Cooperation in Design” *IEEE Computer Graphics and Applications*, vol. 14, no. 3, pp. 54-65, May/Jun, 1994
- [Kvan 00] Kvan, T., “Collaborative design: What is it ?”, *Automation in Construction*, Volume 9, Issue 4, p. 409–415, 2000
- [Kwaiter 98] Kwaiter G., “Modélisation déclarative de scènes: étude et réalisation de solveurs de contraintes”, PhD. Thesis, Institute of Research in Data processing of Toulouse, France, 1998
- [Lee et al. 99] Lee Jae Yeol, Han Sung Bae, Kim Hyun et Park Sang Bong, “Network-Centric Feature-Based Modeling”,

Bibliography

- Seventh Pacific Conference on Computer Graphics and Applications (Pacific Graphics '99), Seoul, Korea, October 1999
- [Li et al. 04] Li W.D., Ong S.K., Fuh J.Y.H., Wong Y.S., Lu Y.Q., Nee A.Y.C., "Feature-based design in a collaborative and distributed environment". *Computer Aided Design* 36:775–97, 2004
- [Li et al. 05] W.D. Li , W.F. Lu, J.Y.H. Fuh, Y.S. Wong, "Collaborative computer-aided design—research and development status", *Computer-Aided Design* 37, 931-940, 2005
- [Liege 96] Liège S., "La modélisation déclarative incrémentale: Application à la conception urbaine", Thèse de doctorat, Nantes, décembre 1996
- [Liege et al 97] Liege S., Hegron G., "An Incremental Declarative Modelling applied to Urban Layout Design", Fifth International Conference WSCG'97, Plzen (Czech Republic), February 1997
- [Lucas et al 90] Lucas M., Martin D., Martin P., Plemenos D., "The ExploFormes project: some steps towards declarative modelling of forms", (in french). *Journées AFCET-GROPLAN*, Strasbourg (France), BIGRE, no 67, pp 35 – 49, January 1990
- [Lucas et al 95] Lucas M., Desmontils, E., "Les Modeleurs Declaratifs Declarative modellers", *Rev. Int. CFAO Inf. Graph.* 10 6, 559–585, 1995
- [Maher et al 98] Maher M. L., Cicognani A., Simoff S. J., "An experimental study of computer mediated collaborative design", *Int. J. Des. Comput.* 1, 06, June 1998
- [Makris et al 03] Makris D., Ravani I., Miaoulis G., Skourlas C., Fribault P., Plemenos D., "Towards a domain-specific knowledge intelligent information system for Computer-Aided Architectural Design" 6th 3IA'03,

Bibliography

- Limoges, France, 2003
- [Makris 05] Makris D., “Etude et réalisation d'un système déclaratif de modélisation et de génération de styles par algorithmes génétiques. Application à la création architecturale”, PhD thesis, University of Limoges, 2005
- [Martin et al. 88] Martin P., Martin D., “ An expert system for polyhedra modeling”, Eurographics'88, Nice, France, 1988
- [Miaoulis 02] Miaoulis G., “Contribution à l'étude des Systèmes d'Information Multimédia et Intelligent dédiés à la Conception Déclarative Assistée par l'Ordinateur – Le projet MultiCAD” (in French), Thèse de Doctorat, University of Limoges, France 2002
- [Miaoulis et al 96] Miaoulis G., Plemenos D., “Propositions pour un système d'information multimédia intelligent dédié à la CAO – Le projet MultiCAD”, Rapport de recherche MSI 96-03, Université de Limoges, France, 1996
- [Molina et al 95] Molina, A. Al-Ashaab, T. Ellis, R. Young, “A review of computer-aided simultaneous engineering systems”, Research in Engineering Design 7, 38–63, 1995
- [Mori et al. 98] Mori T, Cutkosky M.R. “Agent-based collaborative design of parts in assembly”. Proceedings of 1998 ASME design engineering technical conferences, [DETC98/CIE-5697]. Atlanta, GA, USA, 1998
- [Mounier 98a] Mounier J.-P., “The use of genetic algorithms for path planning in the field of declarative modelling”, 3IA'98, Limoges, France, April 1998
- [Mounier 98b] Mounier J.-P., “Le projet UrbaFormes : modélisation déclarative de parcours pour la prise de connaissance d'un environnement architectural ou urbain”, Thèse de Doctorat, Nantes, 10 juillet 1998
- [Nam et al. 01] Nam Tek-Jin et Wright David K., “The Development and Evaluation of Syco3D: A Real-Time Collaborative

Bibliography

- 3D CAD System”, Design Studies, vol. 22, n_ 6, p. 557-582, November 2001
- [Nam et al. 98] Nam Tek-Jin, et Wright D., K., “CollIDE: A Shared 3D Workspace for CAD”, Proceedings of the Fourth EATA International Conference on Networking Entities, Leeds, United Kingdom, October 1998
- [Pajot-Duval 94] Pajot-Duval L. “Modélisation déclarative de configurations de segments de droite : application aux tableaux de fils”. Infographie Interactive et Intelligence Artificielle 3IA'94, Limoges, 1994
- [Plemenos 02] Plemenos D., Miaoulis G., Vassilas N., “Machine learning for a General Purpose Declarative Scene Modeller”. Intern. Conference GraphiCon 2002, Nizhny Novgorod, Russia 2002
- [Plemenos 91] Plemenos D., “Contribution to the study and development of techniques of modelling, generation and display of scenes. The MultiFormes project”, Professorial dissertation, (in French), Nantes, France, 1991
- [Plemenos 93] Plemenos D, “Techniques for implementing learning mechanisms in a hierarchical declarative modeller”. Research report MSI 93 - 04, (in French), Limoges France, 1993
- [Plemenos 95] Plemenos D., “Declarative Modelling by Hierarchical Decomposition. The actual state of the Multiform Project”, Int. Conf. GraphiCon' 95, St Petersburg, Russia, 1995
- [Plemenos et al 96] Plemenos D., Benayada M., “Intelligent Display in Scene Modeling. New Techniques to Automatically Compute Good Views”, Graphicon'96, St Petersburg (Russia), 1-5 july, 1996
- [Poulet 94] Poulet F. “Modélisation déclarative de scènes

