



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Prise en compte de situations de coopération non prédéfinies dans un système de workflow

Poos, William

Award date:
1994

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés universitaires Notre-Dame de la Paix
Institut d'informatique
Rue Grandgagnage, 21 - Namur

**Prise en compte de situations
de coopération non prédéfinies
dans un système de workflow**

Directeur de mémoire :

M^r F. Bodart

? Ch 2: la tentative de
modélisation est elle
propre à l'étudiant ou pas?

Mémoire présenté par William Poos
en vue de l'obtention du grade de
Licencié et Maître en Informatique

Année académique 1993-1994

*Nous remercions Monsieur F. Bodart
et Madame T. PetitJean qui nous ont dirigé tout
au long de la réalisation de ce travail*

Table des matières

Introduction.....	2
Chapitre 1 :La problématique des S.I.A.C. (système informatique d'aide à la coopération)	4
1.1. Introduction.....	4
1.2. Définitions	5
1.3. Histoire organisationnelle	5
1.3.1. La situation après la seconde guerre mondiale.....	5
1.3.2. Les années 70-90.....	6
1.3.2.1. La demande changeante des utilisateurs.....	6
1.3.2.2. Les innovations technologiques	6
1.3.2.3. La restructuration des organisations.....	7
1.4. Les systèmes de gestion de la coopération.....	11
1.4.1. La coopération [PETIT 93][SCHAEL 93].....	11
1.4.2. Supports technologiques de la gestion de la coopération.....	12
1.4.2.1. Support de la coopération collégiale	12
1.4.2.2. Support à la coopération répartie.....	13
1.5. Evolution attendue	14
1.5.1. Les supports de la coopération répartie.....	14
1.5.2. Les supports de la coopération collégiale	15
Chapitre 2 : WoorKS	16
2.1. La connaissance générique de WoorKs	17
2.1.1. Le modèle organisationnel [Ader 92 (1)].....	17
2.1.1.1. Introduction	17
2.1.1.2. La structure organisationnelle	18
2.1.1.3. Les compétences organisationnelles.....	22
2.1.1.4. Représentation de l'environnement	24
2.1.2. Le modèle des références [ADER 92 (5)] [ADER 92 (8)].....	25
2.1.2.1. Introduction	25
2.1.2.2. Définition	25
2.1.2.3. Application	26
2.1.3. Le modèle de l'information [ADER 90][ADER 92 (7)]	26
2.1.3.1. Introduction	27

2.1.3.2. Définitions.....	27
2.1.3.3. Application.....	28
2.1.4. Le modèle des opérateurs.....	28
2.1.4.1. Introduction.....	28
2.1.4.2. Définition.....	29
2.1.4.3. Typologie.....	33
2.1.5. Le modèle du temps [ADER 90][SRIVA 92 (2)][SRIVA 92 (3)].....	36
2.1.5.1. La représentation du temps.....	36
2.1.5.2. Le calendrier.....	37
2.1.5.3. L'agenda.....	38
2.1.5.4. La gestion des événements.....	38
2.1.6. Le modèle des procédures.....	40
2.1.6.1. Introduction.....	40
2.1.6.2. Procédure [MOLLE 93].....	41
2.1.6.3. Activité[MOLLE 93] [ADER 92 (6)].....	47
2.1.6.4. Représentation E/A de la sémantique des procédures et des activités.....	51
2.1.7. La description des workflows.....	53
2.1.7.1. Fonctionnalités offertes par le modèle de l'organisation.....	53
2.1.7.2. Fonctionnalités offertes par le modèle des opérateurs.....	54
2.1.7.3. Fonctionnalités offertes par le modèle du temps.....	54
2.1.7.4. Fonctionnalités offertes par le modèle des procédures.....	54
2.1.8. Le méta-modèle.....	63
2.1.8.1. Définition des composants.....	63
2.1.8.2. Architecture du modèle.....	65
2.1.8.3. Les mécanismes génériques.....	66
2.2. L'environnement logiciel de WoorKs.....	68
2.2.1. Les fonctions des acteurs.....	69
2.2.1.1. L'ingénieur d'application et le développeur d'application[LU 93].....	69
2.2.1.2. Les utilisateurs d'un workflow particulier.....	70
2.2.2. Les supports de la réalisation des fonctions des acteurs.....	70
2.2.2.1. Les outils de gestion de la connaissance générique.....	70
2.2.2.2. Les outils de mise en oeuvre de la connaissance générique.....	71
2.2.2.3. Les outils d'exécution d'un workflow.....	71
2.3. Le processus de développement.....	72
2.3.1. Introduction.....	72
2.3.2. Première étape : Identification du problème.....	73

2.3.2.1. Identification des besoins du client	74
2.3.2.2. Spécification du traitement.....	74
2.3.3. Deuxième étape : Utilisation de la connaissance générique	75
2.3.3.1. Utilisation du modèle organisationnel.....	75
2.3.3.2. Utilisation du modèle informationnel	75
2.3.3.3. Utilisation du modèle du temps.....	76
2.3.3.4. Utilisation du modèle des procédures	76
2.3.3.5. Utilisation du modèle des opérateurs et du méta- modèle	76
 Chapitre 3 : ACTA.....	 79
3.1. Introduction.....	79
3.2. L'objectif	79
3.2.1. Coopération répartie.....	80
3.2.1.1. La coordination externe.....	81
3.2.1.2. La coordination interne	81
3.2.2. La coopération répartie et ad hoc	82
3.2.2.1. Les activités de construction de la situation de coopération	82
3.2.2.2. Les activités de gestion de la situation de coopération.....	83
3.3. Les moyens	83
3.3.1. Les idées.....	84
3.3.1.1. Le modèle descriptif de la situation de coopération.....	84
3.3.1.2. Conception des mécanismes d'interaction.....	85
3.3.2. ACTA : un modèle de la coopération logistique.....	85
3.3.2.1. Le modèle d'organisation des tâches	86
3.3.2.2. Le modèle des acteurs	94
3.3.2.3. Le modèle des conversations.....	96
 Chapitre 4 : Comparaison d'ACTA et de WoorKS.....	 98
4.1. Introduction.....	98
4.2. Les domaines d'application	99
4.3. Les processus de développement	99
4.3.1. Préambule.....	99
4.3.2. Le processus de développement propre à WoorKS	100
4.3.3. Le processus de développement propre à ACTA.....	101
4.3.3.1. Introduction	101

4.3.3.2. Simplicité du processus de développement.....	102
4.3.3.3. Flexibilité d'ACTA.....	103
4.4. Représentations de l'activité.....	105
4.4.1. Introduction.....	105
4.4.2. Les cycles de vie.....	105
4.4.2.1. Rappel.....	105
4.4.2.2. Comparaison.....	106
4.5 Représentations de l'organisation.....	107
4.5.1. ACTA.....	107
4.5.2. WoorKs.....	108
4.6. Répartition du travail au sein de l'organisation.....	108
4.6.1. WoorKs.....	108
4.6.2. ACTA.....	109
 Chapitre 5 : Intégration d'ACTA et de WoorKs.....	 110
5.1. Introduction.....	110
5.2. Choix d'implémentation pour ACTA.....	111
5.2.1. Introduction.....	111
5.2.2. Représentation des informations.....	111
5.2.3. La corbeille.....	113
5.2.3.1. Définition.....	113
5.2.3.2. Fonctionnalités.....	113
5.2.4. L'outil de spécification et de gestion des situations de coopération.....	114
5.2.4.1. Définition.....	114
5.2.4.2. Fonctionnalités.....	115
5.2.5. L'outil de réalisation des tâches.....	116
5.2.5.1. Définition.....	116
5.2.5.2. Fonctionnalités.....	116
5.2.6. Le gestionnaire du modèle de l'organisation.....	117
5.2.7. Le système de communication [BIGNO 90][AGOS 91].....	118
5.2.7.1. Introduction.....	118
5.2.7.2. Interfaçage entre ACTA et UTUCS.....	119
5.3. Interfaçage d'ACTA et de WoorKs.....	121
5.3.1. Introduction.....	121
5.3.2. Extension de WoorKs.....	122
5.3.2.1. Approche générale.....	122
5.3.2.2. Présentation approfondie.....	125

5.4. Implémentation d'ACTA-WoorKs.....	138
5.4.1. Introduction	138
5.4.2. Utilisation des modèles de WoorKs.....	139
5.4.2.1. Utilisation du modèle organisationnel.....	139
5.4.2.2. Utilisation du modèle des références.....	140
5.4.2.3. Utilisation du modèle de l'information.....	140
5.4.2.4. Utilisation du modèle des opérateurs	140
5.4.2.5. Utilisation du modèle du temps.....	141
5.4.2.6. Utilisation du modèle des procédures	141
5.4.2.7. Utilisation des objets T.O.I	143
5.4.2.8. Utilisation du méta-modèle	143
5.4.3. Les nouveaux modèles	144
5.4.3.1. Le modèle des plans de réalisation	144
5.4.3.2. Le système de communication	144
Conclusion	145
Bibliographie	146

Introduction

Depuis leur apparition, les systèmes informatiques supportant les situations de coopération répartie n'ont cessé de se développer. Au départ, l'effort de recherche s'est centré sur les supports des situations de coopération répartie prédéfinie. Vu l'intérêt croissant porté à ces systèmes par le monde de l'entreprise, l'idée de les réinvestir dans le cadre des situations de coopération répartie ad hoc a vu le jour.

Nous avons donc voulu aller plus avant dans l'analyse de cet état de choses, en l'abordant de manière plus systématique. Dans cette optique, nous tenterons de mettre sur pied un système intégré supportant les situations de coopération répartie prédéfinie et ad hoc.

Pour ce faire, nous débuterons cette recherche en rappelant les particularités de ces deux types de coopération et en considérant, pour l'un et l'autre, les outils informatiques existants. Cela constituera le premier chapitre qui sera consacré à la problématique des systèmes informatiques d'aide à la coopération.

Suite aux constats posés dans cette première partie, nous envisagerons l'élaboration du système intégré comme une extension d'un support aux situations de coopération répartie prédéfinie. WoorKs s'est avéré particulièrement bien adapté pour servir de base à l'élaboration de celui-ci. Le deuxième chapitre sera donc consacré à la présentation des éléments de WoorKs.

Par ailleurs, des recherches menées dans le cadre des supports aux situations de coopération répartie ad hoc ont donné naissance au modèle ACTA qui a été choisi pour représenter la composante ad hoc du système intégré. Nous considérerons les différents aspects de ce modèle dans le troisième chapitre.

WoorKs et ACTA ont, l'un et l'autre, comme domaine d'application, les situations de coopération répartie. En ce sens, ils partagent certains concepts. Néanmoins, WoorKs et ACTA intéressent des domaines différents de spécialisation, à savoir les situations prédéfinies pour le premier et les situations ad hoc pour le second. Il est donc évident qu'ils se distinguent également l'un de l'autre au niveau de certains concepts. Ces

différences seront présentées dans le quatrième chapitre qui sera consacré à la comparaison de WoorKs et d'ACTA.

Les particularités de WoorKs et d'ACTA étant posées, nous pourrons alors envisager l'intégration d'ACTA à WoorKs de manière concrète. Cela signifie que l'on se situera à un niveau proche de l'implémentation (Chapitre 5 : Intégration d'ACTA à WoorKs).

Chapitre 1 :

La problématique des S.I.A.C. (1) système informatique d'aide à la coopération)

1.1. Introduction

L'apparition des S.I.A.C. est due à l'interaction de nombreux éléments. Nous ne prétendons pas en donner une liste exhaustive mais plutôt aborder ceux qui nous seront utiles pour essayer d'évaluer l'évolution future de ces systèmes.

Nous commencerons par définir quelques concepts de base tels que l'organisation, sa structure, la technologie A partir de ces derniers, nous aborderons l'évolution organisationnelle qui s'étendra de l'apparition du taylorisme jusqu'à l'apparition des "organisations de processus"[SCHA 93].

Suivra une présentation de l'état de l'art en matière d'outils de gestion de la coopération; ce compte rendu comprendra une définition, une typologie et une critique.

Et nous terminerons par la justification de ce mémoire en présentant la tendance à venir en matière de développement de ces systèmes. Pour dégager cette tendance, nous évaluerons le trou technologique actuel, fruit de l'évolution des besoins et des limites des systèmes disponibles.

1.2. Définitions

Une **organisation** est un ensemble de personnes artificiellement réunies pour produire ou prester un service de manière efficace dans un environnement donné. Ses membres emploient certaines *technologies* au sein d'une *structure* pour réaliser leurs tâches.

La **structure** répartit et coordonne le travail au sein de l'organisation.

La **technologie** est l'application concrète de connaissances scientifiques et/ou techniques à la conception, au développement et à la fabrication d'un produit et/ou d'un service, matériel ou immatériel.

1.3. Histoire organisationnelle

1.3.1. La situation après la seconde guerre mondiale

La structure organisationnelle en place à la fin de la seconde guerre mondiale est basée sur une forte division du travail. Cette division se fait par spécialisation horizontale et verticale.

La **spécialisation horizontale** touche le centre opérationnel : on diminue le niveau de compétence requis par travailleur en divisant le travail en petites tâches répétitives.

La **spécialisation verticale** sépare les tâches opérationnelles des tâches administratives. Poussée à l'extrême, on trouve d'un côté les gens du centre opérationnel qui n'ont rien à dire et de l'autre, ceux du top management qui décident.

Cette spécialisation est à la base de la vision fonctionnelle du travail. Ce dernier étant organisé sous forme de procédures qui s'effectuent pas à pas en suivant la définition. Les exceptions sont gérées par un des niveaux supérieurs suivant leurs responsabilités, compétences et autorité de décision.

Ce type de structure a un besoin énorme de coordination, tant entre les différents éléments du centre opérationnel qu'entre le niveau stratégique et le niveau opérationnel. Elle requiert l'existence d'un environnement stable dont l'évolution est prévisible et linéaire.

1.3.2. Les années 70-90

Ces dernières années se caractérisent par un grand nombre de changements. Les entreprises doivent notamment faire face à la demande changeante des clients, au rythme soutenu d'apparition des innovations technologiques, ...

L'énorme désavantage des organisations fonctionnelles est leur manque de flexibilité. Face à un environnement dont la stabilité s'effrite, elles se doivent d'entamer une reconversion qui assurera leur survie.

1.3.2.1. La demande changeante des utilisateurs

Ces dix dernières années, le pouvoir des clients sur les secteurs d'activités n'a cessé de croître. Les consommateurs accordent de plus en plus d'importance à la personnalisation des produits qu'ils achètent. Cette demande engendre une fragmentation du marché et une augmentation de sa complexité. Pour survivre, l'entreprise ne doit plus se contenter de fournir un produit standard bon marché mais le produit attendu par le client.

1.3.2.2. Les innovations technologiques

Les secteurs de l'information, des télécommunications et de la bureautique forment, jusqu'au début des années 80, trois secteurs d'activités distincts. Les évolutions technologiques telles que l'apparition des réseaux locaux, des traitements de textes ..., ont amorcé la fusion de ces trois secteurs d'activité (Fig. 1.1.).

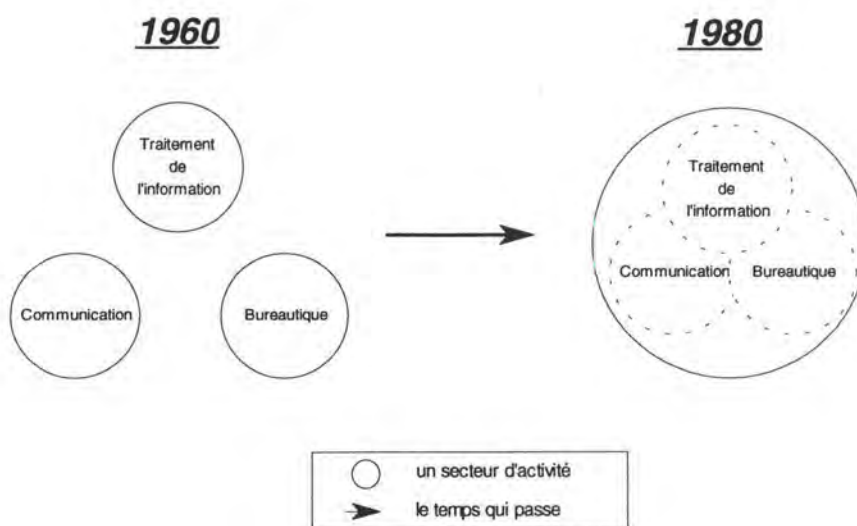


Fig. 1.1. Fusion des trois secteurs d'activités

L'organisation n'est plus un ensemble de petits postes de travail mais un ensemble de groupes de travail qui réalisent chacun un sous-objectif de l'organisation (Fig. 1.3.).

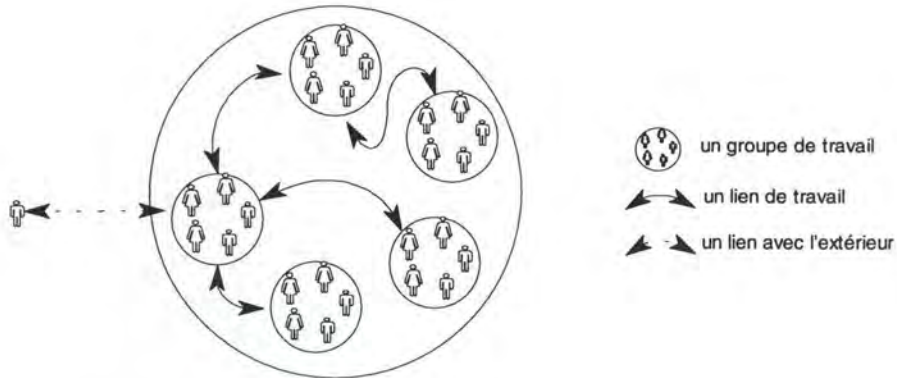


Fig. 1.3. Workflow dans une organisation de processus

Dans ce type d'organisation, on trouve trois types de processus, à savoir : matériel, informationnel et d'affaires.

Un **processus matériel** comprend des activités qui ont pour objectif de modifier le monde physique. Il s'agit de transporter, transformer, assembler, ...

Un **processus informationnel** comprend les informations qui ne peuvent être prise en compte par les processus matériels. Il s'agit d'utiliser les techniques de communication, les bases de données, les systèmes de contrôle de flux, etc. L'action de ces outils ne tient pas compte de la sémantique des messages transmis, cette dernière étant gérée par les processus d'affaires.

Un **processus d'affaires** est un échange d'information entre les acteurs de l'organisation qui influencera le déroulement futur de leurs activités.

Les processus d'affaires sont implémentés par les processus informationnels, eux-mêmes implémentés par les processus matériels.

Ex : Imaginons une société d'entrepreneurs. Le choix de leurs fournisseurs se fait en fonction de la qualité des marchandises, des délais de livraison et des prix. Il s'agit d'un processus d'affaires qui sera implémenté par l'envoi d'une liste de commandes (processus informationnel) qui aura pour effet

l'arrivée des marchandises dans le service d'entreposage (processus matériel).

Cette modélisation du travail redistribue les responsabilités parmi les gens qui effectuent réellement le travail. Les avantages sont :

- une *augmentation de la flexibilité* due à la capacité des groupes de réagir sans devoir recourir à un niveau supérieur de la ligne hiérarchique
- une *amélioration du sentiment de satisfaction* des employés qui effectuent les tâches du début à la fin en en assurant la responsabilité.

(b) Apport au développement des S.I.A.C.

Une des conséquences des organisations de processus est l'accroissement de l'intérêt accordé au concept de groupe coopératif.

Un **groupe coopératif** est un ensemble de personnes qui ont des engagements entre elles, envers d'autres groupes de l'organisation ou envers l'extérieur, dans le but d'atteindre les objectifs propres à leur mission. Il peut être *stable* dans le temps, c'est le cas d'un département, ou *temporaire* dans le cas d'une équipe de projet.

1.3.2.3.2. La fusion des îlots informationnels

Nous commencerons par présenter le rapprochement technologique des îlots. Ensuite, nous justifierons l'intégration de leur gestion et nous terminerons par l'apport de ces nouvelles technologies, ainsi que par l'apport de la fusion de leur gestion au développement des systèmes coopératifs.

(a) Rapprochement technologique

Le passage des années 70 aux années 80 est marqué par l'apparition d'innovations technologiques qui rapprochent les trois îlots. Les bases de ces innovations sont les L.A.N. et les bases de données partagées. Chacune d'elles va donner naissance à des applications qui sont les ancêtres des systèmes coopératifs.

(b) Fusion de la gestion

A partir des années 80, la plupart des organisations vont consolider au moins le contrôle politique et, peut-être, la gestion des trois îlots en un seul service d'information.

Cette fusion est due aux faits suivants :

- les décisions à prendre dans chacun des domaines impliquent de grandes sommes d'argent et des évaluations techniques et financières complexes ;
- les aptitudes de gestion de projet nécessaires au planning et à l'implémentation de telles applications sont similaires pour chacune des technologies ;
- plusieurs systèmes demandent la combinaison de ces technologies en un réseau pour effectuer du calcul, des télécommunications et de la bureautique de façon intégrée.

(c) Apport au développement des S.I.A.C.

Le rapprochement technologique offre les moyens techniques qui serviront au développement des outils coopératifs.

Les L.A.N. ont favorisé l'établissement de structures organisationnelles dans lesquelles on trouve un grand nombre de connexions entre les employés. Ces liens augmentent les possibilités de coopération.

Les bases de données partagées favorisent la collaboration entre les utilisateurs. Ils ont, par exemple, la possibilité de travailler ensemble à l'élaboration d'un document.

La fusion de la gestion des technologies de l'information améliore la coordination des efforts faits en matière d'utilisation des outils de support du travail des groupes coopératifs. C'est cette fusion qui permet de voir les technologies de l'information comme un tout plutôt que comme trois éléments distincts (bureautique, communication, traitement) qui doivent, tant bien que mal, essayer de cohabiter.

1.4. Les systèmes de gestion de la coopération

La modification de la structure organisationnelle et les innovations technologiques ont engendré des systèmes de gestion de la coopération. Nous allons définir le concept de coopération et en donner une typologie pour ensuite voir quels sont les outils que l'on peut employer pour les supporter.

1.4.1. La coopération [PETIT 93][SCHAEEL 93]

La **coopération** est l'action de participer à une oeuvre commune. Elle peut être soit répartie, soit collégiale.

La **coopération** est dite **répartie** lorsque l'oeuvre commune, un processus, est divisée en sous-tâches affectées à des acteurs ou des groupes d'acteurs appartenant au groupe coopératif. A la fin du processus, on peut clairement identifier la contribution de chacun.

La **coopération** est dite **collégiale** lorsque les individus doivent travailler ensemble dans le but d'atteindre un objectif commun. La réussite dépend de l'importance du partage de l'information entre les acteurs du groupe. A la fin du processus, la contribution individuelle de chacun à la réalisation de l'objectif ne peut être identifiée.

Ces deux types de coopération peuvent être soit prédéfinis, soit ad hoc.

Une situation de coopération est dite **prédéfinie** si elle est entièrement définie avant le commencement de la réalisation du processus. Dans le cas d'une coopération répartie, cette définition comprend la division du processus principal en sous-processus et, pour chacun d'eux, la définition de la manière dont il doit être réalisé. Dans le cas de la coopération collégiale, cette définition comprend l'ensemble des règles qui géreront le fonctionnement du groupe.

Une situation de coopération est dite **ad hoc** si elle n'est pas entièrement définie avant le commencement de la réalisation du processus. Dans le cas de la coopération répartie, on pourra définir la division et la réalisation du sous-processus en cours d'exécution du processus principal. Dans le cas de la coopération collégiale, les règles qui régissent le groupe pourront évoluer au cours de la réalisation du processus principal.

1.4.2. Supports technologiques de la gestion de la coopération

1.4.2.1. Support de la coopération collégiale

1.4.2.1.1. Présentation

Les supports de la coopération collégiale sont destinés à accroître le nombre et la qualité des contacts entre les différents membres du groupe coopératif. Ce groupe poursuit un objectif commun indivisible, ou du moins pour lequel la collaboration de tous est souhaitable. Cet objectif est le fruit de la collaboration de chacun. Le système devra veiller à ce que chacun soit dans les meilleures conditions pour participer à la réalisation de cet objectif. Les préoccupations principales seront de mettre l'information à la disposition de tous les acteurs concernés, de permettre à tous d'apporter leur contribution et de stimuler le cheminement vers l'objectif en comptabilisant les avancées et les reculs.

Il s'étend du simple courrier électronique en réseau local jusqu'au système intelligent de réunion électronique. Il s'agit de tous les outils qui favorisent la communication et la collaboration entre les travailleurs. Ils comprennent notamment les fonctionnalités suivantes :

- courrier électronique ;
- calendriers et outils de rendez-vous commun ;
- des réunions électroniques audiovisuelles qui permettent, par exemple, à plusieurs personnes de travailler, au même moment, à l'élaboration d'un document ;
- ...

1.4.2.1.2. Evaluation et critique

Cette dernière préoccupation est essentielle pour un groupe de travail: il faut à tout moment savoir où on en est et essayer d'évaluer le chemin qu'il reste à parcourir. Cela ne peut être pris en compte par le système que si on identifie les éléments contribuant à la réalisation de l'objectif. Il faut formaliser, si possible le cheminement ou un élément représentatif du cheminement vers l'objectif.

Ex de collégiale / auto / problème

1.4.2.2. Support à la coopération répartie

1.4.2.2.1. Présentation

Le problème principal est la synchronisation des personnes, des actions et la consistance des actions individuelles au processus en entier. Les gens réalisent séparément des morceaux de la tâche globale. Il faut offrir un outil qui permette de définir la division de la tâche globale et de répartir le travail au sein de l'organisation et d'en contrôler l'exécution en fournissant des informations qui renseignent les responsables sur le déroulement de cette tâche.

Son but est le contrôle du flux d'informations qui circulent entre les objets de l'organisation. Parmi ces objets, on peut citer les acteurs (l'ensemble des personnes employées par l'organisation), le matériel (imprimantes, armoires à bandes, ...) et les logiciels d'application (traitements de texte, tableurs, ...). Parmi les informations se trouvent des informations directement liées à la tâche : ce sont les commandes, les bon de livraisons, etc. Il existe, par ailleurs, des informations de gestion du workflow : ce sont les informations relatives à la définition et au contrôle du flux d'informations.

Il se compose donc des modules suivants : un module de description du flux qui permet de définir un méta-graphe de la situation de coopération, un module de l'organisation qui représente les acteurs qui vont faire réaliser le processus, un module d'application qui représente l'ensemble des outils utilisés, un module d'information qui représente les documents qui seront utilisés et un module d'exécution exécutant le flux défini par le module de description.

1.4.2.2.2. Evaluation et critique

Les supports à la coopération répartie, que l'on trouve actuellement sur le marché, visent les situations de coopération prédéfinie.