Bibliography

- tridimensionnelles: le projet SpatioFormes”.
Infographie Interactive et Intelligence Artificielle
3IA'94, Limoges, 1994
- [Poulet et al. 96] Poulet F., Lucas M., “Modeling megalithic Sites”,
Proceeding of Eurographics'96, pp.279-288, Poitiers,
France, 1996
- [PTC 01] PTC, DIVISON™ Software, Needham, MA.
<http://www.ptc.com/products/division/index.htm>, 2001
- [Ravani et al 03] Ravani I., Makris D., Miaoulis G., Constantinides P.,
Petridis A., Plemenos D., “Implementation of
Architecture-oriented Knowledge Framework in
MultiCAD Declarative Scene Modelling System”, 1st
Balcan Conference in Informatics, Thessaloniki,
Greece, 2003
- [Rodriguez et al. 02] Rodriguez K., Al-Ashaab A., “A review of internet
based collaborative product development systems”,
International Conference on Concurrent Engineering:
Research and Applications, Canfield, United Kingdom,
2002
- [Rodriguez et al. 05] Rodriguez K., Al-Ashaab A., “Knowledge web-based
system architecture for collaborative product
development”, Computers in Industry 56, p.125–140,
2005
- [Roseman et al. 92] Roseman M., Greenberg S., «GroupKit: A Groupware
Toolkit for Building Real-Time Conferencing
Applications», Proceedings of the ACM Conference on
Computer Supported Cooperative Work, Toronto,
Ontario, USA, , p. 43-50, 1992
- [Ruchaud 01] Ruchaud W., “Etude et réalisation d’un moteur de
résolution de contraintes géométriques pour la
modélisation déclarative”, PhD Thesis, (in French),
Limoges, 2001
- [Sellinger 98] Sellinger D. “ La modalisation géométrique déclarative

Bibliography

- interactive. Le couplage d'un modeleur déclaratif et d'un modeleur classique”, PhD Thesis, (in French), Limoges, 1998
- [Shah88] Shah J. J., Rogers M. T., “Expert Form Feature modelling Shell”, *Computer Aided Design*, vol. 20, n° 9, 1988
- [Shyamsundar et al. 02] Shyamsundar N, Gadh R. Collaborative virtual prototyping of product assemblies over the Internet. *Computer Aided Design*;34:755–68, 2002
- [Stork et al. 97] Stork A., et Jasnoch U., “A Collaborative Engineering Environment”, *Proceedings of the TeamCAD '97 Workshop on Collaborative Design*, Atlanta, USA, May, p. 25-33, 1997
- [Stork et al. 98] Stork A., von Lukas U., et Schultz R., “Enhancing a Commercial 3D CAD System by CSCW Functionality for Enabling Cooperative Modelling via WAN”, *Proceedings of the ASME Design Engineering Technical Conferences*, Atlanta, USA, September 1998.
- [Tay at al 03] Tay F. E. H, Roy A. “CyberCAD: a collaborative approach in 3D-CAD technology in a multimedia-supported environment”. *Computers in Industry*; 52:127–45, 2003
- [van den Berg 00a] van den Berg E., “Web-based collaborative modelling with Spiff”, MSc Thesis, Delft University of Technology, 2000
- [van den Berg 00b] van den Berg, E., Bidarra R. et Bronsvooort W. F., “Web-based Interaction on Feature Models”, *Proceedings of the Seventh IFIP WG 5.2 Workshop on Geometric Modelling: Fundamentals and Applications*, Parme, Italie, , p. 113-123, October 2000
- [van den Berg 99] van den Berg E., “Collaborative Modelling Systems”, TU Delft, Netherlands, October 1999
- [van den Berg et al 00] Van den Berg E, Bidarra R, Bronsvooort WF. “Web-

Bibliography

- based interaction on feature models”. Proceedings of seventh IFIP WG 5.2 workshop on geometric modelling: fundamentals and applications, p. 319–20. Parma, Italy, 2000
- [Vassilas et al 02] Vassilas V., Miaoulis G., Chronopoulos D., Konstantinidis E., Ravani I., Makris D., Plemenos D., “MultiCAD-GA: a system for the design of 3D forms based on genetic algorithms and human evaluation”, Lectures Notes on Artificial Intelligence (LNAI 2308), Springer-Verlag, p 203-214, 2002
- [von Lukas 99] von Lukas Uwe, “Synchronous Tools for Concurrent Engineering Based on CORBA”, Sixth European Concurrent Engineering Conference, Erlangen-Nuremberg, Allemagne, , p. 223-227, 1999
- [Wang et al. 02] Wang L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A., “Collaborative conceptual design – state of the art and future trends”, Computer-Aided Design, vol.34, pp.981-996, 2002
- [Webscope 01] Webscope Inc., Webscope™ Software, Sunnyvale, CA. <http://www.webscopeinc.com/products/features.html>, 2001

Bibliography