Ils sont basés sur un système de représentation des processus. Cette représentation comprend un méta-graphe qui définit la décomposition du processus principal en une suite ordonnée de sous-processus et, pour chacun d'eux, la définition de leurs entrées, de leurs sorties et des règles de transformation qui produisent les sorties à partir des entrées.

Cette formalisation du travail réalisé au sein d'un groupe coopératif peut, poussée à l'extrême, être le support de la taylorisation des entreprises de services. Or, le manque de flexibilité qui caractérise cette approche, les exigences du marché et les constantes

modifications technologiques² ont montré son insuffisance (supra 1.3.2.3. **La restructuration des organisations**).

En réaction, l'effort de recherche s'est déplacé vers les systèmes ad hoc que l'on commence à voir apparaître sur le marché. Comme le travail n'a pas été défini a priori, les utilisateurs désireux de bénéficier des fonctionnalités de gestion de la coopération telles que, par exemple, le contrôle des personnes auquel ils ont attribué une partie du travail, doivent passer préalablement par une définition 'on the fly' de la situation de coopération. Ce processus de définition devra être très facile à utiliser car il est destiné à l'ensemble des personnes qui auront le pouvoir de répartir le travail au sein de leur équipe.

1.5. Evolution attendue

1.5.1. Les supports de la coopération répartie

Tout nouveau système informatique n'a de chance de survie que s'il répond efficacement à une demande en provenance des organisations. Comme nous l'avons déjà dit, pour installer un S.I.A.C., il faut disposer d'un support technologique approprié. Le modèle de l'évolution de Nolan identifie cinq étapes révélatrices de l'état de l'informatisation des organisations [LESSU 94]. La première est une phase d'initiation, elle est caractérisée par le développement d'une gestion batch de la comptabilité. Fortes des résultats de la première étape, les entreprises accélèrent leurs dépenses pour propager l'innovation dans d'autres départements de la société; c'est l'étape de propagation. La troisième est un contrôle de la deuxième: on standardise ce qui s'est développé de manière désordonnée. Cette troisième étape est marquée par l'apparition des bases de données et du travail transactionnel. La quatrième comprend la généralisation des bases de données, une perception globale du management du phénomène informatique et surtout l'intégration des tâches opérationnelles, d'où l'apparition des systèmes de workflow. La cinquième est une étape de gestion des ressources qui se caractérise par l'informatisation des tâches de gestion et par une décentralisation, du matériel et des responsabilités. Les utilisateurs deviennent de plus en plus autonomes.

²Cette innovation technologiques concerne actuellement principalement le développement de logiciels. Ces logiciels sont les outils qui sont utilisés pour réaliser les tâches au sein de l'organisation. Si on veut profiter des innovations dans ce domaine, il faut éviter d'imposer aux employés l'utilisation de tel ou tel outil. Ils doivent avoir la possibilité de le choisir en fonction des besoins qu'ils rencontrent dans le cadre de la réalisation d'une tâche. Dans grand nombre de cas, la définition a priori ne peut être qu'une suggestion qui sera ou ne sera pas suivie.

On constate qu'une grande partie des organisations actuelles se situent à la quatrième étape du modèle [LESSU 94]; c'est l'étape d'intégration. Comme nous l'avons dit, elle se caractérise par l'intégration des tâches opérationnelles et l'apparition des systèmes de gestion de coopération prédéfinie et répartie [ADER 90]. Cela justifie l'effort de recherche centré sur ce type d'application. Les organisations les plus en avance sont en train de passer à la cinquième étape. Celle-ci comprend l'informatisation des tâches de gestion. Pour ce type de tâche, les systèmes prédéfinis sont insuffisants: les gestionnaires demandent des applications plus flexibles; ce sont les systèmes de gestion de la coopération répartie ad hoc.

Sont donc apparus un ensemble de systèmes qui répondent à cette demande [PETIT 94][BOGI 93][SWEN 93]. Ces systèmes sont presque exclusivement centrés sur la gestion de processus de coopération ad hoc. Or, la présentation dichotomique des situations de coopération, à savoir la coopération prédéfinie et ad hoc, est une simplification de la réalité. Les processus de coopération complexes regroupent ces deux types de coopération.

Le but de ce mémoire est de fusionner les deux types de support pour répondre à la demande réelle. Ce système comprendrait des fonctionnalités propres aux situations prédéfinies et des fonctionnalités propres aux situations ad hoc. Ce système sera construit à partir d'un système prédéfini qui sera adapté suivant l'optique de [SCHM 91].

' The problem with incorporating models of plans (established procedures, organizational structures, or conceptual schemes) in computer systems is not that plans are fictitious. Rather, plan serve a heuristic function in action by identifying constraints, pitfalls and strategic positions in the field of work. [...] In fact, 'plans' may serve different functions. Consider organisational procedures, for examples: Procedures may of course codify 'good practice', recipes, proven methods, efficient ways of doing things, work routines. In flexible work organizations such procedures are of little value and may actually impede flexibility. [...] Whatever the function, however, organizational procedures are not executable code but rather heuristic and vague statements to be interpreted and instantiated maybe even by means of intelligent improvisation. ' [SCHM 91]

1.5.2. Les supports de la coopération collégiale

Nous n'avons présenté ce type de coopération que dans la mesure où elle est le pendant de la coopération répartie. Dès lors, nous trouvons superflus de nous attarder à la présentation de leur support.

Chapitre 2 :

WoorKS

WoorKS est un framework destiné au développement de système de workflow. Il a été développé dans le but de réduire la durée du processus de développement d'un workflow particulier.

Il se base sur l'hypothèse suivante : les workflows d'entreprises différentes sont rarement équivalents mais ils partagent un grand nombre de caractéristiques communes. L'idée est de rassembler ces caractéristiques. Il suffirait alors de les préciser pour définir un workflow qui répond aux besoins spécifiques d'un client.

Dans une première partie, nous présenterons la connaissance générique de WoorKS, c'est-à-dire l'ensemble des objets qui représentent les caractéristiques que l'on trouve le plus souvent dans les systèmes de workflow. Ces caractéristiques sont regroupées dans les cinq modèles suivant :

- le modèle organisationnel définit le QUI (infra **2.1.1. Le modèle organisationnel**);
- le modèle informationnel définit le QUOI (infra **2.1.3. Le modèle informationnel**);
- le modèle temporel définit le QUAND (infra **2.1.5. Le modèle du temps**);
- le modèle des opérateurs définit le COMMENT (infra **2.1.4. Le modèle des opérateurs**).

Ensuite, nous aborderons le processus de développement, à savoir : comment utiliser la connaissance générique représentée par les modèles pour définir un workflow particulier.

WoorKs est un système de workflow s'appuyant sur une base de données orientée objet. La librairie d'objets génériques facilite la description d'un cas particulier. Par spécialisation, ou simplement par la réutilisation d'objets existants, il est aisé de construire une application.

2.1. La connaissance générique de WoorKs

Nous allons présenter les objets qui représentent les caractéristiques que l'on retrouve le plus souvent dans les systèmes de workflow. Cette connaissance évolue dans le temps en suivant l'évolution des connaissances scientifiques dans ce domaine. La personne chargée de la maintenance et de l'enrichissement de la connaissance générique est appelée l'ingénieur d'application [Lu 93].

En réalité, ce rôle d'ingénieur d'application est joué par un grand nombre de chercheurs répartis dans les entreprises BULL (France) et T.A.O. (Espagne). Le projet étant sur pied depuis 1990 et n'ayant cessé d'évoluer depuis, un grand nombre de personnes ont rempli cette fonction.

La diversité des personnes qui travaille sur le projet engendre une mouvance de la connaissance générique. Disposant de sources d'informations étalées dans le temps et dans l'espace nous ne pouvons nous permettre de la décrire précisément sans faire un amalgame irréel, à savoir : une connaissance générique qui n'existerait que par rapprochement d'idées différentes. Donc, nous avons choisi d'en présenter les concepts en évitant systématiquement les préoccupations liées à sa programmation.

2.1.1. Le modèle organisationnel [Ader 92 (1)]

2.1.1.1. Introduction

Il représente les acteurs du groupe et leur(s) rôle(s) de manière à ce qu'une certaine activité puisse être assignée à un certain acteur.

L'organisation est représentée par trois grandes catégories d'objets: acteur, groupe et rôle. Les groupes forment un arbre doté de la relation 'parent'. Le rôle est l'objet qui fait le lien entre un acteur et un groupe. Un rôle est tenu par un acteur au sein d'un groupe. De façon à

exprimer simplement qui fait quoi au sein de la procédure, le modèle comprend des méthodes ad hoc. Elles permettent notamment d'obtenir le supérieur d'un acteur, tous les membres d'un groupe, un acteur ayant un certain rôle au sein d'un groupe donné. [ADER 90].

Le modèle représente la structure, l'environnement et les compétences organisationnelles. Par soucis de clarté, ces trois composantes seront présentées séparément et illustrées par le cas de l'élaboration du budget d'un institut universitaire [PETIT 93].

2.1.1.2. La structure organisationnelle

Toute organisation est une réunion artificielle de personnes poursuivant un objectif commun. Afin d'en maîtriser la complexité, cet objectif est généralement divisé en plusieurs sous-objectifs qui seront attribués à divers sous-ensembles de personnes; l'ensemble de ces sous-groupes et les relations qui existent entre chacun d'eux forment la structure organisationnelle.

On retrouve parmi ces sous-groupes la comptabilité, le service du personnel, la production du produit X ..., chacun de ces sous-groupes pouvant se diviser à son tour en sous-groupes : le service du personnel peut se diviser en un groupe chargé des paiements, un autre groupe chargé de l'octroi des congés, etc.

Les différents concepts et relations qui définissent la sémantique de la structure organisationnelle sont représentés par le schéma E./A. de la figure 2.1.

Un **groupe** est le sur-type commun aux organisations et aux groupes organisationnels. Il se décompose en groupes organisationnels et il comprend des méthodes de manipulations qui permettent de retrouver des informations relatives à la relation de décomposition dont il fait partie. On peut, par exemple, retrouver les groupes ascendant et descendant d'un groupe (Fig.2.2.).

Un **groupe organisationnel** est le sur-type commun à l'ensemble des groupes qui font partie d'une organisation. Il est composant d'un groupe et descend (Fig. 2.2.) d'une organisation. La connaissance générique de WoorKS propose des groupes organisationnels de deux types : les line unit et les groupes fonctionnels.

Une **line unit** est une composante de l'organisation qui peut être responsable d'un ou de plusieurs groupes fonctionnels. On retrouve généralement ce type de structuration dans les entreprise de production de biens.

Un **groupe fonctionnel** est une partie de l'organisation qui peut être, soit un projet, soit un comité et qui est sous la responsabilité d'une ligne unit. Un **projet** est une réunion momentanée de personnes poursuivant un but commun; un **comité** est un groupe temporaire créé dans un but spécifique. Le projet et le comité se composent de membres parmi lesquels on trouve un leader, une secrétaire et un contrôleur financier.

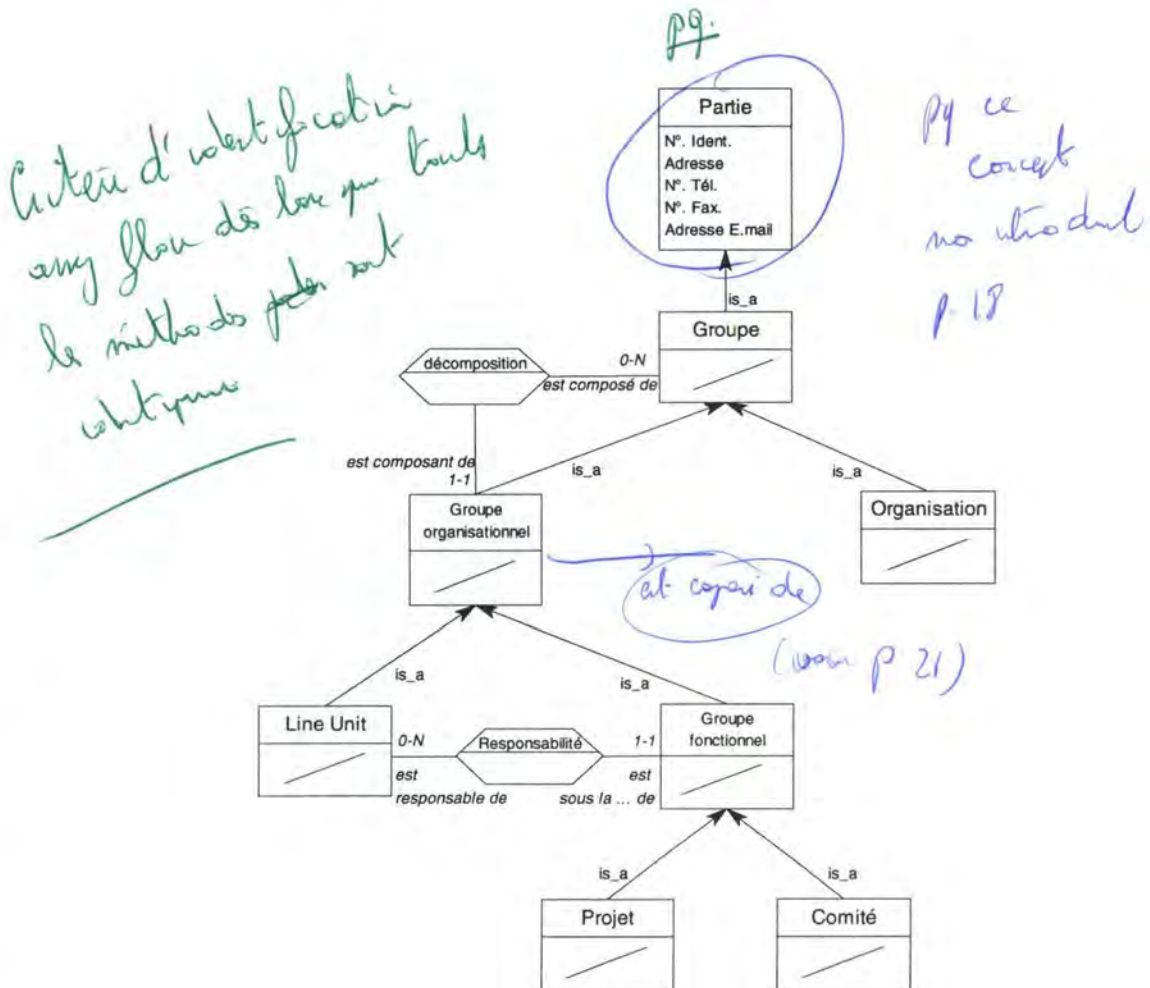


Fig. 2.1. Schéma E/A¹ de la structure organisationnelle

Voici le schéma qui illustre la relation de décomposition qui existe entre les groupes et les groupes organisationnels. Le groupe qui est la racine de l'arbre de décomposition est une organisation. Pour illustrer les notions d'ancêtre, de descendant, de composant et de

¹Nous supposons le modèle entité/association connu. Dans le cas contraire, le lecteur trouvera les renseignements nécessaires à sa compréhension dans [BODA 89].

Deuxièmement, tous les schémas E/A que vous trouverez dans ce travail sont relatifs au moment d'instanciation des types représentés et ils ne tiennent compte que des contraintes d'intégrité relatives aux connectivités et aux sous-typages.

composé, nous allons les identifier pour les groupes organisationnels D et F de la figure 2.2. Considérons le groupe organisationnel f : ses ancêtres sont D,A et l'organisation; ses descendants sont G et H; il est composé de D; il est composé de G et H. Examinons le groupe organisationnel D : ses ancêtres sont A et l'organisation; ses descendants sont E, F, G, H; il est composé de A; il est composé de E et F.

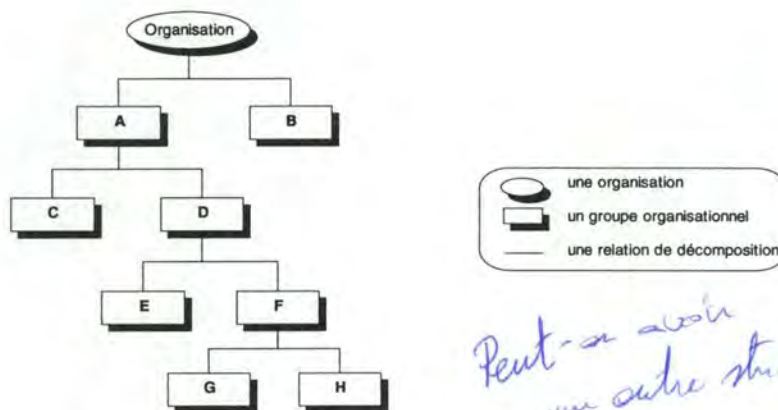


Fig. 2.2. Illustration de la relation de décomposition

2.1.1.2.1. Application au cas d'élaboration du budget dans une université

Rappel

Le but de cette partie, et de celles qui illustreront les compétences et l'environnement organisationnel, n'est pas de donner une description exacte d'un institut universitaire. Il s'agit simplement de mettre en évidence les fonctionnalités qui sont offertes par les objets de la connaissance générique. Ces objets forment un recueil des caractéristiques que l'on retrouve le plus souvent dans les organisations; ils ne constituent pas, à eux seuls, la solution à notre problème de représentation. Il faudra donc les compléter, en les spécialisant, pour représenter le cas particulier qui nous intéresse.

La structure universitaire

Voici une partie de la structure organisationnelle d'une université présentée dans le cas [PETIT 93]. Les éléments qui en font partie sont les suivants : le rectorat, l'administration, le service financier, les facultés (Fig.2.3.). Une faculté se compose d'un secrétariat, de départements et de projets de recherches.

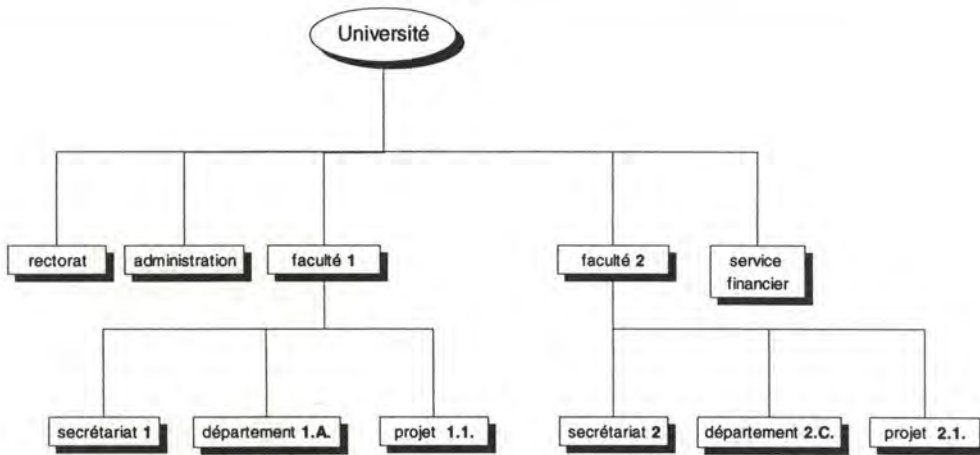


Fig. 2.3. Partie de la structure d'un institut universitaire

Représentation

L'objet université sera représenté par une instanciation de l'objet organisation.

Les composants² de l'université sont des groupes organisationnels. La connaissance générique n'offre de support à la représentation qu'à deux types de groupes organisationnels, à savoir : les line unit et les groupes fonctionnels. De par leurs caractéristiques propres, les composants de l'université ne peuvent être représentés par instanciation de l'un de ces groupes organisationnels. Nous allons donc créer cinq nouveaux types d'objets en spécialisant l'objet groupe organisationnel. Ces nouveaux types disposeront de toutes les caractéristiques d'un groupe organisationnel, plus leurs attributs et leurs méthodes propres. Nous illustrons nos propos par la spécification³ de l'objet faculté :

L'objet faculté est une spécialisation de l'objet groupe organisationnel. Il a comme attributs ceux hérités de son sur-type (N° Ident, Adresse, N° Tel, N°Fax, Adresse E.mail) et des attributs qui représentent la date de fondation, le nombre d'étudiants, etc. Il a comme méthodes celles héritées du sur-type, plus des méthodes qui permettent d'accéder à ses composants : identifier le secrétariat d'une faculté, obtenir la liste de ses départements, obtenir la liste de ses projets de recherche, etc.

⇒ faculté est 1 instance de faculté 5a

²Il s'agit du rectorat, de l'administration, des facultés et du service financier.

³Cette spécification reprend les éléments de représentation de la structure d'une faculté et des éléments généraux, tels que le nombre d'étudiant. Les renseignements relatifs à ses compétences seront définis dans la partie consacrée à l'illustration de la représentation des compétences organisationnelles.

Les facultés se décomposent en un secrétariat, des départements et des projets de recherches. Ces objets sont des groupes organisationnels composants d'une faculté. Pour les représenter, on utilisera l'objet projet dans le cas des projets de recherche et on spécialisera l'objet groupe organisationnel pour représenter les caractéristiques propres aux secrétariats et aux départements. Ces nouveaux types d'objets héritent des attributs et des méthodes de leurs sur-types et ont des attributs et des méthodes propres.

2.1.1.3. Les compétences organisationnelles.

Cette partie du modèle répond aux besoins de répartition du travail parmi les acteurs de l'organisation. Chacun des acteurs dispose de certaines compétences qui lui permettent de jouer un rôle au sein d'un groupe organisationnel. Par exemple, la personne qui fait des études d'informatique est apte à s'insérer dans une équipe de travail formée d'informaticiens. Donc, chacun des acteurs est sensé être capable de réaliser certains types de tâches qui, lorsqu'elles apparaissent, doivent lui être attribuées.

Les différents concepts et relations qui définissent la sémantique des compétences organisationnelles sont représentés par le schéma E./A. de la figure 2.4.

Un **acteur** est employé par une organisation; il joue au moins un rôle au sein d'un groupe de l'organisation par laquelle il est employé et il est au service d'une ligne unit. L'objet qui le représente offre des méthodes qui permettent de lui attribuer ou de lui enlever un rôle, d'identifier l'acteur qui preste le rôle de manager au service de laquelle il est affecté, etc.

Un **rôle** est une association entre un acteur et une fonction exercée au sein d'un groupe. Cet objet n'a pas d'instance, c'est le sur-type commun aux rôles qui seront réellement prestés au sein de l'organisation. Il offre des méthodes qui permettent d'assigner un rôle à un acteur ou à un groupe, d'obtenir le groupe et l'acteur auxquels est associé un rôle, etc.

La connaissance générique n'offre pas de rôles spécifiques. Ceux-ci doivent être créés en spécialisant l'objet rôle. Nous avons représenté les rôles spécifiques **secrétaire** et **président**; en fonction des besoins d'une organisation particulière, d'autres spécialisations de l'objet rôle peuvent être créées.

Un **groupe** propose un ensemble de rôles et offre des méthodes qui permettent de retrouver les acteurs qui prestant un certain rôle au sein de ce groupe. Par exemple, elles

permettent de retrouver la(ou les) personne(s) qui preste(nt) le rôle de secrétaire au sein de l'organisation.

Une **organisation** emploie un ensemble d'acteur et offre une méthode qui permet de retrouver l'acteur qui preste le rôle de président.

Une **line unit** dispose d'un ensemble d'acteurs qui est à son service et offre des méthodes qui permettent d'en ajouter ou d'en supprimer.

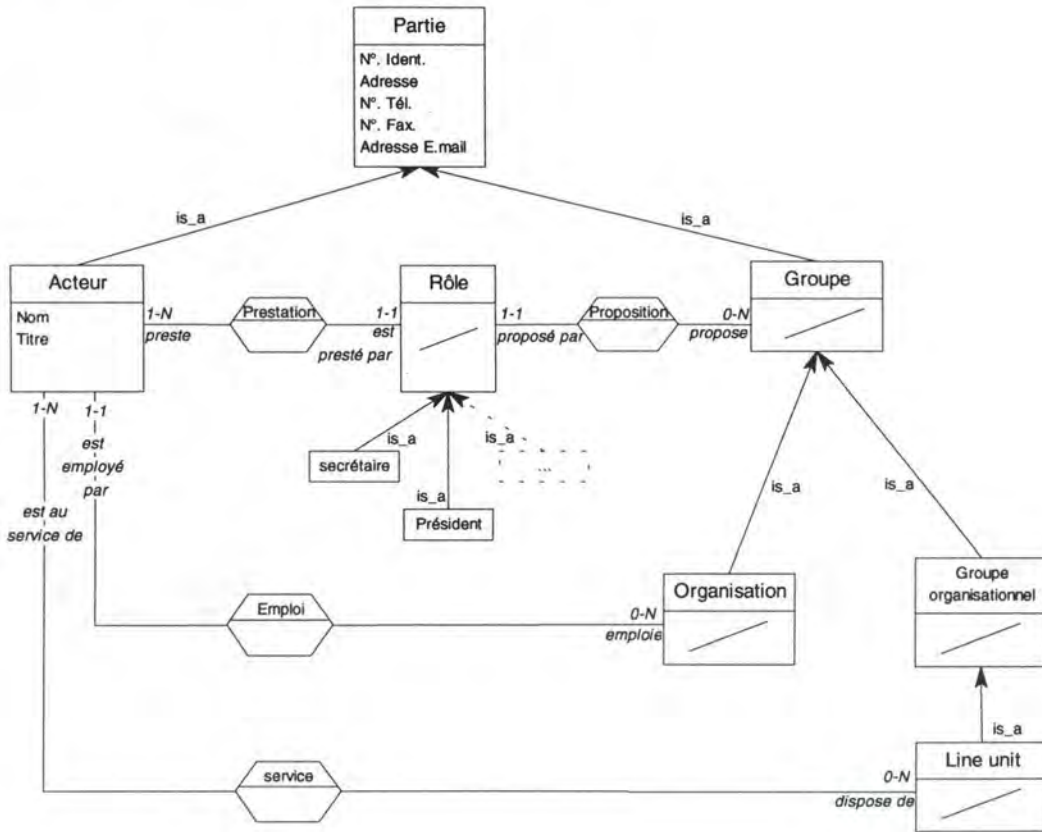


Fig. 2.4. Schéma E/A représentant les compétences organisationnelles

2.1.1.3.1. Application au cas d'élaboration de budget d'une université

L'université est un ensemble de personnes qui ont été engagées pour effectuer un certain type de tâche. Le type de tâche auquel est affecté une personne est représenté par le rôle qu'elle détient. Nous trouvons au sein de l'université les rôles de recteur, administrateur délégué, directeur financier, doyen, professeur, etc. Chacun de ces rôles est tenu par un acteur au sein d'une partie de l'organisation.

2.1.1.4. Représentation de l'environnement

Une organisation se trouve dans un secteur d'activité au sein duquel elle entre en contact avec des clients, des fournisseurs, des concurrents, sous-traitants, etc.

Les différents concepts et relations qui définissent la sémantique de la représentation de l'environnement organisationnel sont représentés par le schéma E./A. de la figure 2.5.

Une **organisation externe** est un ensemble de personnes publiques.

Une **personne externe** est un sur-type commun à l'ensemble des personnes qui ne font pas partie du système de workflow. Elles peuvent être soit publiques, soit privées.

Une **personne publique** est une personne externe qui représente une organisation au sein de laquelle elle joue un rôle. Ex : Madame Dupont, présidente de la société Philips qui est un de nos clients.

Une **personne privée** est une personne externe qui ne fait pas partie d'une organisation avec laquelle on est en contact. Ex : un client privé d'une banque.

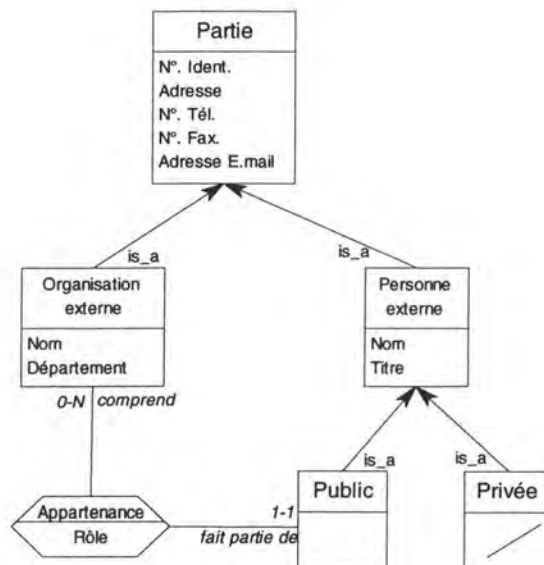


Fig. 2.5. Schéma E/A de l'environnement organisationnel

2.1.1.4.1. Application au cas d'élaboration du budget dans une université

Le délégué gouvernemental, dont le rôle est de faire respecter la législation relative aux versements, sera représenté par une personne publique; il représente le gouvernement qui constitue une organisation externe.

2.1.2. Le modèle des références [ADER 92 (5)] [ADER 92 (8)]

2.1.2.1. Introduction

WoorKs programme et contrôle l'exécution des actions sur les informations. Cependant la plupart des actions ne sont pas exécutées par WoorKs mais par les applications classiques de bureau (traitement de texte, tableurs, ...). Les objets sur lesquels les actions sont exécutées sont aussi externes à WoorKs. L'objet de référence et ses spécialisations sont la vue qu'a WoorKs de ces objets externes. Ils encapsulent ce dont on a besoin pour localiser ces objets externes et pour reconnaître leur type dans le but de pouvoir appeler automatiquement l'outil qui devra les manipuler, lors d'une certaine étape d'une procédure.

2.1.2.2. Définition

Une **application externe** est un outil informatique utilisé par un acteur dans le cadre d'une procédure. Elle manipule les **objets externes** : elle les crée, les modifie et les détruit.

Les différents concepts et relations qui définissent la sémantique des objets de référence sont représentés sur le schéma E/A de la figure 2.6.

Une **référence** est le lien par lequel WoorKs peut accéder à un objet externe.

Une **référence locale** représente un fichier accessible par le gestionnaire des fichiers du système d'exploitation. Ce fichier est identifié par un nom et un type. Le type sera utilisé pour faire le lien entre l'objet externe et l'application externe qui sera appelée pour le manipuler. Il existe deux sous-types de l'objet référence locale, ils ont été créés pour tenir compte des caractéristiques propres aux objets externes partagés et temporaires.

Une **référence partagée** représente un fichier accessible par plusieurs utilisateurs simultanément. Ce type d'objet se place sur un support qui permet les accès simultanés à un fichier.

Une **référence temporaire** représente un fichier qui est créé pour une durée limitée. Ces objets sont regroupés sur un support spécifique caractérisé, par exemple, par une capacité relativement faible et un taux de transfert élevé.

L'objet **référence IMAGEWorks** représente un objet multimédia manipulé par IMAGEWorks [ADER 92 (8)]. Nous n'avons pas complètement spécifié cet objet; pour plus de renseignement, nous renvoyons aux références précitées.

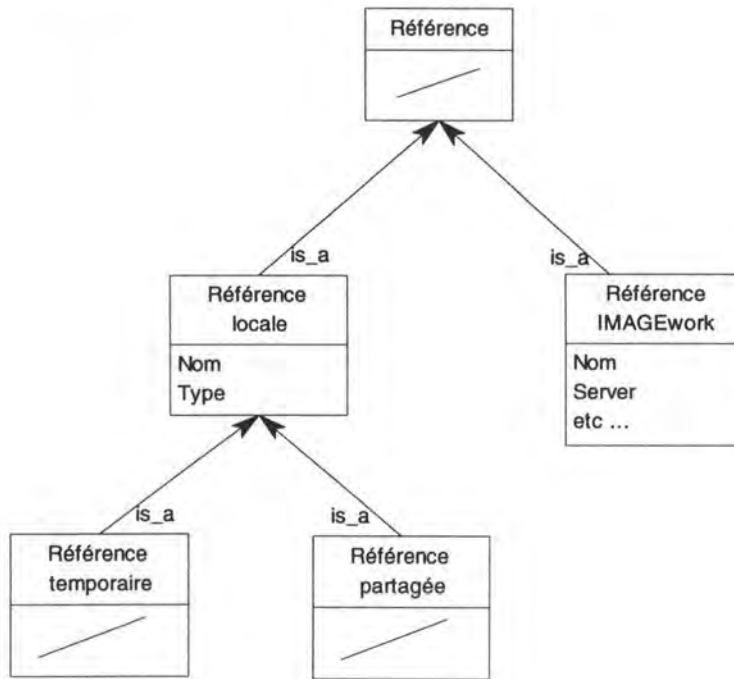


Fig. 2.6. Schéma E/A des objets de références

2.1.2.3. Application

L'administrateur délégué dispose de l'historique des allocations pour chaque département. Ces graphiques sont des objets externes qui ont été créés par une application externe : par exemple, un tableur.

2.1.3. Le modèle de l'information [ADER 90][ADER 92 (7)]

Rappelons, une fois encore, que ce modèle fait partie de la connaissance générique, il représente les caractéristiques les plus communes des informations. Ces caractéristiques sont destinées à être réutilisées pour créer des objets d'information particuliers.

2.1.3.1. Introduction

Le modèle de l'information représente un minimum de la sémantique de l'information manipulée par un workflow lors de son exécution. La sémantique représentée est constituée de l'acteur et du groupe propriétaire, ainsi que du contenu de l'objet d'information. Le contenu de l'objet est, soit un rassemblement d'objets de type standard (entier, alpha numérique, numérique, etc.) ou structuré (tout type d'objet défini dans un des modèles de la connaissance générique), soit un objet créé et manipulé par une application externe.

Le rassemblement d'objets de type standard et/ou structuré est un **formulaire interne**; c'est un objet d'information interne. Les objets manipulés par des applications externes sont des objets d'information externe.

2.1.3.2. Définitions

Les différents concepts et relations qui définissent la sémantique des objets d'informations sont représentés sur le schéma E/A de la figure 2.7.

Un objet **information** est le sur-type commun aux objets internes et aux objets externes. Il est lié au modèle de l'organisation par une relation de propriété. Un objet d'information appartient à un acteur et à un groupe d'acteurs.

Un **objet 'interne'** est un objet qui peut être manipulé sans avoir recours à une application externe, il contient explicitement son contenu, contrairement aux objets externes qui n'en ont qu'une référence. Ce type d'objet n'est pas explicitement représenté sur le schéma de la figure 2.7., car il s'agit d'objets particuliers qui, de par leurs caractéristiques propre ne font pas partie de la connaissance générique. Il s'agit, par exemple, d'un formulaire qui représente la commande d'un client. Ces objets sont définis par spécialisation de l'objet d'information.

Un **document externe** est la représentation d'un objet externe. Il est associé à un objet de référence locale (infra : le modèle des références) qui l'identifie et qui l'associe avec l'application externe qui le manipule.

Un **folder** est une liste de documents externes. Il se caractérise par le nombre d'éléments qu'il contient et par un commentaire textuel qui informe l'utilisateur de sa signification. Par exemple, on pourrait imaginer une liste de feuilles de calcul qui

correspondent à l'ensemble des factures impayées. Le commentaire de ce folder sera, par exemple, ' factures impayées'.

Un **rapport** est un document externe qui a comme objet soit un objet d'information, soit un objet organisationnel, soit un objet du modèle des procédures (infra 2.1.6. **Le modèle des procédures**), etc. Il apporte un complément d'information concernant un objet à ses utilisateurs.

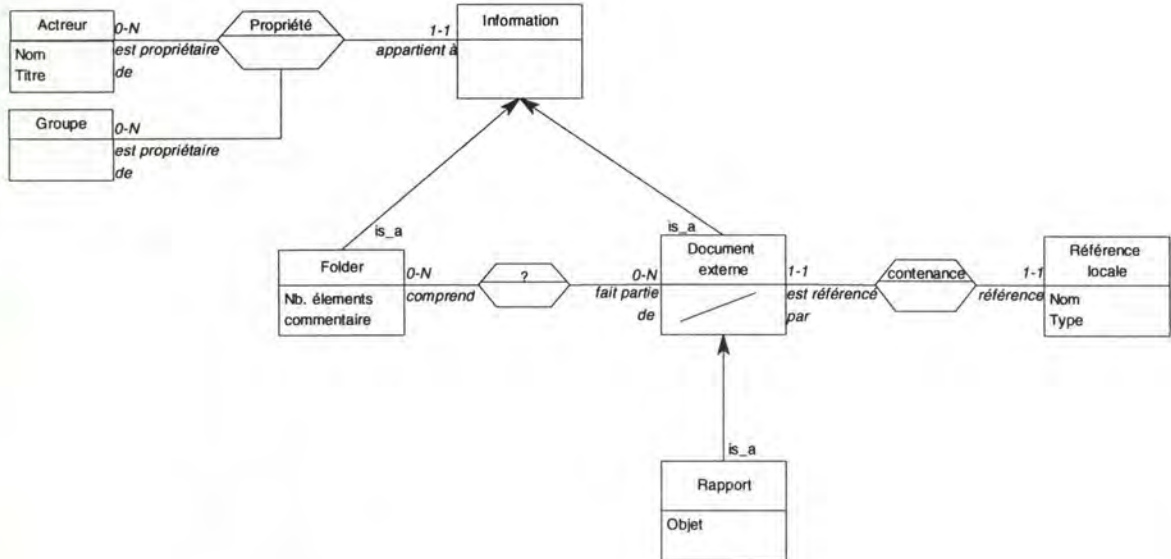


Fig. 2.7. Schéma E/A des objets d'informations

2.1.3.3. Application

N.B. : Si on considère un formulaire de demande de crédit accompagné d'une photo, il se compose d'un objet externe (la photo) et d'un objet interne (le formulaire).

2.1.4. Le modèle des opérateurs

2.1.4.1. Introduction

Dans les bureaux classiques, les employés utilisent un grand nombre d'outils tels que les éditeurs, les traitements de texte, les tableurs, etc.

L'employé souhaite un environnement qui se placerait au-dessus de ses outils préférés. Il y est habitué et ne veut pas en changer. Pour ce faire, WoorKs intègre les applications externes.

La gestion des documents étant l'un des buts principaux des systèmes de workflow, WoorKs fournit des opérateurs génériques (to browse, edit, print, mail or fac documents) pour des documents de tout type.

En utilisant les mécanismes de synchronisation propres aux opérateurs, on peut les assembler pour produire des opérateurs de plus haut niveau. Par exemple, un opérateur d'édition tel que 'vi' peut être combiné avec un opérateur de mailing pour produire un nouvel opérateur qui édite et envoie un fichier à une certaine adresse.

2.1.4.2. Définition

Les différents concepts et relations qui définissent la sémantique des opérateurs sont représentés par le schéma E./A. de la figure 2.9.

Un **opérateur** est un traitement automatisé. Il est défini par une interface, qui spécifie ses entrées et ses sorties, et par des règles de transformation des entrées en sorties. Les informations en entrées sont soit un message résultant d'une interaction avec l'utilisateur, soit un élément de la mémoire du S.I.

Une **relation de décomposition** définit une structuration arborescente des opérateurs d'après la règle suivante :

un opérateur de niveau intermédiaire i ($i \geq 1$) peut faire partie d'un opérateur de niveau $i+1$ et peut se décomposer en $0-N$ opérateurs de niveau $i-1$.

Ex : un opérateur d'impression d'un formulaire de niveau i se décompose en opérateurs d'impression de champs de niveau $i-1$ et peut être utilisé pour construire un opérateur d'impressions d'une liste de formulaire de niveau $i+1$.

Les opérateurs sont des éléments dynamiques qui évoluent au cours du temps. Cette évolution est représentée par l'attribut **état** dont le domaine de valeurs est représenté par la figure 2.8. Cette représentation des états remarquables des opérateurs au moment 't' permet l'implémentation des mécanismes de synchronisation représentés à la figure 2.10.

plutôt l'état d'un processus

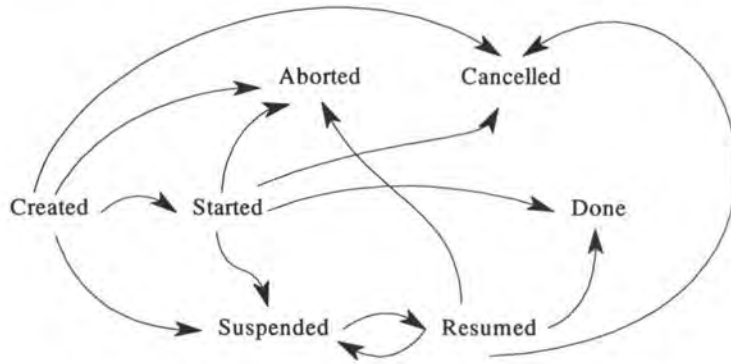


Fig.2.8. Cycle de vie des opérateurs

Le **gestionnaire des opérateurs** est utilisé par les opérateurs qui utilisent des applications externes. Ils utilisent le système d'exploitation de la machine et les applications externes pour réaliser l'objectif de l'opérateur qui l'a appelé. (infra : **2.1.4.3.2. Les opérateurs externes**).

La connaissance générique offre deux types d'opérateurs : les opérateurs de document et les opérateurs action. Les **opérateurs de documents** sont destinés à la manipulation des documents et les **opérateurs action** font la liaison entre le modèle des opérateurs et le modèle des procédures.

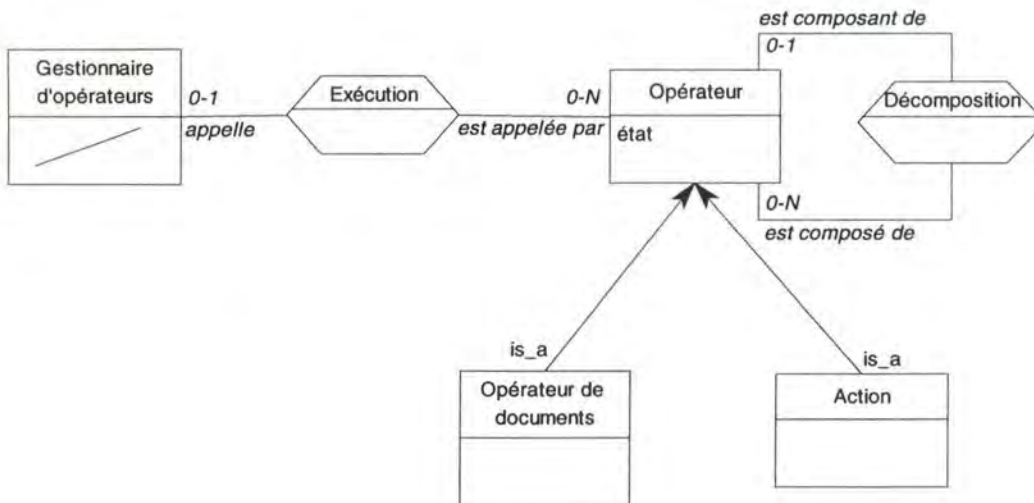


Fig. 2.9. Schéma E/A du modèle des opérateurs

2.1.4.2.1. Les mécanismes de synchronisation

La relation de décomposition qui existe entre les opérateurs de différents niveaux engendre des problèmes de synchronisation. Par exemple, les opérateurs de niveaux 3, composés d'opérateurs de niveaux 2 et 1, ne peuvent se terminer avant la terminaison de leurs composants. C'est pourquoi l'objet opérateur comprend des **méthodes de synchronisation** qui implémentent le protocole de communication de la figure 2.10. La relation de décomposition situe les opérateurs des différents niveaux dans une architecture client/serveur. Au sein de celle-ci, les opérateurs composés sont les clients de leur(s) composant(s) qui sont les serveurs des premiers.

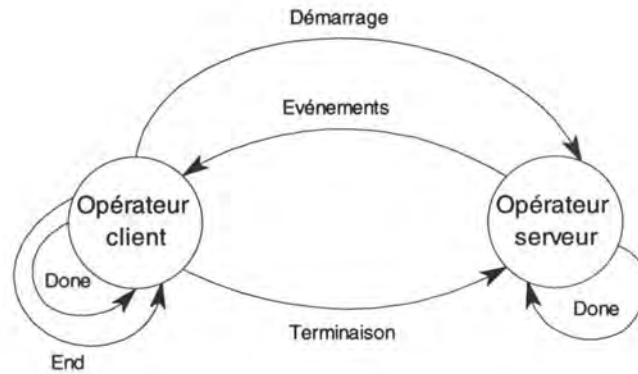


Fig. 2.10. Protocole de communication de deux opérateurs

Le père crée et 'réveille' ses opérateurs fils en leur envoyant un ordre démarrage; il peut les démarrer parallèlement ou séquentiellement. Lorsque le cycle de vie d'un des fils évolue, il doit avertir son père. Ex. : lors de sa terminaison, il générera un événement *action done*. Lorsque le père reçoit un événement de terminaison normale ou anormale en provenance d'un de ses fils, il lui renvoie un ordre de terminaison avant de s'arrêter.

2.1.4.2.2. Les opérateurs de documents [PONS 92 (2)]

C'est un ensemble d'opérateurs de manipulation d'objet référence locale (supra **modèle des références**). La connaissance générique offre les opérateurs d'impressions, d'expédition par courrier électronique, de visualisation et de modification. D'autres opérateurs peuvent être créés en fonction des besoins d'un workflow particulier.

Les objets et relations qui sont offerts par la connaissance générique sont représentés par le schéma E/A de la figure 2.11.

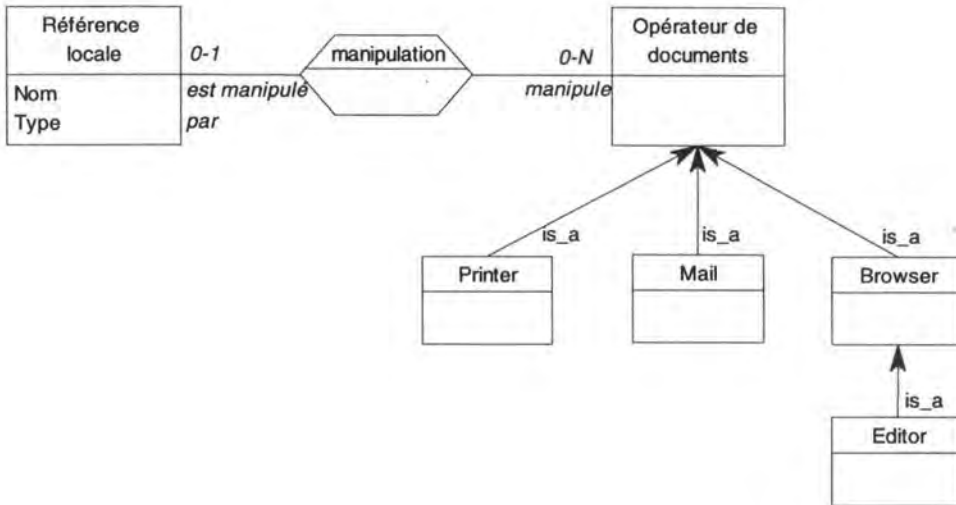


Fig 2.11. schéma E/A des opérateurs de documents

Les objets de référence locale sont accessibles par le gestionnaire de fichiers du système d'exploitation. Ils peuvent donc être de tout type et avoir été créés par n'importe quelle application. Les opérateurs de documents sont génériques dans le sens où ils s'appliquent à tous les types de documents : par exemple, ils permettent de visualiser des documents créés par 'vi' ou par 'word'. En fait, ils utilisent l'application externe de manipulation appropriée à chaque type de documents. Pour effectuer le lien entre le document à manipuler et l'application à utiliser, le gestionnaire des opérateurs consulte le trader qui interprète le type de l'opérateur qui l'appelle et le type du document pour identifier l'application externe qu'il faut démarrer.

2.1.4.2.3. Liaison entre les modèles des procédures et des opérateurs [ADER 92 (6)] [PONS 92 (2)]

Les objets et les relations relatifs à la sémantique des actions sont représentés par le schéma E/A de la figure 2.12.

L'action est l'objet qui lie le modèle des opérateurs au modèle des procédures (infra le modèle des procédures). Une procédure est un workflow particulier qui se décompose en activités qui constituent les pas d'exécution du workflow. Une activité s'exécute par le biais d'une action qui se décompose en opérateurs.

manif ?
 + contradiction avec la pag 40 ?

L'action offre un protocole de communication qui permet au module des opérateurs de communiquer avec les activités. Elle hérite des mécanismes de synchronisation et du cycle de vie de l'objet opérateur.

Une **action interactive** est une action qui collabore avec une commande interactive (infra le **méta model**) pour effectuer des échanges d'informations avec l'utilisateur.

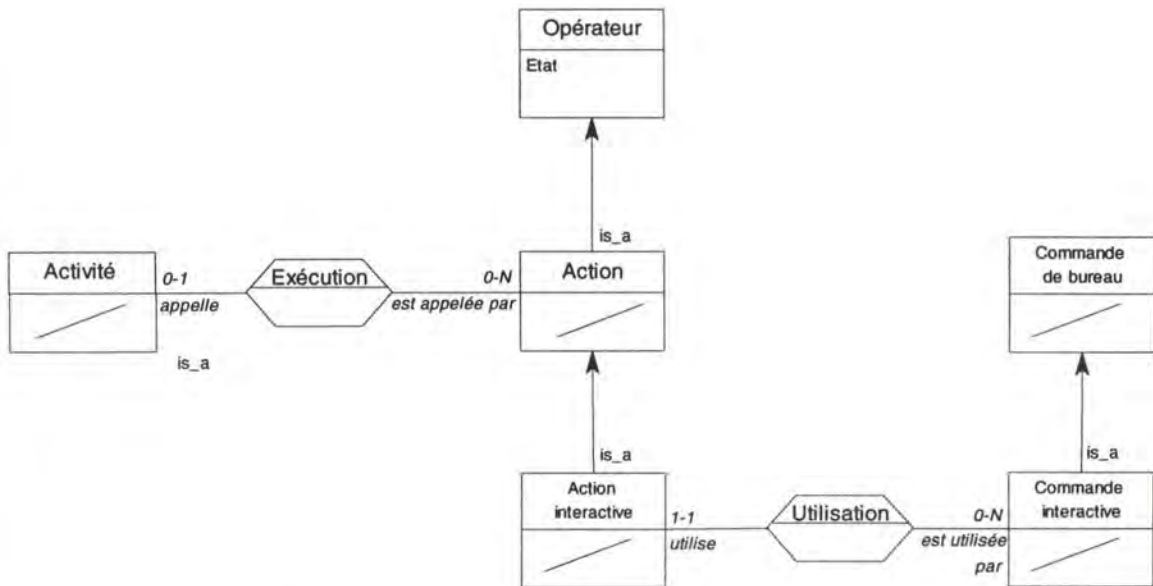


Fig. 2.12. Schéma E/A des actions

2.1.4.3. Typologie

Rappelons que l'ensemble des opérateurs présenté ci-dessus fait partie de la connaissance générique. Ils seront utilisés tels quels ou spécialisés en vue de construire un système de workflow particulier. C'est pourquoi nous présentons une typologie générale qui caractérise les opérateurs présentés ci-dessus et ceux qui seront développés ultérieurement.

Au cours de cette présentation, les composants d'un opérateur seront appelés ses serveurs et le composé sera appelé le client.

2.1.4.3.1. Les opérateurs interactifs

Ils permettent le dialogue avec l'utilisateur, ils sont nécessaires pour obtenir ou montrer de l'information. Ex. : une des étapes de la procédure d'allocations budgétaires dans une université consiste en un remplissage du formulaire de demande par les doyens de chaque faculté; cette étape se fera par le biais d'un opérateur interactif.

L'interaction avec l'utilisateur implique une interface qui affiche de l'information et qui permet à l'utilisateur de communiquer. Les opérateurs interactifs utilisent des objets d'interface (infra **le méta modèle**) dont le rôle est d'afficher des objets interactifs⁴ à l'écran et d'écouter le clavier d'un ordinateur. Ce sont eux qui interagissent avec le système d'exploitation pour utiliser le clavier et l'écran. Ces objets interactifs font partie d'un framework de développement d'interfaces appelé T.O.I. (Task-Oriented Interaction) [LU 92]

Ils héritent des mécanismes de synchronisation et du cycle de vie des opérateurs et ils disposent d'un protocole de communication qui leur permet de collaborer avec les objets du framework T.O.I pour assurer la partie physique du dialogue utilisateur.

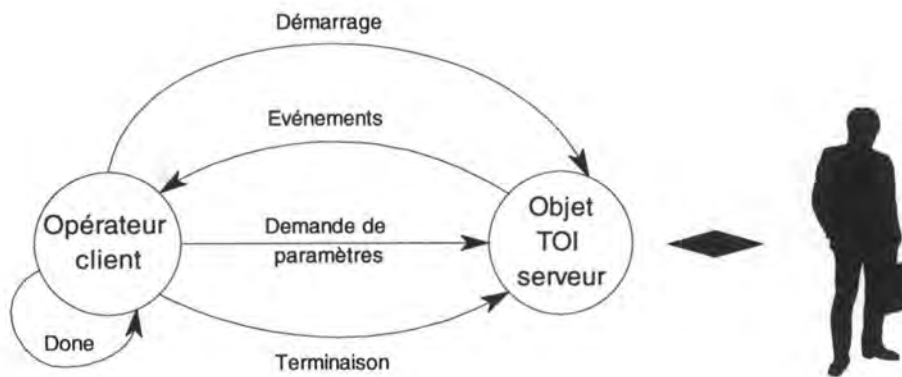


Fig. 2.13. Protocole de communication entre un opérateur et un objet T.O.I.

Au moment de son démarrage, l'opérateur interactif crée et démarre l'objet commande de l'interface qui répond à ses besoins. Ensuite il lui demande d'entrer en interaction avec l'utilisateur en lui envoyant un message de demande de paramètres.

L'objet de commande transmet cette demande à l'utilisateur via l'écran et lui donne la possibilité de répondre en utilisant le clavier. L'objet de commande envoie un message à l'opérateur interactif soit si l'utilisateur a répondu à une demande, soit si l'utilisateur a fait quelque chose qui modifie significativement l'état de l'objet de commande.

Ex : Au moment où le formulaire doit être rempli, il utilisera un objet T.O.I. capable d'afficher un formulaire, de permettre à l'utilisateur de le remplir via

⁴ Ce sont des champs de saisie, des fenêtres d'affichage, des boutons, etc ...

le clavier, etc. L'enregistrement et la mise à disposition du formulaire au sein du workflow sont assurés par l'opérateur interactif.

2.1.4.3.2. Les opérateurs externes

La fonction des opérateurs externes est d'accéder aux applications externes. Imaginons une secrétaire habituée à travailler avec le traitement de texte X. Lors de l'implantation du système de workflow, il faudra créer un opérateur externe qui permettra l'utilisation de ce traitement de texte dans le cadre d'un workflow.

Comme le démarrage et la terminaison d'application externe dépendent du système d'exploitation, on utilise le gestionnaire des opérateurs qui fait office de traducteur entre le système d'exploitation et l'opérateur externe. Il communique avec le gestionnaire des opérateurs suivant le protocole représenté par la figure 2.14.

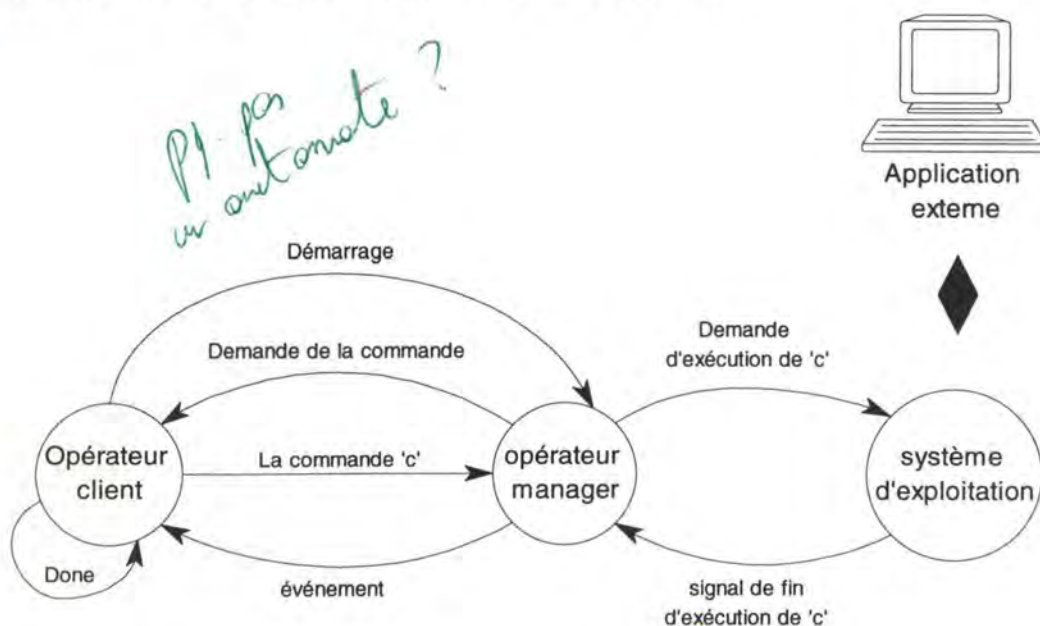


Fig. 2.14. Protocole de communication entre un opérateur et une application externe

L'opérateur externe signale son démarrage au gestionnaire des opérateurs qui l'invite à lui envoyer la commande qu'il désire transmettre au système d'exploitation. L'opérateur externe lui envoie la commande 'c' qui est traduite par l'opérateur traducteur qui l'envoie au S.E. qui va l'exécuter. Le S.E. communique la fin de l'exécution de la commande à l'opérateur traducteur qui la transmettra à l'opérateur client.

2.1.5. Le modèle du temps [ADER 90][SRIVA 92 (2)][SRIVA 92 (3)]

Ce modèle est fort important dans le cadre d'un système de workflow qui, par définition, programme le travail au sein de l'entreprise. Cet aspect prédéfini et les exigences de performances qui caractérisent les entreprises engendrent une nécessité de la prise en compte de cette unité abstraite qu'est le temps. C'est pourquoi cette représentation est, en quelque sorte, la base du système.

Il comprend quatre parties principales : un modèle de représentation du temps, un calendrier, un agenda et un mécanisme de gestion des événements.

2.1.5.1. La représentation du temps

Les différents concepts et relations qui définissent la sémantique de la représentation du temps sont représentés par le schéma E/A de la figure 2.15.

L'objet **time** est le sur-type commun à l'ensemble des objets de représentation du temps, il offre des méthodes qui permettent d'imprimer et de montrer les objets de temps. Ces méthodes sont destinées à être utilisées pour débbuger [VENKA 90].

Un **point de temps** est un objet de temps qui représente un moment. Il se caractérise par le nombre de secondes écoulées depuis le 01-01-1970. Cette valeur est enregistrée par l'attribut time.

• ∈ à intervalle ?

Un **point floue** est un point de temps, il hérite de l'attribut time et se caractérise par une précision. Cette précision est exprimée en seconde, elle exprime un ensemble de valeurs centré sur la valeur time dans lequel se promène le point floue. Par exemple, aujourd'hui à deux heures, avec une précision de 600, représente un point floue qui se promène entre une heure cinquante et deux heures dix.

Un **intervalle** est un objet de temps qui représente une période, il se caractérise par un début et une fin qui sont représentés par des points de temps.

Un **intervalle floue** est un intervalle, il hérite des relations, qui expriment sont début et sa fin, de son sur-type et se caractérise par une précision.

Une **durée** est un objet de temps qui représente le temps qui s'écoule entre deux moments; par exemple une heure, dix minutes, etc. Elle se caractérise par une durée qui est enregistrée par l'attribut durée.

Une **durée floue** est une durée qui se caractérise par une précision; par exemple une durée de dix minutes, avec une précision de 2 minutes, exprime une durée qui peut aller de 8 à 12 minutes.

Une **durée de calendrier** est une durée non floue qui exprime une période du calendrier, par exemple la durée des congés de Noël. Elle hérite de l'attribut durée et se caractérise par l'attribut unité. Elle peut donc s'exprimer en secondes, en minutes, en jours, en mois ou en années.

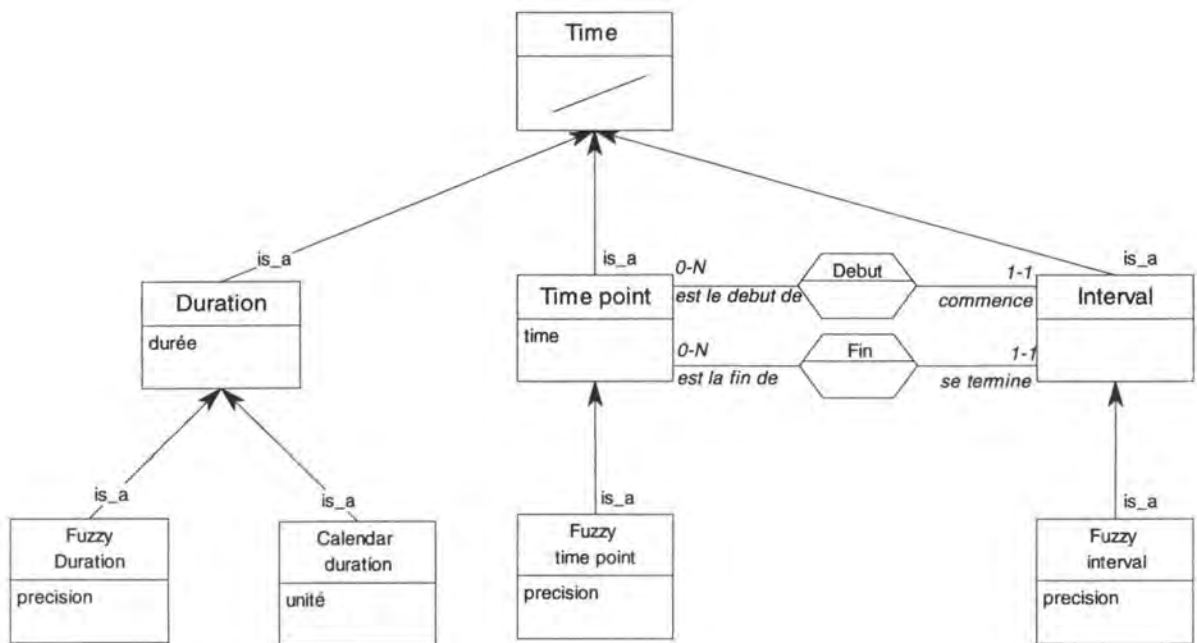


Fig. 2.15. Schéma E/A de la représentation du temps

L'ensemble des objets présentés ci-dessus dispose évidemment d'un ensemble de méthodes de manipulation. Je ne les présente pas ici, car il s'agit une fois encore de connaissance générique qu'il suffit de compléter s'il existe une manipulation dont on a besoin et qui n'est pas disponible. Néanmoins, nous pensons que l'ensemble offert est complet et conviendra dans la plupart des cas.

2.1.5.2. Le calendrier

Il est commun à l'ensemble de l'organisation. Il représente les informations spécifiques à l'organisation et à son pays d'appartenance, à savoir : les jours fériés, les horaires de travail,

etc. Il est dérivé du calendrier officiel du pays auquel on ajoute les informations spécifiques à l'entreprise.

2.1.5.3. L'agenda

Certains des acteurs de l'organisation ont un agenda qui comprend les périodes de disponibilité et de non-disponibilité (réunions, congés).

Malheureusement, il n'y a aucune liaison entre l'agenda et le modèle des procédures. Il n'existe pas de mécanisme qui tiendrait compte des tâches attribuées à un acteur pour remplir son agenda. Par exemple, si un acteur commence une tâche qui ne peut être interrompue et qui durera probablement 1 heure et demie, alors, le mécanisme de liaison pourra faire écho de cette occupation dans son agenda.

2.1.5.4. La gestion des événements

Le gestionnaire des événements permet la définition d'événements temporels, l'association d'une réaction du système à de tels événements et leur déclenchement automatique.

2.1.5.4.1. Définition

Un événement est dit **temporel** lorsque sa survenance est causée par le seul écoulement du temps. [BODA 89]

La réaction du système d'information provoquée par le déclenchement d'un événement peut être tout ce qui est représentable par le modèle des opérateurs. Cela va de l'envoi d'un message à un acteur de l'organisation à la création et au démarrage d'un workflow particulier, en passant par les changements d'état des objets dont la progression est représentée par un cycle de vie (infra **2.1.6. Le modèle des procédures**).

Le **déclenchement** d'un événement est provoqué par la vérification de sa condition de déclenchement. Dans le cas des événements temporels, la condition de déclenchement est le dépassement d'un point enregistré sur l'échelle du temps.

2.1.5.4.2. Typologie

Nous pouvons classer les événements en deux catégories : les événements qui ont comme objet un autre événement et les événements qui ont comme objet un élément du modèle des

procédures (infra : **2.1.6. Le modèle des procédures**). On appellera les premiers des événements indirects et les second des événements directs.

Les événements indirects

Ils ont comme objet un autre événement. Il s'agit par exemple du rappel de l'approche du déclenchement d'un événement. La réaction du système qui leur est associée est soit l'envoi d'un message à un acteur, soit la destruction de l'événement objet. Dans le premier cas on parlera de moniteur et dans le second de destructeur.

Un **moniteur** est caractérisé par un début, une fin et une période. Le début et la fin sont relatifs au moment de déclenchement de l'événement objet. Par exemple, on pourrait imaginer un moniteur qui nous rappelle la date anniversaire de notre femme; il se déclenchera 5 jours avant, nous enverra un message de rappel toutes les 12h et se terminera le jour de son anniversaire.

Un **destructeur** est un événement qui détruit l'événement qu'il a pour objet. Imaginons une tâche du workflow qui doit commencer après la réception d'une communication en provenance d'un acteur. Si cette communication n'arrive pas, le workflow est calé; or, dans les cas urgents on peut réaliser cette tâche sans disposer du message qui devait être envoyé. Cette situation sera représentée par un événement de communication[ADER 92 (7)]⁵, qui fera partie de la condition de déclenchement de la tâche, et un événement destructeur qui aura comme objet cet événement de communication et qui se déclenchera, par exemple, le trois novembre. Si on n'a pas reçu la communication avant cette date, l'événement destructeur se déclenche et détruit l'événement qui retardait le déclenchement de la tâche.

Les événements directs

Ces événements sont relatifs à un des objets du modèle des procédures. Les réaction du système qui sont associée à ces événements engendre l'évolution du cycle de vie de ces objets. Cette évolution peut être répercutée vers d'autres objets, de la même manière qu'un arbre que l'on abat peut engendrer l'effondrement de ses voisins.

Il existe plusieurs types d'événements temporels directs, il ne s'agit pas ici de les énumérer les uns après les autres mais plutôt de donner une idée de ce qu'il y a moyen de

⁵Les événements de communication sont le deuxième type d'événement géré par WoorKs.

représenter en les utilisant. C'est pourquoi nous les avons regroupés en deux familles : les événements d'urgences et les événements de déclenchement.

Un événement **d'urgence** est un événement dont la réaction du système qui lui est associée traite le non-respect de l'échéance de réalisation d'un processus. Cela comprend généralement l'envoi d'un message à la personne responsable du processus en retard et d'autres conséquences déterminées par le contexte dans lequel l'événement survient.

Un événement **de déclenchement** est un événement dont la réaction du système qui lui est associée traite de la participation de l'événement au déclenchement d'un processus. Prenons comme exemple un processus de traitement de commande : il se déclenchera après la réception d'au moins 20 commandes et après 17 heures. Le fait de ne permettre son déclenchement qu'après 17 heures est représenté par un événement direct qui constitue, tout comme l'accumulation des 20 commandes, une partie de sa condition de déclenchement.

2.1.5.4.3. Le mécanisme de déclenchement automatique

Les événements temporels sont la représentation d'un événement futur qui a une importance pour le fonctionnement du workflow. Pour être la représentation future d'un moment il doit avoir été identifié a priori; cette identification donne lieu à l'enregistrement de la définition de l'événement dans une base de donnée prévue à cet effet. Cette définition comprend son moment de déclenchement et la spécification de la réaction du système qui lui sera associée. Suivant une période définie a priori, un serveur d'événement consultera la base de données dans laquelle la définition des événements temporels a été enregistrée et déclenchera ceux dont le moment de déclenchement est antérieur au moment actuel. L'intervalle proposé pour accueillir la valeur de la période de consultation s'étend de 120 à 3600 secondes.

2.1.6. Le modèle des procédures

2.1.6.1. Introduction

Dans un système de workflow, certaines parties du travail sont faites automatiquement, mais la majeure partie est réalisée par les employés. Chacun de ces employés est représenté par un objet acteur du modèle de l'organisation. Il est donc possible d'établir une relation entre les parties de travail, appelées activité ou procédures, et les personnes chargées de leur bon déroulement.

Le but principal d'une procédure de workflow est de décrire les différentes activités qui font partie des procédures. Pour ce faire, le module des procédures offre des fonctionnalités de définition des procédures et utilise les services offerts par le modèle des opérateurs pour définir les activités.

Au-delà de son rôle de description du workflow, le modèle des procédures se charge de son exécution. Cette exécution tient compte de la définition de la procédure et de la définition de l'organisation pour répartir le travail parmi les employés. Un employé dispose d'une corbeille (infra 2.1.7.4.3. L'exécution du workflow) qui, d'une part l'informe du travail qu'il a à accomplir et d'autre part, lui permet d'entrer en contact avec le système soit pour réaliser ses tâches, soit pour utiliser les fonctionnalités ad hoc comme, par exemple, communiquer librement avec l'un ou l'autre de ses collègues.

pour-t-a structure de workflow avec corbeille ?

Nous commencerons la présentation de ce modèle en décrivant séparément les concepts de base que sont les procédures et les activités. On trouvera ensuite une description détaillée de ces concepts aux figures 2 et 3 qui sont des schémas E/A qui regroupent ces deux concepts en exprimant leurs liens respectifs.

2.1.6.2. Procédure [MOLLE 93] ?

2.1.6.2.1. Définition / Objectif

Une **procédure** est un ensemble prédéfini de traitements ordonnés; *prédéfini* dans le sens où la procédure est entièrement définie avant son exécution; *ordonné* dans le sens où il existe un contrôle du flux d'exécution qui fait partie de la définition de la procédure.

2.1.6.2.2. Caractérisation et spécification d'une procédure

Une procédure est identifiée par un nom logique et caractérisée par :

- un rôle qui représente les initiateurs potentiels et le responsable ;
- l'ensemble des objets qu'elle manipule ;
- une condition de déclenchement ;
- des éléments temporels ;
- un état de progression ;
- les activités qui la composent et leur ordonnancement.

Initiateurs et responsable (Fig. 2.20.)

ou rôle

Un **initiateur** est un acteur qui démarre une procédure.

bon bas p-41

Un **responsable** est acteur qui veille à la bonne exécution d'une procédure. C'est lui qui sera chargé du contrôle du travail effectué par ses collaborateurs.

Les objets

Les différents concepts et relations qui définissent la liaison entre les objets et les procédures sont représentés par le schéma E/A de la figure 2.21..

En tant que traitement d'information, une procédure a des objets en entrées, des objets intermédiaires et des objets en sorties. Les premiers sont les données, les seconds les variables intermédiaires et les derniers les résultats de la procédure.

Un objet d'information a un nom identifiant et un type qui doit être connu du système. Le type doit avoir été défini dans un des modèles, soit dans celui de l'organisation (ex.1), soit dans le modèle de l'information (ex. 2), ou encore dans un de ses autres modèles.

ex.1 : Imaginons une procédure qui a pour objet l'enregistrement d'un nouveau client. Ce client est une partie externe de l'organisation qui doit être définie comme telle dans le modèle de l'organisation.

ex.2 : Imaginons une procédure d'enregistrement de commande client qui a comme résultat la commande du client. La structure de ce document doit avoir été déclarée dans le modèle de l'information.

Sur le schéma de la figure 2.21., nous remarquons deux attributs types relatifs aux objets. Le premier qui est attribut de l'entité objet est celui dont nous venons de parler. Le second, qui est attribut de l'association de manipulation, représente le type de la relation qui l'unit à la procédure. Par exemple, s'il s'agit d'un résultat, alors son type sera 'résultats'; s'il s'agit d'une entrée statique(infra **Les entrées**), alors son type sera 'entrée statique', et ainsi de suite pour l'ensemble des types de relation qui existent entre les objets et les procédures.

Ex ou le lien de Fbo

Cela veut-il dire que l'on peut
replacer le modèle de la procédure I/A

Les entrées

Les entrées forment l'ensemble des objets à partir desquels vont s'effectuer les activités qui font partie de la procédure. Elles sont statiques ou dynamiques.

Une **entrée** est dite **statique** lorsque sa valeur est constante durant l'exécution de la procédure.

ex : Imaginons une procédure qui a parmi ses entrées statiques le directeur financier d'un département. Au début de la procédure, les informations concernant cet acteur sont recopiées comme entrée de la procédure. Si au cours de cette dernière, le département change de directeur financier, la procédure n'en tiendra pas compte et continuera son exécution sur base de l'identité de l'acteur dont elle a recopié les informations.

Une **entrée** est dite **dynamique** lorsque sa valeur peut évoluer au cours de l'exécution de la procédure. Sa valeur est évaluée à chaque fois que l'on y accède.

ex : Reprenons l'exemple précédent en déclarant le directeur financier du département parmi les entrées dynamiques. Si le directeur financier du département change, la procédure en tiendra compte car son identité n'est pas figée par un recopiage mais dérivée lors de chaque accès.

Les sorties

L'objectif d'une procédure est défini par ses résultats. Ces résultats sont obtenus en appliquant des règles de transformation sur les données. Suivant le contenu de ses données et les règles de transformation, la procédure peut se terminer de différentes manières.

ex : Imaginons une procédure d'allocation de prêt, elle se terminera soit par une acceptation, soit par un refus. Dans le cas de l'acceptation, elle aura comme résultat un rapport reprenant les modalités du prêt accordé et dans le cas du refus, elle fournira un feuillet de motivation du refus.

Ces différentes façons de terminer une procédure sont spécifiées par les **alternatives de sortie** qui se composent d'un nom, d'un ensemble d'objets résultats et d'un lien de contrôle qui les relie à une ou à plusieurs alternatives d'activités terminales de la procédure.

Explicitons : voici (Fig 2.17.) le schéma de représentation des liens de contrôle liant les alternatives de sortie de la procédure d'octroi de prêt (ALT 1 et ALT 2)aux alternatives de sorties des activités terminales (1 et 2). L'alternative 1 est la façon dont on sort de la procédure si l'activité 1 se termine par l'alternative A. L'alternative 2 sera choisie si l'activité 1 se termine par l'alternative B et si l'activité C se termine.

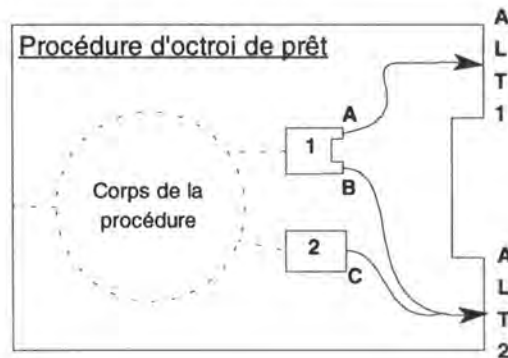


Fig. 2.17. Représentation des alternatives de sortie d'une procédure

Il existe un langage des prédicats à partir duquel on peut exprimer des liens de contrôle élaborés, en combinant les connecteurs logiques **et** et **ou** avec des liens de contrôle élémentaires.

Les éléments temporels

Les éléments temporels des procédures sont utilisés pour avoir des informations concernant le temps écoulé lors de l'exécution d'une procédure. Les informations peuvent porter sur l'ensemble ou sur une portion de la procédure. Par exemple, si une procédure se compose des activités 1,2,3 qui s'exécutent séquentiellement et que l'information temporelle qui nous intéresse est le temps passé pour réaliser les deux premières activités, on définit un élément temporel qui aura comme objets les activités 1 et 2.

Ces éléments temporels sont appelés des **durées de procédures**. Ils se caractérisent par un nom, un début et une fin, un ensemble d'intervalles à décompter et éventuellement une valeur maximale [MOLLE 93].

Le début et la fin forment un intervalle qui est défini au sein de la procédure par des références aux activités qui la composent. Par exemple, on peut définir un intervalle qui va de l'alternative A de l'activité 1 à l'alternative C de l'activité 2.

L'ensemble intervalle à décompter est un intervalle inclus à intervalle de durée de procédure dont le temps écoulé lors de son exécution devra être décompté du temps écoulé durant l'exécution de l'intervalle principal.

Si on lui attribue une valeur maximale elle exprime une contrainte temporelle qui devra être respectée lors de l'exécution. Si cette durée n'est pas respectée, l'ensemble des activités appartenants au morceau contraint seront mises en urgence (infra **Cycle de vie et événements remarquables des activités**). Pour chaque objet de durée de procédure on peut définir un moniteur (supra **2.1.5. Le modèle du temps**) qui renseignera les personnes concernées à propos du temps écoulé lors de l'exécution de la procédure. Par exemple, on peut l'utiliser pour avertir le responsable de la procédure de l'approche de l'échéance d'une durée.

Elle renseigne le responsable de la procédure sur le temps d'exécution de certains morceaux de la procédure. Pour chacune des durées, on peut définir le moment et la personne à qui les renseignements seront envoyés. Cela permet, par exemple, d'accélérer l'exécution d'une procédure si on a été informé, par une durée, du fait que nous avons passé beaucoup de temps pour effectuer les premières activités de la procédure. Ces informations peuvent aussi servir a posteriori, après l'exécution de la procédure, pour améliorer la prochaine exécution en redéfinissant, par exemple, certains morceaux pour lesquels on a surestimé ou sous-estimé le temps d'exécution.

Une condition de déclenchement

Il existe deux types de condition de déclenchement: la condition initiateur et la condition activité. Le type de la condition de déclenchement engendre une typologie des procédures : les procédures et les sous-procédures. Les premières ont une condition de déclenchement dont la réalisation dépend de la décision d'un acteur et les secondes ont une condition de déclenchement dont la réalisation dépend des circonstances d'exécution du workflow auquel elle appartient.

Une **condition initiatrice** est vérifiée lorsque l'un des acteurs initiateurs potentiels de la procédure décide de l'initialiser.

Une **condition activité** est vérifiée lorsque la procédure est appelée par une activité. Il existe trois types d'appel de sous-procédure :

- le lancement de la sous-procédure lors du démarrage de l'activité ;

- le lancement d'une sous-procédure pour chaque élément d'un ensemble d'une liste d'objets de l'activité ;
- le lancement d'une sous-procédure lors de la survenance d'un certain événement spécifié par l'activité.

La vérification de la condition de déclenchement engendre l'exécution de la procédure.

Les activités et leur ordonnancement [MOLLE 93]

Une procédure est formée d'une ou plusieurs activités ordonnancées. La spécification de la composition d'une procédure est représentée par le schéma E/A de la figure 2.20.

L'état de progression

La progression de la procédure se fait suivant un cycle de vie équivalent à celui des activités. C'est pourquoi, plutôt que de nous répéter, nous le présenterons en identifiant une typologie de ses événements remarquables⁶. Nous les avons regroupés en trois familles. La description précise du cycle de vie est présentée pour les activités. Le cycle de vie des procédures étant équivalent, nous invitons à consulter la figure X.

Les trois familles d'événements remarquables sont identifiées en fonction de la provenance des événements qui en font partie : la première regroupe les événements qui proviennent du responsable de la procédure; la deuxième ceux qui proviennent de ses composantes; quant à la troisième, elle regroupe ceux qui proviennent de ses caractéristiques propres. Précisons ces trois familles en donnant des exemples pour chacune d'elles.

Les premiers événements représentent une décision du responsable de la procédure. Par exemple, il décide d'abandonner son exécution ou de la démarrer.

Les seconds représentent l'évolution d'une activité composante. Par exemple, si l'activité terminale de la procédure se termine, alors la procédure dont elle faisait partie se termine. Il s'agit d'une réaction du cycle de vie de la procédure à l'évolution du cycle de vie d'une de ces composantes. Il en est de même pour plusieurs événements remarquables qui caractérisent l'évolution du cycle de vie d'une activité.

⁶Ce sont les événements responsables de l'évolution du cycle de vie.

Les événements de la troisième famille représentent une caractéristique propre à la procédure. Par exemple, si on a défini une échéance de réalisation pour une certaine procédure et que cette échéance n'est pas respectée, alors un événement provoquera l'évolution de son cycle de vie.

2.1.6.3. Activité[MOLLE 93] [ADER 92 (6)]

2.1.6.3.1. Définition / Objectif

Une **activité** est une étape de travail qui fait partie d'une procédure. Il existe deux types d'activités : les activités simples qui se réalisent par le biais d'un opérateur et les activités complexes qui se réalisent par le biais d'une ou plusieurs procédures.

2.1.6.3.2. Caractérisation et spécification d'une activité

Une activité est caractérisée par :

- un rôle qui représente le réalisateur de l'activité ;
- l'ensemble des objets qu'elle manipule ;
- une condition de déclenchement ;
- des contraintes temporelles ;
- la description des traitements ;
- un état d'avancement ;

Le réalisateur

Un **réalisateur** est un acteur qui est chargé de la réalisation d'une activité. Il est représenté par un rôle qui sera remplacé par son acteur lors de la création de l'activité.

Les objets

Les objets de l'activité forment un sous-ensemble des objets de la procédure dont elle fait partie. On retrouve les entrées (statiques ou dynamiques), les variables intermédiaires et les sorties. Le type de chacun de ces objets doit être défini soit dans le module de l'organisation, soit dans celui de l'information, ou bien dans celui du temps.

Les entrées

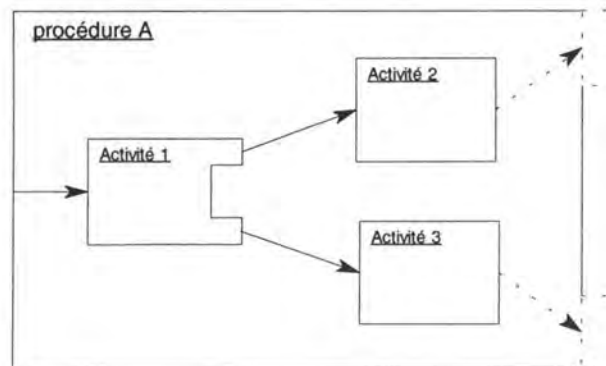
Elles sont définies par des données et un lien de contrôle qui lie l'activité à (ou aux) alternative(s) de sorties de (ou des) l'activité(s) qui la précède(nt) directement.

ex : une activité d'enregistrement de commande reçoit une commande valide et elle s'exécute, soit après l'activité d'enregistrement de commande téléphonique, soit après l'activité d'enregistrement de commande écrite.

Un **lien de contrôle** est l'expression de la position de l'activité au sein de la procédure à laquelle elle appartient. Le lien de contrôle relie l'activité avec :

- soit une (ou plusieurs) alternative(s) de sortie d'une (ou plusieurs) activité(s) de la même procédure ;
- soit avec le début de la procédure.

Illustration :



- un lien de contrôle d'activité
- une activité avec 2 alternatives de sortie
- une activité avec une alternative de sortie

Rappel

- - - éléments relatifs aux alternatives de sortie de la procédure

Fig. 2.18. Illustration du concept de lien de contrôle

Les sorties

Elles sont définies par une (ou plusieurs) alternative(s) de sortie. Une **alternative de sortie** d'une activité est caractérisée par un nom et un ensemble d'objets résultats.

ex : l'activité de validation d'une commande peut se terminer soit par l'alternative d'acceptation qui a comme résultat une commande valide, soit par l'alternative de refus qui a comme résultats la commande et une motivation de son refus.

Le choix de l'alternative de sortie est fonction de l'exécution du corps de l'activité. C'est le rôle de la **post condition** d'évaluer les résultats de l'exécution du corps de l'activité pour choisir l'alternative de sortie qui sera utilisée.

Condition de déclenchement

La condition de déclenchement d'une activité se compose de l'activation d'un des liens de contrôle qui fait partie de la définition de ses entrées, d'un événement de retard et d'une précondition. Les spécifications d'un événement de retard et d'une précondition sont optionnelles.

Un **événement de retard** est soit un événement de temps, soit un événement de communication. S'il s'agit d'un événement de temps, il exprime une date avant laquelle l'activité ne peut être démarrée. Dans le cas d'un événement de communication, il exprime le fait qu'une activité ne peut démarrer si on n'a pas reçu une certaine communication.

Une **précondition** est une condition qui porte sur les objets d'entrée de l'activité. Elle doit être vérifiée pour permettre l'exécution de l'activité. Si elle ne l'est pas, l'activité est abandonnée, ainsi que la procédure dont elle ait fait partie.

Contraintes temporelles

La durée d'exécution d'une activité peut être limitée par une durée maximale (ex.1) ou par un moment (ex.2). L'une et l'autre expriment une échéance de réalisation.

ex.1 : Imaginons une activité de traitement de commande; elle ne doit pas dépasser 10 minutes, sans quoi l'opératrice ne pourrait encoder l'ensemble des commandes reçues pendant la journée.

ex. 2 : Imaginons une activité de réalisation du bilan de l'entreprise, elle doit être terminée pour le 31/12/94.

Pour ces deux types de contraintes, il existe un mécanisme de monitoring destiné à signaler aux personnes concernées l'approche d'une échéance de réalisation.

Description des traitements

Les activités s'exécutent, soit par le biais d'une action, soit en appelant une sous-procédure. Une **action** est un opérateur caractérisé par des entrées, des sorties et des règles de transformations des entrées en sorties.

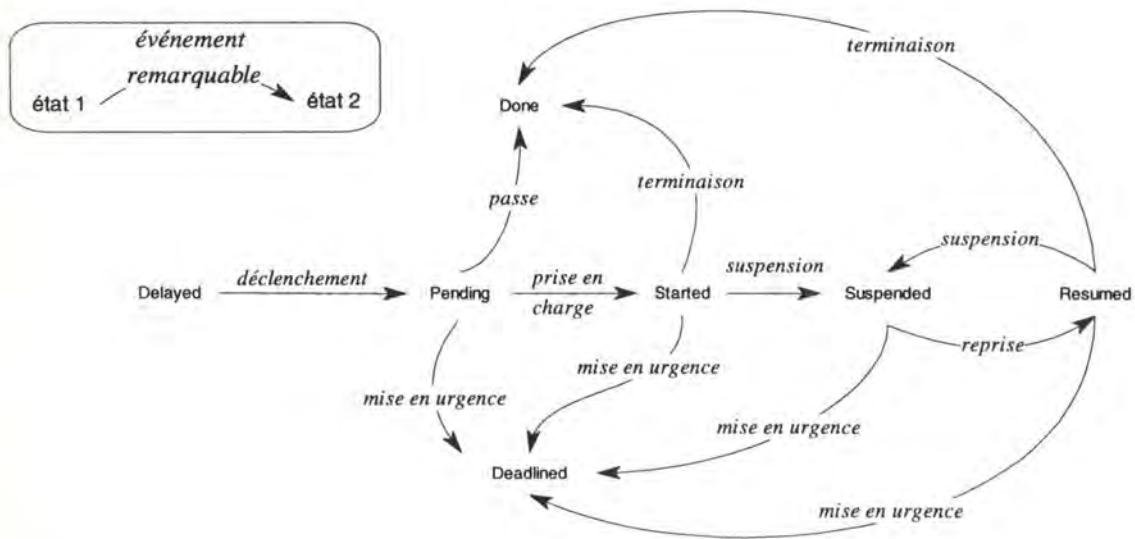
I

Cycle de vie et événements remarquables

Une activité est une entité dynamique dont la progression peut-être décrite par certains états remarquables et par les transitions pour passer de l'un à l'autre. Ces transitions sont causées par la survenance des événements remarquables dont voici la liste.

- La **création** est l'instanciation du type de l'activité. Cette instanciation résulte de la vérification d'une partie de la condition de déclenchement, à savoir : l'activation d'un de ces liens de contrôle. Cette activation correspond à la terminaison des activités qui se trouvent en amont au sein de la procédure. L'état de l'activité devient soit *delayed*, si la condition de déclenchement n'est pas totalement vérifiée (il existe un événement de retard), soit *pending* dans le cas contraire.
- Le **déclenchement** est simultané à la création s'il n'est pas retardé par un événement de retard. Il engendre la mise à disposition de l'activité (*pending*) à la personne chargée de sa réalisation.
- La **prise en charge** correspond au commencement de la réalisation de l'activité. Elle résulte d'un choix du réalisateur qui décide de commencer la réalisation de telle activité.
- La **mise en urgence** (*deadlined*) résulte du non-respect d'une échéance de réalisation qui est signalé par le déclenchement d'un événement qui représente le dépassement.

- La **suspension** correspond à une interruption de l'exécution de l'activité. Elle résulte soit du choix de l'utilisateur, soit d'un message en provenance du corps de l'activité. Ce dernier peut être une action ou une sous-procédure.
- La **reprise** correspond à la reprise de l'exécution d'une activité qui a été suspendue.
- La **terminaison** correspond au moment où l'objectif de la tâche est satisfait. L'activité se termine par une de ses alternatives de sorties.

Fig. 2.19. Cycle de vie⁷ d'une activité

2.1.6.4. Représentation E/A de la sémantique des procédures et des activités

Nous avons présenté les concepts d'activité et de procédure séparément pour faciliter la compréhension. Or ces deux objets sont fortement liés puisque les activités ne peuvent exister qu'au sein d'une procédure et les sous-procédures ne peuvent exister qu'au sein d'une activité. C'est pourquoi les schémas E/A qui sont présentés représentent simultanément ces deux composants.

⁷ Il s'agit du cycle de vie simplifié qui ne reprend que les éléments qui nous seront utiles dans la suite de ce travail.

Nous avons choisi de scinder le schéma général en deux parties : la première représente les relations qui lient le modèle des procédures au modèle de l'organisation et au modèle des opérateurs; la seconde représente les relations qui, d'une part, relient les procédures aux activités et d'autre part, définissent le contrôle de flux qui sera utilisé lors de l'exécution du workflow.

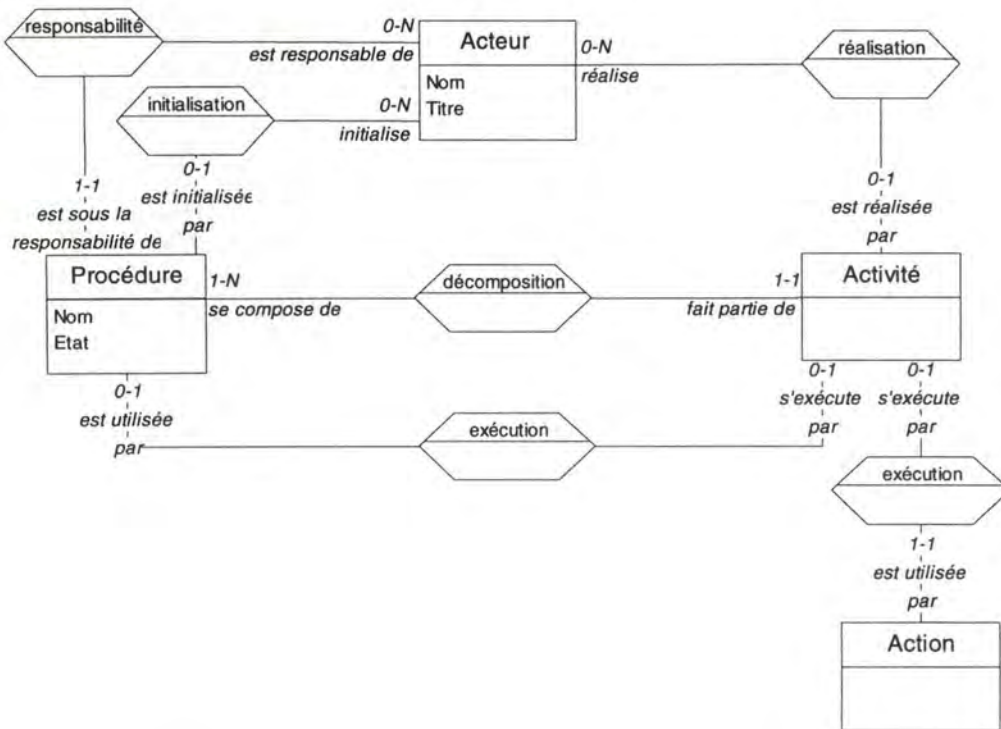


Fig. 2.20. Schéma E/A des relations avec le modèle de l'organisation et des relations avec le modèle des opérateurs.

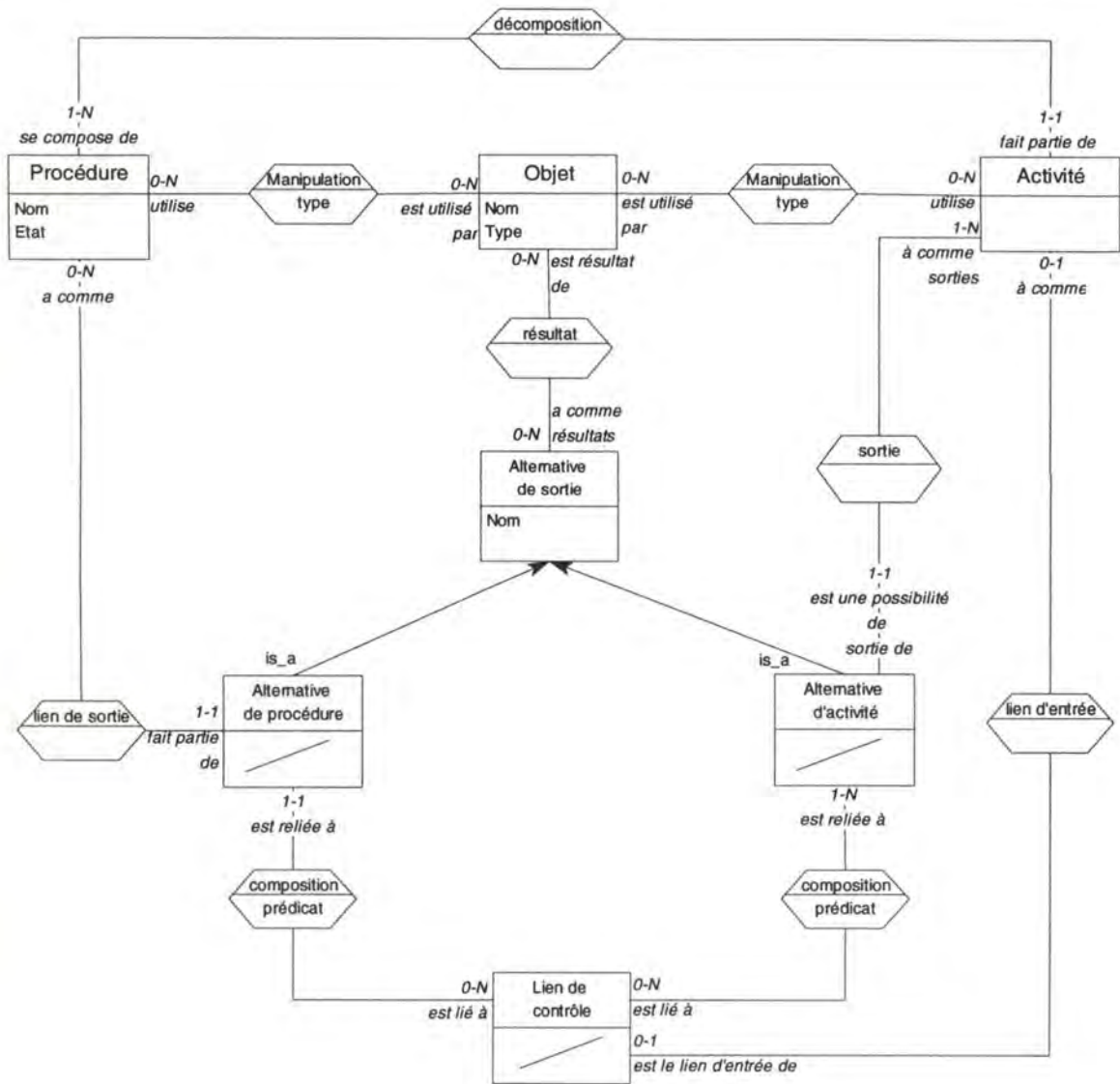


Fig. 2.21. Schéma E/A des relations qui lient les procédures aux activités et qui définissent le contrôle de flux utilisé lors de l'exécution du workflow.

2.1.7. La description des workflows

L'ensemble des modèles présentés précédemment sont au service du modèle des procédures. Ils lui offrent un ensemble de concepts sans lesquels il est impossible de définir un workflow. Voyons comment chacun de ces modèles y contribue, pour ensuite aborder les fonctionnalités propres au modèle des procédures.

2.1.7.1. Fonctionnalités offertes par le modèle de l'organisation

Pour savoir 'qui fait quoi' il faut avoir une relation entre le modèle de l'organisation et le modèle des procédures. En offrant les concepts de rôle et d'acteur le modèle de

l'organisation définit le Qui. Via les concepts de responsable, d'initiateur et de réalisateur, le modèle des procédures établit une relation entre les acteurs et les traitements qui représentent le ' Qui fait Quoi '.

L'initiateur et le responsable d'une procédure, ainsi que le réalisateur d'une activité sont définis par un rôle. En se basant sur le concept de rôle, le modèle de l'organisation offre un mécanisme de répartition dynamique du travail, dans le sens où les personnes ne seront réellement choisies qu'au moment de la création de la procédure ou de l'activité. Cela permet de tenir compte du contexte du moment pour choisir l'acteur qui est le plus à même de réaliser le travail.

Un autre avantage de cette modélisation est la localisation des modifications de l'organisation à son modèle. Explicitons : si un ensemble d'activités doit être réalisé par le directeur financier et que le directeur financier s'appelle Dupont, lorsque Dupont part en vacances, il suffit de faire jouer le rôle de directeur financier à une autre personne pour que l'ensemble du travail normalement attribué à Dupont devienne le sien. On n'a donc pas besoin de modifier la définition du workflow.

2.1.7.2. Fonctionnalités offertes par le modèle des opérateurs

Il offre le concept d'action qui est un opérateur à partir duquel on définit le corps d'une activité simple.

2.1.7.3. Fonctionnalités offertes par le modèle du temps

Il offre les concepts à partir desquels on va définir le 'Quand' en spécifiant un ensemble de contraintes temporelles sur les activités ou sur les procédures.

2.1.7.4. Fonctionnalités offertes par le modèle des procédures

2.1.7.4.1. Structuration des procédures et des activités

De manière similaire à la structuration des opérateurs, il existe une relation de décomposition qui définit une structuration arborescente des procédures et des activités. Cette décomposition suit la règle suivante :

une activité de niveau intermédiaire i ($i > 1$) provient de la décomposition d'une procédure de niveau $i-1$ et peut se décomposer en $0-N$ procédures ou opérateurs de niveau $i+1$.

On peut la représenter par le schéma suivant :

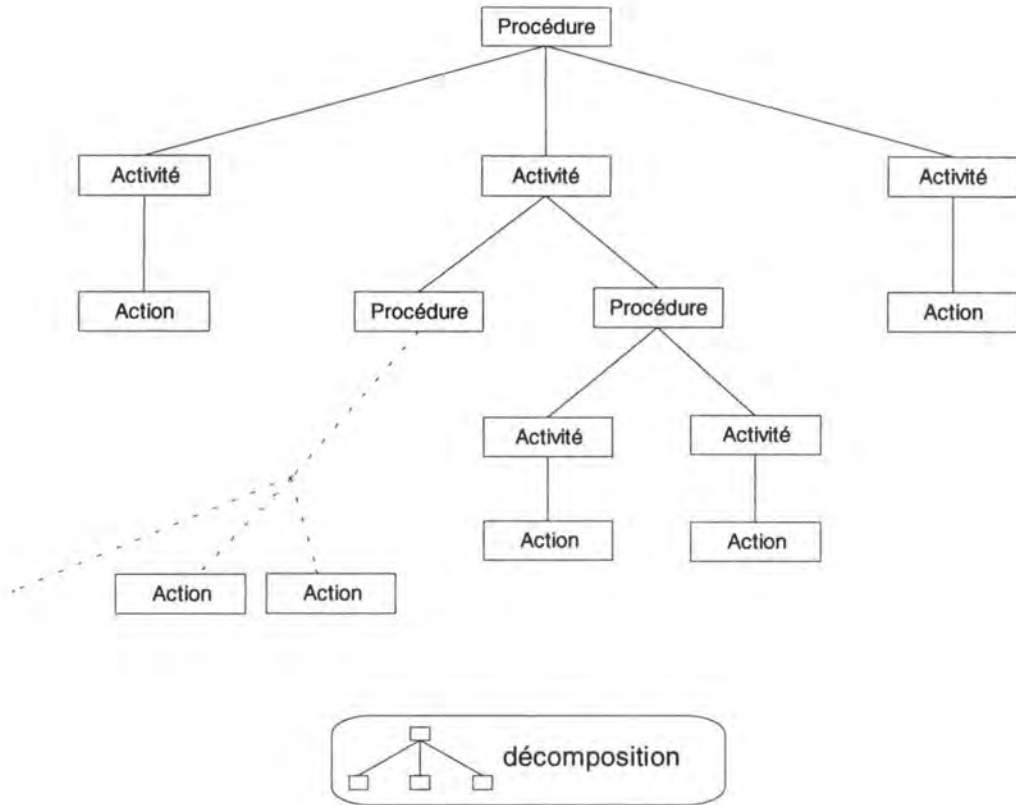


Fig. 2.22. Structuration des procédures et des activités

En plus de la sémantique de la relation de décomposition des procédures en action, ce schéma représente un arbre complet de la structuration des traitements. Une procédure est un traitement qui s'effectue en utilisant des activités. Une activité est un traitement qui s'effectue en utilisant, soit une ou plusieurs sous-procédures, soit une action. Une action est un traitement offert par le modèle des opérateurs. Les feuilles de l'arbre sont des actions car ce sont les traitements de bas niveau, à savoir : la spécification précise⁸ de ce qui va être exécuté.

2.1.7.4.2. Dynamique des procédures et des activités

Le modèle offre un ensemble de concepts à partir desquels on spécifie les conditions d'exécution et d'enchaînement des procédures et des activités. Voyons comment on peut les utiliser pour décrire les situations réelles d'ordonnancement des traitements.

⁸Elle est rédigée en Cool (langage de programmation orienté objet)[COOL 92 (1)][COOL 92(2)].

Nous ne prétendons pas évaluer la puissance sémantique des concepts offerts par WoorKs. Il s'agit plutôt de le mettre en mouvement pour expliciter la portée des concepts qui ont été présentés ci-dessus.

Nous nous baserons sur le modèle de la dynamique des traitements proposés par F. Bodart et Y. Pigneur. Ce modèle représente ' [...] les conditions d'exécution et d'enchaînement des traitements en vue de caractériser le comportement d'un système informatique.' [BODA 89] Les systèmes informatiques concernés sont les systèmes de gestion. Ce modèle est exhaustif, dans le sens où il offre un ensemble de concepts à partir desquels il est possible de représenter l'ensemble des systèmes informatiques de gestion.

En vue d'évaluer la capacité des concepts de WoorKs à représenter l'ensemble des situations de gestion, nous allons essayer d'exprimer chacun des concepts du modèle de [BODA 89] en utilisant les concepts de WoorKs.

Le mécanisme d'enchaînement

Définition

'Un enchaînement est dit **séquentiel** lorsque chaque processus d'un type donné précède systématiquement un traitement d'un autre type.' [BODA 89].

'Un enchaînement est dit **parallèle** lorsque chaque processus d'un type donné provoque le déclenchement simultané de plusieurs processus de types différents.' [BODA 89].

'Un enchaînement est dit **convergent** lorsque des processus de plusieurs types donnés provoquent chacun, et séparément, le déclenchement d'un processus d'un autre type.' [BODA 89].

Expression

Voyons comment les concepts de WoorKs expriment ces différents types d'enchaînement.

Les enchaînements séquentiels et parallèles sont notamment exprimés à l'aide des liens de contrôles de la façon suivante :

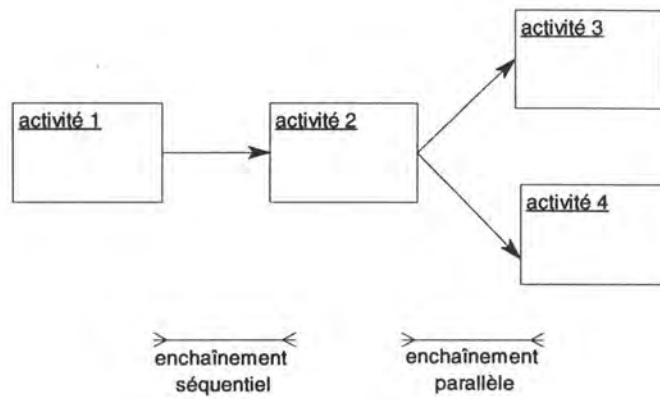


Fig. 2.23.

Les enchaînements convergents sont notamment exprimés à l'aide des événements de déclenchement des sous-procédures (à savoir une procédure qui est initialisée par une activité complexe).

Le mécanisme de synchronisation

Définition

'Le mécanisme de *synchronisation* est utilisé pour représenter une situation dans laquelle il faut **attendre** la survenance d'une combinaison de deux ou plusieurs événements avant de provoquer une réaction particulière dans le système d'information.' [BODA 89]

Expression

Voici deux exemples de définition de point de synchronisation qui utilisent respectivement une condition de déclenchement composée d'un événement de retard et d'un lien de contrôle, et une condition de déclenchement composée de deux liens de contrôles.

Les *conditions de déclenchement* des activités et des sous-procédures :

ex. : une activité ne peut débuter avant une certaine date ou avant d'avoir reçu une certaine communication. De plus, cette activité ne peut débuter avant la terminaison de l'activité qui se trouve en amont dans la structure de la procédure.

On exprime cette condition en définissant un événement de retard et en définissant un lien de contrôle qui lie l'activité aux précédentes.

Les *liens de contrôle* expriment des points de synchronisation de la façon suivante :

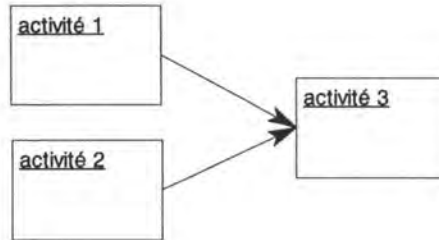


Fig. 2.25.

Soit l'activité 3 : une activité simple pour laquelle n'a été défini aucun événement de retard; elle ne sera mise à disposition de son réalisateur que lorsque les activités 1 et 2 seront terminées.

Le mécanisme de sélection

Définition

'Le mécanisme de *sélection* est utile pour modéliser une situation dans laquelle un événement ne contribue pas systématiquement au déclenchement d'un processus mais bien en fonction d'une *condition* portant sur l'état du système d'information au moment de la survenance de l'événement.' [BODA 89].

Il est naturellement adapté pour prendre en compte des enchaînements conditionnels de traitement.

' [...] un **enchaînement** est dit **conditionnel** si un processus provoque sélectivement le déclenchement d'un autre processus en fonction d'une condition.' [BODA 89].

Expression

La *précondition* des activités est un mécanisme de sélection qui définit les conditions pour lesquelles l'événement de déclenchement entraîne la mise à

disposition de l'activité à son réalisateur. Si elle n'est pas vérifiée, l'exécution de l'activité est abandonnée.

La *post condition* des activités permet la définition d'enchaînements conditionnels de traitements. Suivant certaines conditions (1) portant sur l'exécution du corps d'une activité, on en sortira par telle ou telle alternative (2 ou 3), et suivant les liens de contrôle auxquels elles sont rattachées, on déclenchera telle ou telle activité (b ou c) (Fig. 2.26.)

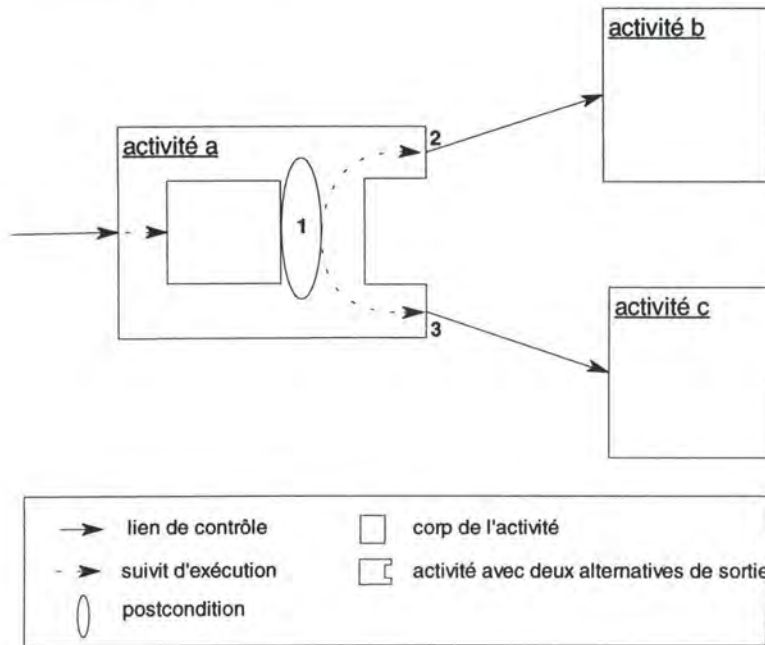


Fig. 2.26.

Le mécanisme de duplication

Définition

'Le mécanisme de *duplication* est introduit pour modéliser une situation dans laquelle un même événement peut contribuer au **déclenchement** de plusieurs processus de même type [...]'[BODA 89].

Expression

Les activités peuvent manipuler des objets qui forment une liste d'éléments, par exemple, une liste de factures. Les activités complexes peuvent démarrer une procédure pour chaque élément d'une liste: par exemple, l'activité de facturation

peut démarrer une sous-procédure de traitement de facture pour chacune des factures qui se trouve dans la liste (Fig 2.27.).

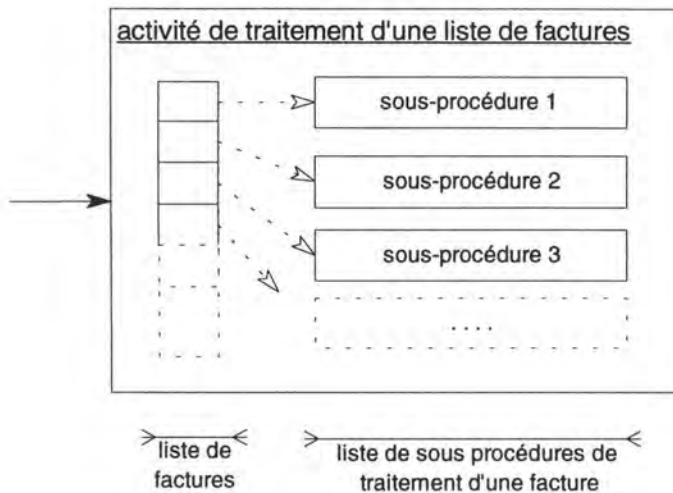


Fig. 2.27.

2.1.7.4.3. L'exécution du workflow

Introduction

Les éléments présentés dans la partie consacrée aux modèles de WoorKs offrent les concepts de procédures, d'activité, de document, d'acteur ... ,qui permettent de définir un workflow. A partir de ces descriptions, le module de coordination va exécuter le workflow.

Le **module de coordination** est la partie dynamique du modèle des procédure. Il est chargé de faire évoluer les workflows conformément à leur définition. Il interagit avec l'utilisateur par le biais de sa corbeille.

Nous le présenterons en nous basant sur trois scénarios significatifs, à savoir : l'exécution d'une procédure, l'exécution d'une activité et l'exécution d'une action.

Par souci de clarté, la création, la destruction et le déclenchement des événements autres que l'activation d'un lien de contrôle (événement de retard, contraintes temporelles, ...), ne sont pas pris en compte.

Avant de commencer la présentation, il faut parler de la corbeille qui est l'outil que l'utilisateur emploie pour entrer en contact avec le système, et dont le système se sert pour entrer en contact avec l'utilisateur.

La corbeille [PONS 92 (3)][PONS 92 (4)]

Une **corbeille** est un des objets qui lient le modèle des procédures à celui de l'organisation. Elle appartient à un acteur et comprend les objets du modèle des procédures avec lesquels il a des relations. Ces objets sont les activités qu'il devra réaliser ou qu'il est en train de réaliser. Ce sont aussi les procédures en cours d'exécution et se trouvant sous sa responsabilité, ainsi que les procédures qu'il a le droit d'initialiser.

La corbeille offre des fonctionnalités qui seront utilisées par le modèle des procédures et par l'utilisateur. Les fonctionnalités au service du modèle des procédures sont l'ajout et l'enlèvement d'une activité ou d'une procédure. Celles utilisées par l'utilisateur sont : la *prise en charge* d'activité, l'initialisation de procédure (ce qui correspond à sa *création*) et l'abandon de procédures dont il est le responsable.

Le texte en italique correspond à des événements remarquables du cycle de vie des activités ou des procédures présenté lors de la définition de l'état de progression des activités.

Scénarios

Exécution d'une procédure

Un initialiseur potentiel (supra : **2.1.6. Le modèle des procédures**) sélectionne, par l'intermédiaire de sa corbeille, le type de la procédure qu'il désire démarrer.

Le module de coordination :

- initialise les données de la procédure ;
- attribue la responsabilité de la procédure à un acteur en la plaçant dans sa corbeille;
- consulte la structure de la procédure et crée ses premières activités :
 - place la référence de l'activité dans la liste des activités appelées par la procédure ;
 - attribue la réalisation de l'activité à un acteur ;

- place l'activité dans la corbeille de son réalisateur ;

- lorsqu'une activité se termine :
 - sauvegarde la date de terminaison de l'activité ;
 - recopie les sorties de l'activité dans le contexte de la procédure ;
 - enlève l'activité de la corbeille de son réalisateur ;
 - enlève l'activité de la liste des activités appelée par la procédure ;
 - détruit l'activité ;
 - modifie l'état d'avancement de la structure de la procédure;
 - active les liens de contrôle reliés à l'alternative de sortie par laquelle s'est terminée l'activité;
 - si le lien est relié à une activité, et qu'il est activable, alors créer l'activité (supra : création des activités de départ);

 - si le lien est relié à une alternative de sortie de la procédure, et qu'il est activable alors terminer la procédure;

Exécution d'une activité

Toute activité fait partie d'une procédure (supra **2.1.6.3. Activités**) par laquelle elle est créée lors de l'activation d'un de ses liens d'entrée.

Le module de coordination :

- prend les données de l'activité dans le contexte de la procédure et les enregistre dans le contexte de l'activité ;
- enregistre l'identifiant de la procédure appelante dans l'activité ;
- crée l'action par laquelle s'exécute l'activité :
 - transfère les données de l'activité vers l'action ;
 - démarre l'exécution de l'action ;

- lorsque l'action se termine :
 - transfère les résultats dans le contexte de l'activité ;
 - évalue la post-condition de l'activité et détermine l'alternative de sortie de l'activité ;

- avertit la procédure appelante de la terminaison de l'activité en lui donnant l'alternative de sortie qui a été empruntée;

Exécution d'une action

Une action est un opérateur (supra **2.1.4. Le modèle des opérateurs**); elle s'exécute sur les données que lui a donné l'activité qui l'a appelée. Lorsqu'elle se termine, elle avertit l'activité appelante en lui rendant les résultats.

2.1.8. Le méta-modèle

Il offre un ensemble de mécanismes qui permettent de créer, modifier, visualiser les objets du système. Il utilise les services offerts par le framework T.O.I. (task-oriented interaction), une typologie générale des objets et une représentation de la hiérarchie des types d'objets.

Nous commencerons par définir ses composants pour ensuite aborder leurs interactions en vue de spécifier les mécanismes offerts par le modèle.

2.1.8.1. Définition des composants

2.1.8.1.1. T.O.I.

T.O.I. est un framework destiné à la génération d'interface utilisateur.

Il offre un ensemble d'objets interactifs à partir desquels on spécifie la couche de dialogue⁹ de l'interface utilisateur. Parmi ces objets on retrouve notamment des boutons, des champs de saisie et d'affichage de texte, un toolkit de construction de menu, etc.

Il existe un type d'objets interactifs qui mérite que l'on s'y attarde plus longuement: il s'agit de l'objet à partir duquel seront implémentées les commandes du système. Commençons par définir une commande pour ensuite aborder brièvement l'objet qui servira de base à sa construction.

Une **commande** est un objet interactif identifié par un nom logique et caractérisé par un objectif qu'elle réalise en communiquant avec l'utilisateur et avec les objets sujets de son action, suivant des protocoles qui sont fonctions de la commande et du type de l'objet sujet.

⁹ La couche dialogue de l'interface utilisateur spécifie la présentation des objets à l'écran.

Les commandes sont créées soit lors de la sélection d'une feuille d'un menu par un utilisateur, soit par un opérateur interactif¹⁰.

Ex : la commande de création à partir de laquelle seront créés tous les objets du système.

L'objet qui servira de base pour la spécification des commandes offre des fonctions primitives de communications avec l'utilisateur.

Ex : l'initialisation du processus de demande des paramètres initiaux de lancement d'une commande.

2.1.8.1.2. La représentation des types

Définition

Un **méta-type** est un objet qui représente explicitement un type d'objets. Il est caractérisé par :

- son sur-type ;
- son type, à savoir : la catégorie d'objet à laquelle il appartient (objets nommés ou non, objets instanciables ou non)
- ses sous-types ;
- l'objet de représentation que l'on doit utiliser pour manipuler les objets de ce type ; (=nil s'il s'agit d'un type d'objet non-nommé)

Ex. : la représentation du méta-type d'un acteur est la suivante :

- son sur-type est une partie de l'organisation ;
- son type est un objet nommé, on peut donc y accéder directement par son nom ;
- il n'a pas de sous-type ;
- l'objet de présentation est l'objet acteur ;

L'**interface** est la représentation hiérarchique de l'ensemble des types d'objets qui sont définis dans Woorks.

¹⁰ Voir le modèle des opérateurs pour plus de renseignements concernant les opérateurs interactifs.

Fonctions

Parmi l'ensemble des fonctions offertes par l'objet d'interface, on retrouve notamment :

- la recherche du nom de l'objet de présentation que l'on va utiliser pour effectuer une commande interactive qui a comme objet un objet nommé ;
- le contrôle de la création d'instances, justifié par l'existence d'objets dit de structuration qui ne doivent jamais être instanciés.

2.1.8.1.3. Les objets

Du point de vue de l'interface il existe trois types d'objets dans le système.

Les objets **nommés** sont des objets qui sont enregistrés dans une base de données et auxquels on peut accéder directement à partir de leur nominateur et de leur type. Ils utilisent un moniteur qui permet de les verrouiller lorsque l'on est en train de les modifier (ex. : un acteur, une organisation, ...).

Les objets **non-nommés** sont des objets qui sont enregistrés dans une base de données mais qui n'ont pas de nom et, par conséquent, ne peuvent être atteints directement. Il faut passer, soit par leur OID, soit par un autre objet (ex : une procédure, une activité,).

Les objets **transitoires** sont des objets qui ne sont pas enregistrés dans une base de donnée et qui n'ont pas de nominateur. Les objets transitoires ne sont accessibles que par un pointeur ou par un autre objet.

Les objets **références** sont des objets qui encapsulent un objet qui a été créé par une application externe.

2.1.8.2. Architecture du modèle

Il s'agit ici de préciser de manière générale les liaisons qui existent entre les composants que sont les objets T.O.I., l'objet interface et les objets nommé, non-nommés et transitoires.

Les flèches représentent des échanges d'informations.

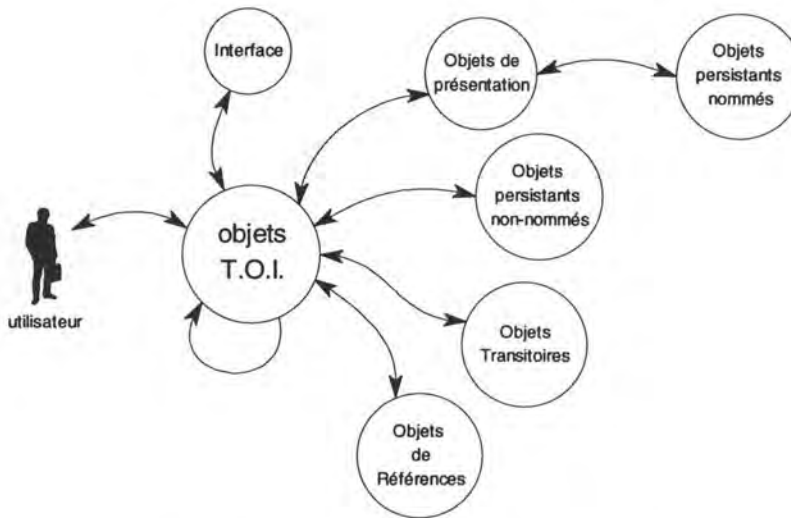


Fig. 2.28. Architecture du méta-modèle

2.1.8.3. Les mécanismes génériques

Ils sont représentés par un objet de commande qui, suivant le type de l'objet auquel il s'applique, interagit avec l'un ou l'autre des éléments du méta-modèle suivant le protocole de communication adéquat. Les commandes génériques offertes par le méta-modèle sont la création, la modification et la visualisation des objets.

Voyons comment elles fonctionnent pour les objets nommés, non-nommés, transitoires et de référence.

2.1.8.3.1. Les objets nommés

Pour chaque type d'objet nommé, on définit un type d'objet de présentation dont la fonction est d'interagir avec la couche dialogue¹¹.

Lors de sa création la commande reçoit le type de l'objet qu'elle va manipuler. Elle consulte l'objet interface pour obtenir des informations concernant ce type. Dans le cas d'objets nommés, l'objet interface envoie à la commande l'identifiant de l'objet de présentation à utiliser.

La création d'objets

La commande de création a questionné l'objet interface (1) et dispose de l'identifiant de l'objet de présentation à utiliser (2).

¹¹ Ce sont les services offerts par le TOI.

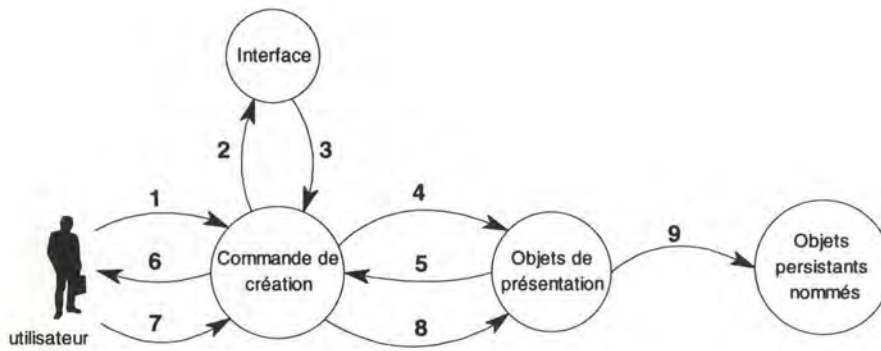


Fig. 2.29.

La commande de création demande à l'objet de présentation (4) les informations concernant les données qui doivent être fournies par l'utilisateur. Ces informations reprennent pour chacun des champs de données de l'objet les caractéristiques suivantes :

- les valeurs par défauts ;
- obligatoire ou optionnel¹² ;
- affichable à l'écran ;
- une note explicative du champ ;
- ...

Cet ensemble d'informations constitue un **descripteur** de champ.

En tenant compte des descripteurs reçus (5), la commande va inviter l'utilisateur à entrer ses informations en lui présentant un écran dont les spécifications se trouvent dans l'objet de présentation (6). L'utilisateur remplit les champs de saisie (7), la commande vérifie le contenu de ses champs avant de les rendre à l'objet de présentation (8) qui va effectivement créer l'objet dans la base de données(9).

La modification d'objet

La commande de modification utilise la méthode *getrevise* pour afficher les champs modifiables + les vérifications et le help pour les réviser. Ensuite elle utilise la commande *make revise* pour effectivement réaliser le changement lorsque tout a été vérifié.

¹² l'utilisateur est-il obligé de donner une valeur au champ ?

La visualisation d'objet

La commande de visualisation utilise la méthode *getshow* de l'objet de présentation pour savoir ce que l'on doit visualiser de l'objet à visualiser.

2.1.8.3.2. Les objets transitoires et non-nommés

Ils n'utilisent pas d'objet de présentation: la spécification du dialogue utilisateur est partie intégrante des objets de ce type. Les commandes utilisent directement leurs méthodes de présentation.

2.1.8.3.3. Les objets de référence

Une commande, qui a pour sujet un objet de référence, est traduite au niveau de l'interface par la création et l'activation d'un opérateur externe. Le but de ce dernier sera d'effectuer la commande en se servant de l'application externe 'mère', de l'objet encapsulé par l'objet de référence¹³.

2.2. L'environnement logiciel de WoorKs

Dans la partie précédente, nous avons présenté les modèles de WoorKs. Ils représentent la connaissance générique qui doit être adaptée en vue de représenter une situation de coopération particulière.

Dans cette partie, nous présenterons les éléments dynamiques de WoorKs. Ces éléments se divisent en trois catégories :

- les outils qui permettent de développer la connaissance générique ;
- les outils qui permettent d'utiliser la connaissance générique en vue de l'adapter à un cas particulier ;
- les outils qui supportent l'exécution d'un workflow particulier.

Les outils de ces trois catégories sont utilisés par des personnes différentes. Nous commencerons en présentant les fonctions remplies par ces personnes. Ensuite, nous

¹³ Voir le modèle des opérateurs et le modèle de l'information pour plus de renseignements concernant les opérateurs externes et les objets de référence.

verrons quels sont les outils proposés par WoorKs en vue de les assister dans l'exercice de leurs fonctions.

2.2.1. Les fonctions des acteurs

2.2.1.1. L'ingénieur d'application et le développeur d'application[LU 93]

Ces deux rôles sont remplis par des professionnels du traitement de l'information qui sont responsables de la faisabilité de workflows correspondant aux situations de coopération réelles. Le domaine d'application de WoorKs étant les situations de coopération répartie prédéfinie, ces personnes doivent veiller à ce qu'il soit possible de représenter le plus grand nombre de situations de ce type à l'aide des concepts de WoorKs.

L'**ingénieur d'application** est responsable de la connaissance générique; c'est lui qui va l'enrichir en vue de faciliter son utilisation, d'augmenter sa puissance sémantique, de veiller à sa cohérence, etc.(Fig. 2.30.).

Le **développeur d'application** est chargé de la mise en oeuvre d'un workflow particulier (infra 2.3. **Le processus de développement**) en utilisant la connaissance générique (Fig 2.30.). Il dialogue avec les clients en vue de leur fournir un workflow qui corresponde à la situation de collaboration qu'ils aimeraient intégrer au fonctionnement de l'entreprise.

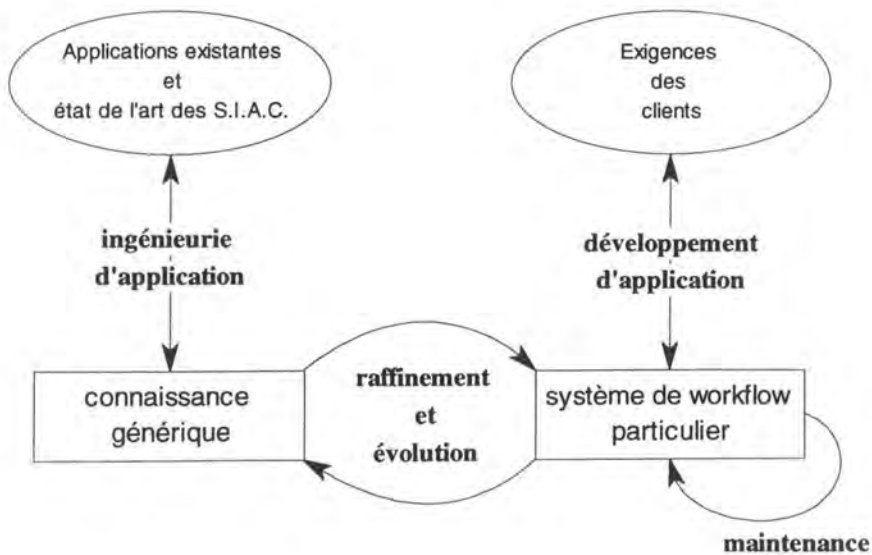


Fig. 2.30. Processus de développement d'un workflow particulier

Le développeur collabore avec l'ingénieur pour s'informer des fonctionnalités offertes par la connaissance générique et pour l'informer des fonctionnalités les plus souvent demandées par les utilisateurs.

Ex. : Les doyens des facultés aimeraient avoir un agenda commun qui les aiderait à fixer leurs rendez-vous. Si le concept d'agenda commun ne se trouve pas dans la connaissance générique, le développeur prévient l'ingénieur de cette lacune. L'ingénieur peut décider d'intégrer ce concept dans la connaissance générique, ce qui le mettrait à la disposition du développeur qui pourrait s'en servir pour un autre workflow particulier.

2.2.1.2. Les utilisateurs d'un workflow particulier

Il s'agit de l'ensemble des acteurs de l'organisation qui prennent part à l'exécution d'un workflow soit en tant que réalisateur, soit en tant que responsable.

Les **réalisateurs** effectuent les traitements non-automatisés qui font partie du workflow. Par exemple, remplir un bon de commande, rédiger une lettre de licenciement, etc.

Les **responsables** sont chargés de faire respecter les obligations des réalisateurs. Ces derniers sont censés remplir les objectifs opérationnels qui leur sont affectés dans les délais fixés.

2.2.2. Les supports de la réalisation des fonctions des acteurs

L'environnement WoorKs comprend des outils destinés à assister les acteurs dans l'exercice de leurs fonctions.

2.2.2.1. Les outils de gestion de la connaissance générique

WoorKs n'offre qu'un support très rudimentaire à la fonction de l'ingénieur d'application. Ce dernier dispose d'une série de documents décrivant sommairement le contenu de chaque modèle, et de Cool [Cool 92 (1)][Cool 92 (2)] qui est le langage de programmation dans lequel ils ont été élaborés.

2.2.2.2. Les outils de mise en oeuvre de la connaissance générique

WoorKs offre le langage A.D.L.(activity description language) [MOLLE 93] qui, comme son nom l'indique, est présenté comme étant capable de décrire la totalité d'un workflow particulier. Or, malgré une complexité d'utilisation non négligeable, il ne décrit que la structure des procédures, à savoir :

- la décomposition des procédures en activité ;
- le flux des documents en entrée et sortie des procédures et des activités.

Le développeur devra utiliser le langage de programmation Cool pour définir :

- la structure des documents utilisés ;
- les opérateurs qui constituent les activités ;
- la composition de l'organisation.

En fait, excepté le langage A.D.L. et son compilateur, le développeur d'application devra utiliser Cool. Il existe des documents d'information concernant le contenu et l'utilisation des modèles de WoorKs mais ils ne sont pas suffisants. Dès lors, le développeur devra parcourir le code pour être capable de définir un workflow particulier.

2.2.2.3. Les outils d'exécution d'un workflow

WoorKs comprend un grand nombre d'outils utilisés par les acteurs lors de l'exécution du workflow. Certains sont directement liés à l'exécution d'une activité. C'est le cas des traitements de texte, des tableurs, des éditeurs de graphiques, des logiciels de dessins, etc. D'autres sont des accessoires qui assistent les acteurs dans l'organisation de leur travail. C'est le cas de la corbeille (supra **2.1.7.4.3. L'exécution du workflow**), de l'agenda (supra **2.1.5. Le modèle du temps**) et du système de communication.

2.2.2.3.1. Les outils de réalisation

Les outils de réalisation font partie de la définition de la réalisation des activités. Ils sont soit des outils externes référencés par des opérateurs externes, soit des outils internes qui sont des opérateurs définis en Cool.

Le développeur d'application est responsable de la définition de la réalisation des activités. Suivant la rigueur de cette définition, le réalisateur sera plus ou moins fortement dirigé, plus ou moins fortement encadré. Ainsi, c'est le développeur d'application qui choisira

d'offrir de bon ou de mauvais outils au réalisateur. Néanmoins, l'usage des opérateurs externes en vue d'utiliser des outils externes assure un très bon environnement de réalisation, cela étant du aux faits suivants : le parc logiciels à la disposition du développeur d'application est vaste et la majorité de ces outils sont macro-commandable.

Les outils disposant d'un langage de description de macro-commande permettant au développeur d'application d'y faire appel en utilisant des macro-commandes en vue d'initialiser le cadre de travail dans lequel évoluera le réalisateur.

2.2.2.3.2. Les accessoires

Les accessoires forment l'environnement général dans lequel évolue les acteurs. La corbeille leur permet de prendre part à la réalisation des workflows et de se situer par rapport au travail qu'ils ont à faire. L'agenda leur permet d'organiser leurs rendez-vous, leurs congés, etc. Le système de communication leur permet d'échanger des informations avec d'autres acteurs de l'organisation.

2.3. Le processus de développement

Rappelons le but de WoorKs, qui est de limiter à un mois la durée du processus de développement d'un système de workflow particulier[SRIVA 92]. Il offre un ensemble d'objets génériques à partir duquel on développe le système qui correspond aux besoins d'un client.

2.3.1. Introduction

Précisons : il s'agit ici de la démarche suivie par le développeur d'application pour répondre aux besoins de clients en leurs offrant un système de workflow adapté. En fonction de leurs connaissances en informatique, et plus précisément dans le domaine des workflows, ils rédigeront leur demande de manières diverses. C'est pourquoi la première étape est l'identification du problème. Ensuite, il faut rechercher dans les modèles génériques les objets qui serviront de base ou qui seront réutilisés comme tels pour construire le système particulier. Ces objets serviront, suivant leur modèle d'origine, à la définition du QUI, du QUOI, du QUAND ou du COMMENT (infra **2.1. La connaissance générique de WoorKs**).

Ces deux grandes étapes seront illustrées par le cas d'élaboration du budget dans une institution universitaire [PETIT 93].

2.3.2. Première étape : Identification du problème

A partir de la demande du client et d'entretiens avec lui, le développeur doit préciser le problème auquel il est censé répondre en créant un système informatique. L'identification du problème comprendra une description en termes de qui, de quoi, de quand et de comment.

Pour illustrer cette étape et la suivante, nous avons choisi la partie de modification des formules budgétaires par l'administrateur délégué [PETIT 93]. Les formules budgétaires (Fig. 2.31.) sont les formulaires à partir desquels le doyen de chaque faculté rédige sa demande de budget ; cette demande est adressée à l'administrateur délégué.

Prévision des dépenses 1991

N° de l'article	Intitulé	Montant
202.1	Matériel et mobilier didactique	
202.2	Matériel et mobilier scientifique	
202.5	Matériel informatique et logiciels	
.....	
	TOTAL	

Fig. 2.31. Une formule budgétaire

Chaque début d'année, les doyens ont le droit de demander une modification de la formule en raison de son inadéquation par rapport aux demandes à exprimer. Par exemple, s'il veut acheter un animal, il demande à l'A.D. (administrateur délégué) de créer une rubrique dans laquelle il pourra mentionner son achat (209.5. achat d'animaux domestiques).

L'A.D. fixe une date à partir de laquelle les demandes de modification ne seront plus prises en compte. Une fois cette date arrivée, il examine l'ensemble des suggestions de modification qu'il a reçu pour établir la nouvelle formule budgétaire qui sera distribuée à l'ensemble des doyens. Cette révision se fait sur base des demandes reçues les années précédentes, des justifications accompagnant chaque suggestion de modification et des changements acceptés les années antérieures.

Cette partie du workflow va illustrer le travail réalisé par le développeur d'application pour répondre aux besoins de l'A.D..

2.3.2.1. Identification des besoins du client

L'A.D. voudrait un système qui lui permettrait d'analyser rapidement les demandes de modification et de modifier aisément les formules. En outre, ce système pourrait gérer le flux de documents entre lui et les doyens.

2.3.2.2. Spécification du traitement

Entrées et sorties

Les entrées sont les suggestions de modification des doyens avec pour chacune d'elles, une justification. Les sorties sont soit un nouveau formulaire de demande budgétaire qui sera distribué à tous les doyens, soit l'ancienne formule dans le cas où aucune demande de modification n'aurait été prise en compte.

Données internes

Les données internes sont les informations qui doivent être à la disposition de l'A.D. lors de la réalisation de sa tâche. Elles se composent de l'historique des exécutions des tâches antérieures, à savoir : les modifications acceptées ou refusées, les modifications refusées mais qui valent la peine d'être réexaminées, etc.

Les outils de réalisation

Pour effectuer sa tâche, l'A.D. disposera d'un outil d'analyse des demandes et d'un outil de manipulation de la formule budgétaire. Les fonctionnalités de l'outil d'analyse sont les suivantes :

- analyse par poste, selon le type (éclatement de poste, nouveau poste, modification de libellé de poste, destruction de poste) et selon la provenance ;
- éliminer directement certaines demandes (cf. la demande de création d'un poste pour achat d'animaux domestiques) ;
- marquer les demandes judicieuses en vue d'une réexamination.

Les fonctionnalités de l'outil de manipulation de la formule budgétaire sont les suivantes :

- insérer un poste ;
- supprimer un poste ;
- éclater un poste.

2.3.3. Deuxième étape : Utilisation de la connaissance générique

Après avoir parcouru l'ensemble des objets de la connaissance générique, le développeur d'application identifie ceux qu'il va utiliser comme tels et ceux qu'il devra spécialiser pour spécifier le traitement particulier qui nous préoccupe.

2.3.3.1. Utilisation du modèle organisationnel

Le développeur réutilise l'objet acteur pour représenter les personnes jouant le rôle de doyen et la personne qui joue le rôle d'A.D.. Il spécialise l'objet de rôle pour représenter le rôle de doyen et d'administrateur délégué.

2.3.3.2. Utilisation du modèle informationnel

Les informations qui seront utilisées sont : les suggestions de modification envoyées par les doyens, leurs justifications et les formules de demandes budgétaires. Elles seront représentées en spécialisant l'objet *information* qui contient les informations nécessaires pour identifier l'auteur et le propriétaire du document.

L'objet *modificationsformule* est une liste de records qui comprennent une modification et sa justification. Une modification est un record qui est un type et un contenu : le type est un éclatement de poste ou une destruction de poste ou ... ; le contenu aura une structure différente suivant le type de la modification¹⁴. La justification est exprimée textuellement¹⁵.

L'objet *formule* est un ensemble de record constitué d'un numéro d'article, d'un intitulé et d'un montant (supra Fig.2.31.).

¹⁴ Nous ne présentons pas précisément les caractéristiques des objets utilisés car le but de cette présentation du processus de développement est de donner une idée du processus plutôt que d'apporter une solution complète à l'exemple utilisé.

¹⁵ Il est possible d'utiliser des documents photo ou sonore mais, une fois encore, cela compliquerait la présentation de ce processus.

2.3.3.3. Utilisation du modèle du temps

L'ensemble de la procédure est soumise à des contraintes temporelles qui seront représentées en utilisant les objets offerts par le modèle temporel. Par exemple, l'administrateur délégué peut demander au développeur d'application que soient refusées toutes les demandes de modifications présentées après une date limite.

2.3.3.4. Utilisation du modèle des procédures

Il permet de définir la partie de la procédure d'élaboration du budget qui correspond à la modification des formules budgétaires.

L'objet *axformule* est créé en spécialisant l'objet activité. Il comprendra comme attributs les documents reçus en entrée (la liste des modifications de formules) et les documents qui seront produits en sortie. Les sorties seront définies par deux alternatives de sorties : la première, *altinchangé*, n'a pas de résultats car elle ne correspond à aucune modification des formules budgétaires; la seconde, *altchangé*, a comme résultat la nouvelle formule budgétaire.

Cette nouvelle activité sera mise à la disposition de l'A.D. via sa corbeille, après la réalisation des activités qui correspondent à l'élaboration des demandes de modification effectuée par les doyens.

L'activité *axformule* a deux alternatives de sorties : une dans le cas où l'A.D. a modifié la formule de demande et l'autre dans le cas où il n'y a eu aucune modification. Le choix de l'alternative de sortie aura des conséquences sur la suite de l'exécution de la procédure: dans le cas de la modification de la formule, la nouvelle formule sera envoyée à tous les doyens; dans l'autre cas, ils réutiliseront la formule qu'ils ont utilisés l'année dernière.

2.3.3.5. Utilisation du modèle des opérateurs et du méta-modèle

2.3.3.5.1. Introduction

Le modèle des opérateurs permet de définir les traitements qui seront effectués par l'activité *axformule*. Cette activité est un ensemble d'outils qui sont utilisés par son réalisateur, à savoir : l'administrateur délégué. Pour lui permettre d'entrer en contact avec le système, il faut définir une interface grâce à laquelle il pourra manipuler ces outils. On développe cette interface en utilisant les objets du méta-modèle.

La phase de spécification du traitement a identifié deux outils : un outil d'analyse des demandes et un outil de manipulation de la formule budgétaire. Nous n'allons pas implémenter l'ensemble des fonctionnalités de ces outils mais nous allons simplement essayer d'en donner une idée en présentant les éléments importants.

2.3.3.5.2. L'interface utilisateur

L'interface utilisateur comprend deux types d'objets : les objets de manipulation des objets qui permettent de les montrer, de les modifier, de les effacer (supra **2.1.8. Le méta-modèle**) et les objets de définitions des outils de manipulation à savoir l'interface graphique.

Le méta-modèle et la manipulation des objets

L'initialisation de l'objet interface

Chaque nouveau type d'objet qui a été défini devra être décrit en définissant un méta-type qui se placera dans la hiérarchie de l'objet interface. Cela permettra notamment aux commandes génériques de création, de modification et de destruction d'appeler les bonnes méthodes des bons objets pour manipuler l'objet auquel on s'intéresse.

Les objets de présentation

Les objets d'information , *modificationsformule et formule*, définis par le modèle de l'information, sont des objets nommés (supra **2.1.8. Le méta-modèle**). Pour chacun de ces objets, on définit un objet de présentation (*modificationsformulep et formulep*) dont la fonction est de définir la forme des attributs qui seront affichés (la taille des caractères, l'orientation, l'espace d'affichage, etc.), les attributs que l'on peut modifier, les méthodes de vérification des modifications, etc.

Les objets de commandes

Les objets de présentation *modificationsformule et formule* sont utilisés par les commandes génériques de création, de modification et de destruction. Ces objets de commande seront utilisés par les opérateurs pour créer un objet, le modifier ou le détruire.

Les objets T.O.I. et la création de l'interface graphique

Les deux outils présentés doivent offrir une interface qui permettra à l'utilisateur de les manipuler. Parmi les objets de définition de l'interface graphique, on trouve des objets de structuration du dialogue, tels que des menus, et des objets qui permettent l'exécution d'un opérateur, par exemple des items de menu ou des boutons poussoirs. L'ensemble de ces objets est construit en utilisant les objets T.O.I..

Cette interface est présentée à l'utilisateur lorsqu'il utilise sa corbeille (supra 2.1.4.7.3. **L'exécution d'un workflow**) pour démarrer l'activité *axformule*. On peut, par exemple, définir deux fenêtres dans lesquelles s'exécuteraient les deux outils. Ainsi, l'administrateur pourrait simultanément analyser les demandes et modifier la formule budgétaire.

2.3.3.5.3. Définition des opérateurs

Les opérateurs sont associés à certains des objets du framework T.O.I. Par exemple, on peut avoir un item de menu dont la sélection engendre l'appel d'un sous-menu et un autre item dont la sélection engendre la création et le démarrage d'un opérateur qui réalise une fonctionnalité propre aux outils qui ont été définis (infra : spécification du traitement). Cette fonctionnalité sera, par exemple, l'éclatement d'un poste de la formule budgétaire.

La définition des opérateurs se fait en utilisant le langage de programmation Cool [COOL 92 (1)][COOL 92 (2)]. Il utilise les méthodes offertes par les objets de commande générique pour manipuler les objets d'information, les créer, les modifier, les détruire ou les visualiser.

Prenons, par exemple, la fonctionnalité d'éclatement de poste de l'outil de manipulation de la formule budgétaire. Elle est implémentée par un opérateur qui sera créé et exécuté lors de la sélection par l'administrateur délégué de l'outil T.O.I. auquel il est associé. Par exemple, après avoir placé le curseur sur le poste à éclater, il sélectionne le bouton poussoir d'éclatement de poste. Une fois créé et démarré, on passe à l'opérateur le poste sur lequel se trouve le curseur; l'opérateur utilise l'outil de commande associé aux postes pour le détruire puis il crée deux nouveaux postes. Pour créer les deux nouveaux postes, l'opérateur utilise l'objet de commande qui utilisera l'objet de présentation pour connaître les attributs qui doivent être initialisés par l'utilisateur. Dans ce cas-ci, il s'agit de l'intitulé du poste. L'administrateur devra donner un intitulé à chacun de ces deux nouveaux postes.

Notre but n'étant pas, ici, de donner une description détaillée de tous les éléments de ce processus de développement mais plutôt d'en donner une idée, nous ne précisons pas la spécification et l'implémentation des autres fonctionnalités des outils.

Chapitre 3 :

ACTA

3.1. Introduction

Dans la partie consacrée à la problématique des S.I.A.C., nous avons cité quelques systèmes de support des situations de coopération réparties et ad hoc parmi lesquelles on trouve le système ACTA, en cours de conception à l'institut d'informatique de Namur. Nous commencerons par présenter ses objectifs pour ensuite aborder les moyens utilisés. La présentation des objectifs comprendra une définition et une présentation des particularités des situations prédéfinies et ad hoc. La présentation des moyens utilisés sera relative à la description de la situation de coopération et aux mécanismes d'interactions utilisés par les acteurs pour coordonner leurs actions.

3.2. L'objectif

Dans son état actuel de développement, ACTA est un modèle descriptif de la connaissance nécessaire à la gestion des situations de coopération répartie et ad hoc. A terme, ce modèle devrait donner lieu à un environnement destiné à supporter ce type de situation. Dans le chapitre premier nous avons brièvement abordé la description de ce type de situation(supra **1.4.1. La coopération**). Nous la compléterons en vue d'identifier aussi clairement que possible les types de fonctionnalités qu'il est nécessaire de mettre en oeuvre pour supporter les situations de coopération répartie et ad hoc.

3.2.1. Coopération répartie

' La **coopération** peut être qualifiée de **répartie** si la tâche collective cible est réalisée par le biais d'une division de la tâche en sous-tâches distinctes affectées chacune à un sous-ensemble des acteurs impliqués, en fonction par exemple, de compétences ou spécialisations particulières.'

Dans ce cas, la **contribution individuelle** des acteurs à la réalisation de la tâche collective cible est **différenciée**, puisque confinée au contexte de la(des) sous-tâche(s) qui lui(leur) est (sont) affectée(s) : [...] [PETIT 94].

Lorsque la tâche collective cible est réalisée dans le cadre d'une situation de coopération répartie, on parlera, par abus de langage, de tâche répartie. (Fig. 3.1.)

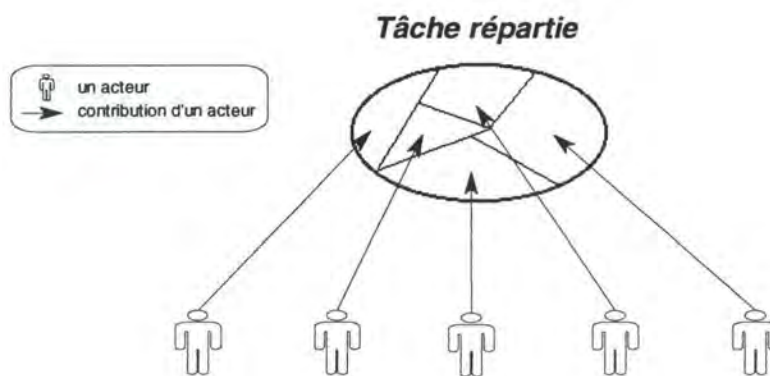


Fig. 3.1. Représentation d'une tâche répartie

La tâche étant divisée en sous-tâches distinctes, nous pourrions répartir la responsabilité de l'ensemble de la tâche répartie en attribuant la responsabilité des sous-tâches aux acteurs chargés de la réalisation de celles-ci. Cependant, le manque de rigueur de ces derniers ne nous le permet pas. C'est pourquoi, par rapport à une tâche, on distinguera deux rôles : celui de responsable et celui de réalisateur.

Comme les sous-tâches font partie d'une tâche répartie commune et que pour chacune d'elles il existe un responsable et un réalisateur, il faut définir des mécanismes de coordination qui assureront le bon déroulement de l'exécution de la tâche répartie.

Si on analyse les relations liant le(s) responsable(s) et le(s) réalisateur(s) d'une (de plusieurs) sous-tâche(s), on identifie deux niveaux de coordination : la coordination externe et la coordination interne.

3.2.1.1. La coordination externe

Elle est à charge des responsables de tâches et elle se divise en deux catégories d'activités: celles de construction et celles de gestion de la situation de coopération. Les activités de construction sont :

- la **division** de la tâche cible en sous-tâches ;
- la **planification** qui vise à coordonner le travail des réalisateurs en leurs imposant principalement des contraintes temporelles telles que des échéances de réalisations ou des conditions de déclenchement ;
- la **délégation** qui vise à attribuer la réalisation des tâches spécifiées par les activités de division et de planification.

Les activités de gestions de la situation de coopération sont :

- la **consolidation** qui vise à vérifier, compléter, consolider les résultats des différentes sous-tâches constitutives de la tâche répartie de manière à fournir les résultats de cette dernière ;
- la **gestion des contrats** qui vise à maintenir la cohérence de l'ensemble des contrats induits par l'activité de délégation ;
- le **suivi d'avancement** qui vise à assurer le respect des échéances de réalisation en contrôlant le travail des réalisateurs.

3.2.1.2. La coordination interne

Elle est à charge des réalisateurs des tâches et elle couvre les mécanismes par lesquels ces réalisateurs coordonnent leurs activités, eu égard aux interdépendances induites par la coordination externe. La coordination interne consiste à :

- si possible, respecter strictement les échéances de réalisations et sinon, à négocier de nouvelles échéances acceptables pour les réalisateurs dépendant de ce report¹ ;
- mettre à disposition les informations nécessaires à la réalisation des tâches en aval.

¹S'il y a conflit entre les réalisateurs, il sera réglé par le(ou les) responsable(s) concerné(s) par le biais d'une reconduction des activités de coordination externe.

3.2.2. La coopération répartie et ad hoc

' Une situation de **coopération ad hoc** est [...] caractérisée par l'inexistence de "procédure" de réalisation de la tâche répartie, soit parce que celle-ci correspond à une situation nouvelle (problème nouveau, opportunité nouvelle), soit parce que, indépendamment de son caractère potentiellement répétitif, aucune règle organisationnelle ne guide sa réalisation.' [PETIT 94].

Dès lors, la situation de coopération sera définie dynamiquement par les acteurs en exerçant les activités de construction propres à la coordination externe. Ils divisent la tâche cible en sous-tâches dont ils planifient l'exécution et dont ils attribuent la réalisation aux acteurs compétents.

En tant que situations de coopération répartie, les situations de coopération répartie et ad hoc sont basées sur les concepts de responsable, et de réalisateur et sur les mécanismes de coordination externe et interne. Leur définition dynamique et donc leur caractère ad hoc se traduit par des adaptations particulièrement sensibles au niveau de la coordination externe.

3.2.2.1. Les activités de construction de la situation de coopération

Etant réalisées de manière ad hoc, les activités de constructions 'nécessitent une **implication active, effective et réfléchie** du responsable'[PETIT 94].

Le responsable qui est confronté à une nouvelle situation ne dispose que d'une faible expérience organisationnelle pour l'aider à mener à bien les activités de **division**, de **planification** et de **délégation**. La validité du plan de réalisation ainsi construit est mise en cause par ;

- la pertinence des sous-tâches identifiées pour atteindre l'objectif associé à la tâche cible ;
- l'objectivité dans la fixation de leurs délais de réalisation ;
- l'évaluation des compétences effectives des acteurs auxquels la réalisation des tâches a été affectée.

Excepté le manque d'expérience organisationnelle propre aux situations ad hoc, les activités de division et de planification sont similaires pour les situations de coopération répartie et prédéfinie (supra **1.4.1. La coopération**). Par contre, l'activité de délégation liée aux situations ad hoc se différencie par l'obligation d'obtenir **l'accord explicite** de

l'acteur auquel un responsable désire affecter la réalisation de la tâche qu'il a spécifiée. Cette obligation découle de l'absence de règles organisationnelles d'affectation des sous-tâches. Dès lors, nous considérons comme non souhaitable le fait qu'un acteur puisse imposer la réalisation d'une tâche à un autre acteur et ce, indépendamment de leurs positions hiérarchiques respectives.

3.2.2.2. Les activités de gestion de la situation de coopération

L'inexistence de règles organisationnelles de validation du plan de réalisation défini durant l'activité de construction de la situation de coopération a pour conséquence une augmentation de la charge que représentent les activités de gestion.

Le plan de réalisation initial sera souvent remis en question par l'impossibilité pour certains acteurs, d'honorer leurs contrats de réalisation. Il peut s'agir de non-respect de l'échéance de réalisation, d'incompétence entraînant l'impossibilité de réalisation, d'indisponibilité des informations nécessaires pour entamer la réalisation, etc.

Du fait des interdépendances qui existent entre les sous-tâches, il existe des interdépendances entre les contrats qui, lors de toute modification, devront être gérées en vue de garder l'ensemble cohérent. Ainsi, l'activité de **gestion des contrats** représente une charge plus importante car les modifications du plan de réalisation engendrent presque toujours une modification d'un ou de plusieurs contrats liant les responsables et les réalisateurs.

L'activité de **suiti d'avancement** représente une charge plus importante car l'incertitude concernant la validité des échéances de réalisation imposées aux réalisateurs nécessite une plus grande attention des responsables. Le rôle de ceux-ci sera d'adapter le plan de réalisation à la situation réelle. Par exemple, si nous demandons à un acteur de rédiger un rapport, il se peut que, ne disposant que d'une faible expérience concernant l'évaluation de la complexité liée à la réalisation de ce rapport, nous sous-estimions le temps nécessaire à sa réalisation. C'est pourquoi, il faudra que nous adaptions les obligations du réalisateur en vue de fixer une échéance plus raisonnable.

3.3. Les moyens

Nous débuterons cette partie par une présentation des idées relatives, d'une part, au modèle descriptif de la situation de coopération et ,d'autre part, aux mécanismes d'interaction propres à l'activité de délégation. Ces idées seront suivies par une présentation visant à

introduire les principaux concepts choisis pour supporter les situations de coopération répartie ad hoc.

3.3.1. Les idées

3.3.1.1. Le modèle descriptif de la situation de coopération

' [...] le modèle proposé repose sur la différenciation d'un certain nombre de mécanismes de division d'une tâche qui se concrétisent dans le modèle par des types de relations entre **tâches et sous-tâches**. Ces types de relations ont une définition sémantique précise sur base de laquelle ont été dégagées un ensemble de règles permettant, d'une part, la validation de la description de la situation de coopération (règles de contrôle) et d'autre part, la gestion de la progression d'une tâche, du fait de la progression de ses sous-tâches (règles de comportements).' [PETIT 94].

L'introduction de relations entre tâche et sous-tâches engendre une représentation arborescente de la réalisation de la tâche cible. Prenons l'exemple de la préparation d'un festival musical : il faut déterminer la date, choisir l'endroit, inviter les artistes, assurer l'encadrement et faire la promotion du festival. La réalisation de la tâche cible peut être représentée par la figure 3.2.

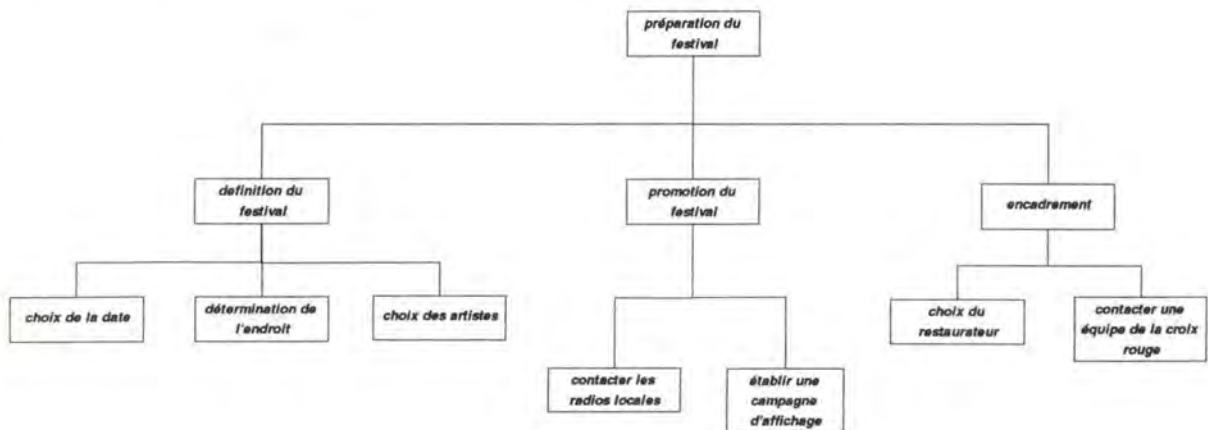


Fig. 3.2. représentation de l'organisation d'un festival musical

Ce type de représentation reflète les différentes étapes de construction de la situation de coopération. Dans notre exemple, on a commencé par identifier la tâche de préparation du festival. Ensuite, on l'a divisée en tâches de définition, de promotion et d'encadrement, chacune de ces tâches pouvant elle-même être subdivisée. Chaque étapes est réalisée par

un acteur qui, étant réalisateur d'une tâche de niveau n crée des sous-tâches de niveau $n+1$ pour lesquelles il assurera au moins le rôle de responsable et peut-être le rôle de réalisateur.

De cette manière on représente la contribution effective de chaque acteur au processus de construction. Cette connaissance permet '[...] une **adaptation dynamique** du support fourni à chaque acteur en fonction de la modification de son rôle dans le contexte de la situation de coopération '[PETIT 94]'

3.3.1.2. Conception des mécanismes d'interaction

L'affectation de la réalisation des tâches prédéfinies se fait en appliquant des règles organisationnelles. Dans le cas de tâches ad hoc, elle se fait par consentement explicite du candidat réalisateur. Il y a une interaction effective entre le délégant (le responsable) et le délégataire (le réalisateur).

La première chose que doit faire le délégant consiste à rechercher, dans l'organisation, un acteur dont les compétences lui permettront de réaliser la tâche qu'il a définie. Dès lors, il est nécessaire de représenter la connaissance relative aux acteurs de l'organisation pour permettre au délégant de faire son choix. Cette connaissance est confinée dans le modèle des acteurs (infra **3.3.2.2. Le modèle des acteurs**).

La deuxième chose consiste à interagir avec le délégataire potentiel. Dès lors, on identifie des mécanismes de délégation supportés par des protocoles de conversation. Ces derniers seront construits de manière à favoriser l'aboutissement rapide (avec ou sans succès) du processus de délégation.

' Le lecteur reconnaîtra dans cette option, l'optique préconisée par l'approche conversationnelle [...]. Soulignons cependant que l'idée de différencier et de faire coexister, dans un même environnement, la connaissance relative à la situation de coopération et celle "transportée" dans les conversations de délégation, dissipe les limites intrinsèques à cette approche. ' [PETIT 94]

3.3.2. ACTA : un modèle de la coopération logistique

Cette partie est entièrement basée sur les éléments présentés dans la partie consacrée aux idées. Celles concernant la description de la tâche cible ont donné naissance au **modèle d'organisation des tâches**. Les idées concernant la répartition de la tâche cible ont donné naissance au **modèle des acteurs** et au **modèle des conversations**.

Le but de cette partie n'est pas de donner une description détaillée de tous les objets et des relations qui les relient; il s'agit plutôt d'en donner une idée pour asseoir la présentation faite dans la partie précédente. Pour plus de renseignements, on consultera [PETIT 94].

3.3.2.1. Le modèle d'organisation des tâches

3.3.2.1.1. Contenu et objectif

Il s'agit d'un modèle de description des situations de coopération répartie ad hoc basé sur les concepts d'acteur et de tâche (infra **3.3.2.1.2. Les concepts du modèle**). Il comprend également des règles de contrôle et de guidance qui assurent la cohérence de la représentation de la situation de coopération.

Une **règle de contrôle** accompagne et contrôle l'acteur qui spécifie une situation de coopération. Les règles de contrôle veille à ce que la définition et les propriétés des concepts utilisés soient respectés.

Une **règle de comportement** assure la cohérence de l'évolution du modèle de la réalisation de la tâche cible. C'est une règle automatique qui régit le comportement dynamique des objets de la spécification.

L'objectif est de décrire précisément la situation de coopération répartie ad hoc, de manière à pouvoir fournir aux acteurs un support aux activités de coordination interne et externe. Les concepts et relation du modèle d'organisation des tâches s'appuient sur les caractéristiques propres aux mécanismes de coordination (figure 3.3.) :

- l'activité de **division** est entreprise par un acteur 'concepteur' qui, à partir d'une tâche (sujet), crée des sous-tâches (résultat) pour lesquelles il jouera le rôle de concepteur ;
- l'activité de **planification** définit les relations de séquençement et les échéances de réalisation des tâches ;
- l'activité de **délégation** permet au responsable d'une tâche d'en affecter la réalisation à un tiers de l'organisation. Eu égard au rôle joué par les acteurs dans le processus de délégation, on baptisera le premier le délégant et le second le délégataire.

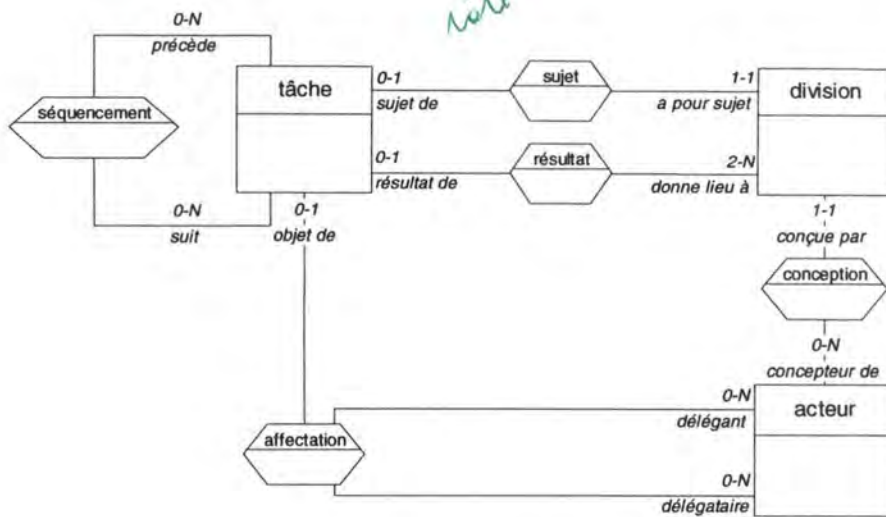


Fig. 3.3. Représentation synthétique du modèle d'organisation des tâches
[PETIT 94]

3.3.2.1.2. Les concepts du modèle

Nous allons préciser les concepts de base que sont **la tâche** et **l'acteur**.

Le concept d'acteur

Il s'agit du concept commun au modèle d'organisation des tâches et au modèle de représentation de l'organisation. Nous limitons sa présentation à sa définition (infra 3.3.2.2. **Le modèle des acteurs**).

'Un **acteur** est un individu ou un ensemble d'individus assumant une ou plusieurs fonctions dans l'organisation. Etant reconnu comme élément de l'organisation, tout acteur est susceptible d'être impliqué dans une situation de coopération.' [PETIT 94].

Le concept de tâche

Définition

'Une **tâche** correspond à une unité de traitement définie par un objectif opérationnel et dont la réalisation est régie par des contraintes temporelles.' [PETIT 94].

Caractérisation et spécification d'une tâche

Elle est identifiée par un nom logique et caractérisée par :

- l'objectif opérationnel la définissant ;
- une description des types de résultats attendus ;
- ses conditions de déclenchement qui spécifient le moment à partir duquel on peut commencer sa réalisation ;
- son échéance de réalisation qui spécifie le moment à partir duquel sa réalisation devrait être terminée ;
- un état de progression ;
- son concepteur, à savoir l'acteur qui exerce les activités de coordination externe.

Cycle de vie

Une tâche est un élément dynamique dont la progression peut être représentée par une succession d'états remarquables, le passage de l'un à l'autre étant provoqué par la survenance d'événements remarquables (Fig 3.4.).

Les **événements remarquables** [PETIT 94] sont la création, l'affectation, le déclenchement, la prise en charge et la terminaison.

- La *création* résulte de l'identification de l'objectif opérationnel qui la définit. Après sa création la tâche est dite **définie**.
- L'*affectation* résulte de l'aboutissement favorable d'un processus de délégation. Elle consiste à attribuer la responsabilité de réalisation d'une tâche à un acteur de l'organisation qui la réalisera lui-même ou qui initialisera une nouvelle situation de coopération dont il sera responsable. Après son affectation, la tâche est dite **affectée** si elle n'a pas été déclenchée; elle est dite **réalisable** dans le cas contraire.
- Le *déclenchement* résulte de la vérification de ses conditions de déclenchement. Après son déclenchement, la tâche est dite **déclenchée** si elle n'a pas été affectée; elle est dite **réalisable** dans le cas contraire.

n.b. : les deux chemins parallèles qui mènent de l'état **définie** à l'état **réalisable** représentent le fait que l'*affectation* d'une tâche peut avoir lieu avant ou après son *déclenchement*.

- La *prise en charge* résulte, soit de la décision de son délégataire (acteur auquel elle a été affectée) de commencer sa réalisation, soit de la prise en charge d'une de ses sous-tâches. Elle fait passer la tâche de l'état **réalisable** à l'état **en cours**;
- La *terminaison* résulte de la satisfaction de l'objectif opérationnel de la tâche. La terminaison fait passer la tâche de l'état **en cours** à **terminé**.

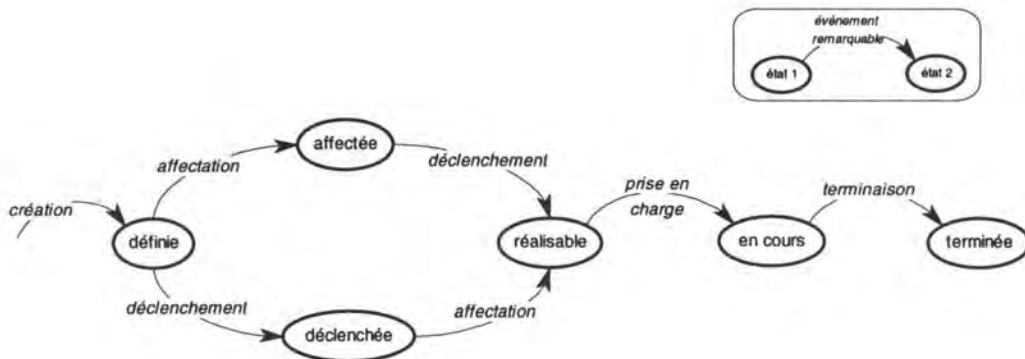


Fig. 3.4. Cycle de vie d'une tâche

3.3.2.1.3. Les relations du modèle

Le modèle comprend les relations de conception et d'affectation définies entre une tâche et un acteur, ainsi que les relations de séquençement et de division définies entre les tâches. Pour chacune de ces relations nous donnerons une définition et nous aborderons leurs mécanismes de génération.

La relation de conception

Définition

'Une **relation de conception** définie entre un acteur A et une tâche T exprime que la tâche T a été créée et définie par l'acteur A.' [PETIT 94]

Mécanismes de génération

La création d'une tâche résulte soit de l'identification d'un objectif opérationnel indépendant, c'est le cas de la création d'un nouveau plan de réalisation, soit de l'identification d'un sous-objectif opérationnel, c'est le cas de la création des sous-tâches. Ainsi, la relation de conception résulte du processus de création de

nouveaux plans de réalisation, de la division d'une tâche(infra **La relation de décomposition**), du greffage d'une tâche(infra : **La relation de greffage**) ou, enfin, de la duplication simple (infra : **La relation de duplication simple**) d'une tâche.

La relation d'affectation

Définition

'Une **relation d'affectation** définie entre une tâche T et un acteur A exprime que la tâche T est affectée à l'acteur A, c'est à dire que l'acteur A est considéré, par le concepteur de T comme responsable de la tâche T. La responsabilité d'un acteur par rapport à une tâche consiste à faire en sorte que la tâche soit correctement réalisée , c'est à dire que les résultats attendus soient produits selon l'échéance fixée. L'acteur affecté à une tâche n'est pas tenu d'assumer la responsabilité qui lui incombe en réalisant lui-même la tâche.'[PETIT 94]

Mécanismes de génération

Les relations d'affectation sont générées par l'aboutissement favorable d'un processus de délégation.

Définition

Un processus de délégation est initialisé par le concepteur d'une tâche non-affectée et vise un ou plusieurs acteurs. Son but est d'attribuer la responsabilité de la réalisation d'une tâche en obtenant l'accord explicite de l'un ou de plusieurs des acteurs visés(supra **3.2.2.1. Les activités de construction de la situation de coopération**).

Modes de délégation dynamique

L'acteur qui désire affecter la réalisation d'une tâche a le choix parmi 5 types de processus de délégations : l'auto-délégation, l'assignation négociable, l'assignation non-négociable, l'appel d'offre négociable et l'appel d'offre non négociable.

L'**auto-délégation** permet à cet acteur de s'auto-affecter la responsabilité de la réalisation de la tâche qu'il a créé.

Le **caractère négociable** de la délégation d'une tâche porte sur l'objectif opérationnel et l'échéance de réalisation de cette dernière. Dès lors, les concepteurs peuvent utiliser des processus de délégation non-négociable, négociable sur le contenu et/ou sur l'échéance de réalisation.

L'**assignation** permet au concepteur de solliciter un délégataire potentiel préalablement choisi. Le choix a été fait sur base de connaissances personnelles et/ou sur base de la connaissance organisationnelle relative aux compétences des acteurs. Si le délégataire potentiel accepte la responsabilité de réalisation de la tâche objet du processus de délégation, il devient automatiquement le délégataire effectif.

L'**appel d'offre** lui permet de solliciter un ensemble de délégataires potentiels. Contrairement à l'assignation, le choix de ceux-ci fait partie du processus de délégation. Le délégant détermine un profil type du délégataire potentiel sur base duquel sera constitué l'ensemble des acteurs qui seront contactés. Ensuite, le délégant recueillera les réponses positives en vue de sélectionner le (ou les)² délégant(s) potentiel(s) qui sera (seront) réellement affecté(s) à la tâche objet du processus de délégation.

Concept dérivé : le contrat

L'affectation d'une tâche à un acteur lie ce dernier avec le concepteur de la tâche. Ce lien est représenté par le contrat.

'Un **contrat** est une agrégation structurée d'informations reflétant, **au terme de l'affectation d'une tâche**, les obligations du délégataire vis-à-vis du délégant c'est-à-dire du concepteur, de cette tâche.' [PETIT 94].

La relation de séquençement

Définition

Une **relation de séquençement** relie une tâche origine à une tâche cible et exprime une dépendance de résultats de la tâche cible par rapport à la tâche origine. Dès lors,

² Le choix de plusieurs acteurs au terme d'un appel d'offre donne lieu à une situation de coopération par affectation multiple représentée par une relation de duplication particulière non abordée dans le contexte de ce travail [PETIT 93].

on ne peut entamer la réalisation de la tâche cible avant la terminaison de la tâche origine. En ce sens cette dernière définit une condition de déclenchement de la tâche cible.

Spécification

Les relations de séquençement sont spécifiées par le concepteur des tâches origine et cible. Cette dernière ne doit pas avoir été affectée, car si c'était le cas, l'ajout d'une condition de déclenchement modifierait le contrat liant délégrant et déléataire(supra **La relation d'affectation**).

La relation de décomposition

C'est une relation de structuration des traitements qui résulte de la division de l'objectif opérationnel en sous-objectifs de complexité moindre.

Définition

'Une tâche T_i sera dite **décomposée** si elle est divisée en sous-tâches [...] correspondant à des unités de traitement :

- guidées par des objectifs opérationnels distincts et de complexité moindre que celui définissant T_i ;
- dont l'union constitue le traitement réalisant T_i . '[PETIT 94].

Les interdépendances existant entre sous-tâches et tâche 'mère' sont complexes. Nous pouvons dire, pour simplifier, que le cycle de vie de la tâche 'mère' comprend le cycle de vie de ses filles. Nous introduisons le sujet en présentant une idée des règles régissant le déclenchement et la terminaison de la tâche 'mère'. Nous invitons le lecteur à consulter [PETIT 94] pour plus de renseignements.

Exemples de règles

La **condition de déclenchement** de la tâche décomposée contraint le déclenchement de l'ensemble de ses sous-tâches. Les conditions de déclenchement propres aux sous-tâches seront identiques ou plus contraignantes que celle de leur 'mère'.

Si la tâche mère n'a pas de condition de déclenchement, elle sera déclenchée par le déclenchement d'une de ses sous-tâches.

La **terminaison** de la tâche décomposée est et la terminaison de l'ensemble de ses sous-tâches sont simultanées.

Représentation graphique

La décomposition d'une tâche cible est représentée par un arbre dont la racine représente la tâche cible et dont les feuilles sont les sous-tâches (supra Fig. 3.2.).

Mécanisme de génération

Le relation de décomposition est définie par le concepteur d'une tâche qui n'a pas été affectée ou par le délégataire d'une tâche dont l'état est affectée ou réalisable.

La relation de greffage

Définition

Lorsqu'en cours de réalisation d'une tâche Tj, un objectif opérationnel complémentaire à celui de Tj mais utile, voire nécessaire à sa réalisation est identifié, nous dirons que la tâche Ti correspondant au traitement auxiliaire concrétisant la réalisation de ce nouvel objectif se greffe sur Tj.

Tj sera appelée **tâche greffée** et Ti **tâche greffon**. [PETIT 94].

Mécanisme de génération

La tâche greffon est le fruit de l'identification d'un sous-objectif complémentaire de la tâche greffée. La tâche greffon permet de préciser partiellement la spécification de la réalisation de la greffée. "Partiellement", car contrairement à la relation de décomposition, la réalisation de la tâche greffée n'est pas remplacée par la réalisation des tâches greffons, mais elle est complétée par celles-ci.

L'identification d'un sous-objectif complémentaire d'une tâche peut se faire lorsque son état d'avancement est antérieur à *terminée* et supérieur à *en cours*.

La relation de duplication simple

Elle permet aux acteurs de déléguer totalement la réalisation de la tâche à laquelle ils sont affectés. Citons l'exemple d'un acteur qui a accepté la réalisation d'une tâche et qui

s'aperçoit qu'il n'aura pas le temps de la réaliser. Il peut alors contacter un de ses amis en lui demandant de la réaliser pour lui.

Définition

La **relation de duplication simple** décrit une situation dans laquelle un acteur A affecté à la tâche T_i désire sous-traiter sa réalisation à un acteur B. A est le sous-traitteur et B est le sous-traitant. Cela se fait en créant une tâche sous-traitante T_j dont le concepteur est A et qui sera affectée à B. On crée également une tâche de contrôle T_c dont le concepteur est A et qui sera affectée à A. Cette dernière permet au sous-traitteur de contrôler les résultats de T_j .

Les règles propres à la sous-traitance sont les suivantes :

- le contrat qui lie A avec le concepteur de la tâche T_i est inchangé ;
- les objectifs opérationnels de T_i et de T_j sont identiques ;
- T_c est séquentiellement dépendante de T_j ;
- la tâche T_i est réalisée par les tâches T_j et T_c ;

Mécanismes de génération

La relation de duplication simple est le fruit de l'aboutissement favorable d'un processus de délégation qui a été initialisé par l'acteur affecté à la réalisation de la tâche objet de la délégation.

Si le processus de délégation est un appel d'offre et que le délégant (affecté à la réalisation de la tâche objet du processus) sélectionne plusieurs délégataires, on créera plusieurs tâches sous-traitantes qui seront reliées entre elles par une relation de duplication multiple [PETIT 93]. Cette dernière est identique à la relation de duplication simple, excepté la présence de plusieurs tâches sous-traitantes.

3.3.2.2. Le modèle des acteurs

3.3.2.2.1. Contenu et objectif

Rappelons que, dans le cas des situations ad hoc, la délégation est dynamique, c'est-à-dire qu'il n'existe aucune règle organisationnelle de répartition du travail. Donc, cette activité comprend :

- une phase de recherche au cours de laquelle le délégant potentiel cherche dans l'organisation les acteurs qui seraient susceptibles d'accepter la tâche qu'il désire déléguer ;
- une phase de négociation, interaction entre délégant et délégataire potentiel, dont le but est d'obtenir le consentement de ce dernier.

Le modèle des acteurs supporte la première phase du processus de délégation. Il regroupe l'information relative aux fonctions et aux compétences des acteurs faisant partie de l'organisation.

3.3.2.2.2. Les concepts du modèle

Les différents concepts et relations qui définissent la sémantique du modèle de l'organisation sont représentés par le schéma E/A de la figure 3.5.

'Un **acteur** est un individu ou un ensemble d'individus assumant une ou plusieurs fonctions dans l'organisation.'[PETIT 94]. La relation de subordination est définie du domaine entité acteur vers lui-même et elle exprime, la position hiérarchique de chacun des acteurs.

Une **fonction** est une aptitude à la réalisation d'un certain type de tâches. Par exemple, une personne occupant la fonction de secrétaire est apte à la réalisation de tâches telles que la prise de rendez-vous, la réservation d'avion, la dactylographie de rapports, etc. Dans certain cas, les fonctions sont explicitement définies par l'objet type de tâches qui permet, par exemple, à un acteur voulant déléguer une tâche d'un certain type, de connaître la fonction que devra occuper la personne à laquelle il va s'adresser.

Un **acteur individuel** est un acteur qui peut être spécialisé dans un ou plusieurs domaines et que l'on peut contacter. Un domaine de compétences est une spécialisation de l'acteur individuel. Par opposition aux compétences fonctionnelles qui sont spécifiques aux fonctions assumées par l'acteur dans l'organisation, ces compétences particulières sont propres à l'individu et résultent de son histoire personnelle.

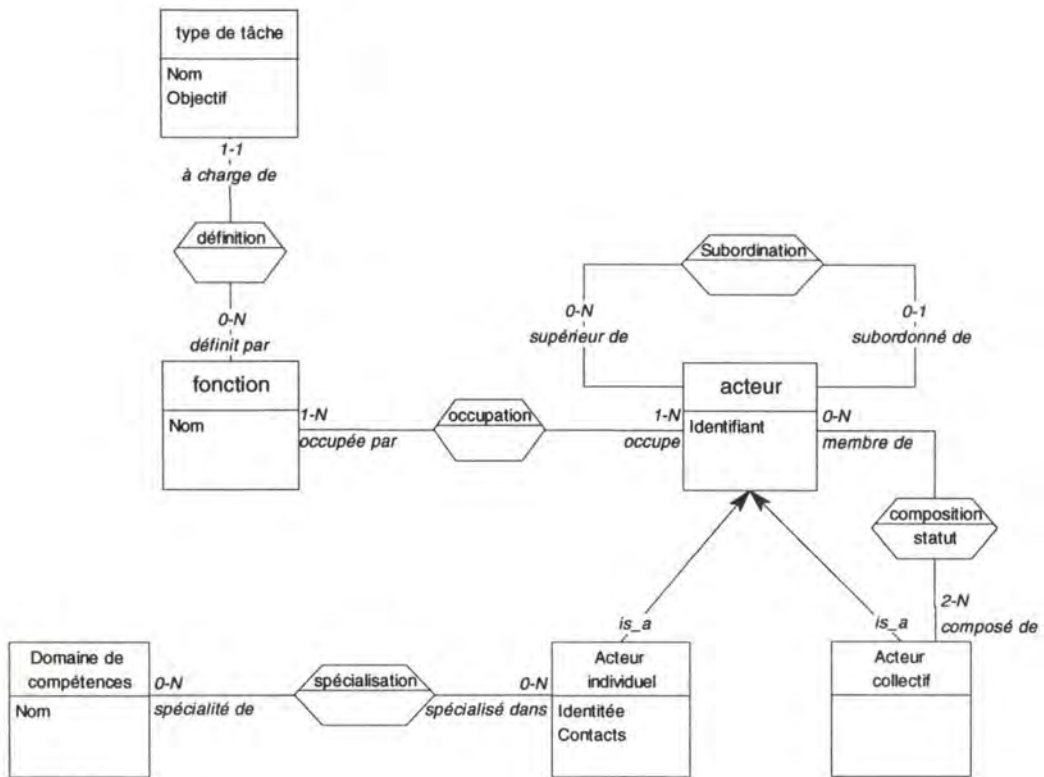


Fig. 3.5. Représentation E/A du modèle des acteurs³ [PETIT 94]

3.3.2.3. Le modèle des conversations

3.3.2.3.1. Objectif

L'objectif du modèle des conversations est d'offrir un support aux échanges d'informations entre acteurs. Ces échanges d'informations ont lieu dans le cadre des activités de délégation et de gestion de contrats. Le modèle offre des protocoles de conversation dirigent la conversation dans le but que celle-ci se déroule dans les meilleures conditions.

3.3.2.3.2. Les concepts du modèle

Le modèle se base sur les concepts définis par la théorie langage/action de [WINO 87]. Ces concepts sont le message, les conversations et les protocoles de conversation.

³ Ne sont représentées que les contraintes d'intégrité relatives aux connectivités et aux relations de sous-typage.

'Un **message** est l'unité élémentaire de transmission d'informations entre deux acteurs.'[PETIT 94].

'Une **conversation** est une suite de messages échangés entre deux acteurs interagissant dans un but précis'[PETIT 94].

Un **protocole de conversation** est un schéma états/transitions qui représente le déroulement de la conversation. Chaque état est caractérisé par des transitions qui représentent les évolutions possibles de la conversation. Ces évolutions sont des messages que peuvent s'envoyer les acteurs. (Fig. 3.6.).

Le schéma états/transitions que voici représente une conversation qui permet à un acteur **a** de demander à un acteur **b** de faire quelque chose. Elle est initialisée par une demande de l'acteur a (request). L'acteur b a alors le choix d'accepter (promise), de refuser (decline), etc. Pour chacun des états, les messages que peuvent échanger les acteurs sont représentés par des arcs.

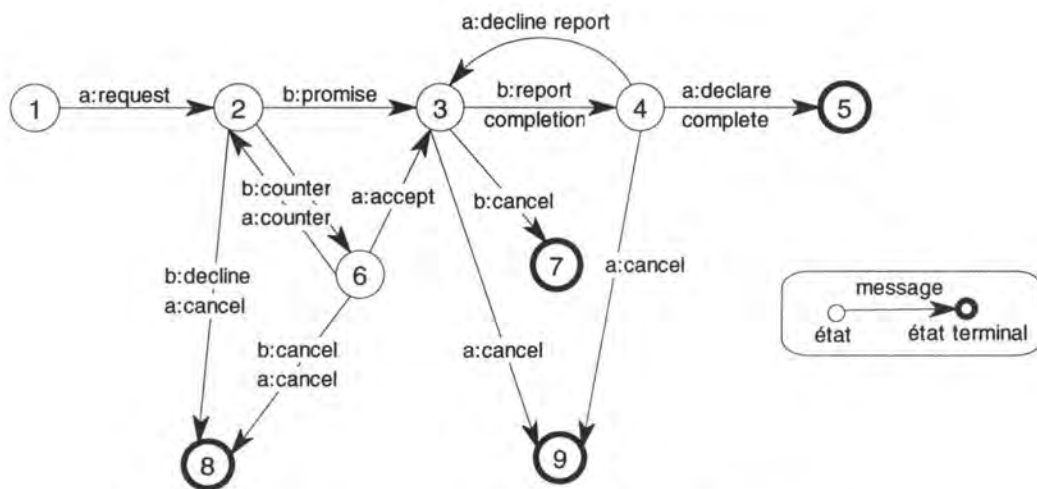


Fig.3.6. Conversation de sollicitation [WINO 88]

Le protocole de conversation représenté par la figure 2 est très général. Pour prendre en compte les situations propres aux activités de délégation et de gestion d'ACTA, ses concepteurs ont défini des protocoles spécifiques [PETIT 94].

Chapitre 4 :

Comparaison d'ACTA et de WoorKS

4.1. Introduction

Au cours de ce chapitre, nous allons comparer ACTA et WoorKS afin d'identifier leurs différences. Nous verrons que ces dernières proviennent de la nature même des deux systèmes, c'est-à-dire des différences qui caractérisent leurs domaines d'application.

Nous nous baserons sur les présentations faites aux chapitres 2 et 3. Ces présentations regroupent des préoccupations différentes : la présentation de WoorKS se situe à la fois au niveau conceptuel et niveau de l'implémentation tandis que la présentation d'ACTA se situe uniquement au niveau conceptuel. Cette différence entre les deux chapitres est une conséquence de la différence de degré de développement de l'un et de l'autre de ces deux systèmes. WoorKS est implémenté et utilisé dans certaines entreprises [DI GI 92] alors que les choix d'implémentation d'ACTA n'ont pas encore été effectués¹.

En vue d'évaluer les possibilités de réutilisation de l'environnement de WoorKS pour implémenter ACTA, nous allons commencer par une comparaison conceptuelle. Elle comprendra une comparaison des processus de développement, une comparaison des

¹ C'est une des raisons de ce travail, à savoir : offrir un environnement de développement facilement réutilisable pour implémenter ACTA.

modèles de représentation des tâches et une comparaison des modèles de représentation de l'organisation².

Les constats exposés dans cette partie seront à la base des choix portant sur l'intégration des concepts d'ACTA et de WoorKs, en vue de proposer un environnement intégré comme support aux situations de coopération répartie prédéfinie et ad hoc (infra : **Chapitre 5 : Intégration d'ACTA et de WoorKs**).

4.2. Les domaines d'application

L'ensemble des différences qui seront présentées dans la suite de cette partie sont le fruit des différences qui séparent le domaine d'application d'ACTA de celui de WoorKs et ce, conformément à la règle selon laquelle la spécification d'un système informatique résulte de son domaine d'application.

ACTA a été conçu en vue de supporter des situations de coopération répartie et ad hoc (supra **1.4. Les systèmes de gestion de la coopération**). Ainsi, sa spécification dépend de la définition de ce type de coopération.

WoorKs est un support aux situations de coopération répartie et prédéfinie (supra **1.4. Les systèmes de gestion de la coopération**). Donc, sa spécification dépend de la définition de ce type de coopération.

4.3. Les processus de développement

4.3.1. Préambule

Nous ne pouvons aborder clairement la comparaison d'ACTA et de WoorKs sans préciser les termes qui seront utilisés.

La **spécification** d'une situation de coopération consiste à la définir en utilisant un formalisme compréhensible par l'ensemble des acteurs chargés de la définition.

La **mise en oeuvre** d'une situation de coopération consiste à traduire sa spécification en objets et relations compréhensibles pour le système informatique sensé la supporter.

² Nous n'aborderons pas la comparaison des concepts propres aux modèles des communication.

Remarquons que le lien entre la spécification et la mise en oeuvre est le formalisme de définition des situations de coopération. Ce formalisme devra être compréhensible pour l'ensemble des personnes chargées de la spécification et de la mise en oeuvre de la situation de coopération; mais il devra aussi être complet et modélisable.

Un formalisme de définition est dit **complet** lorsqu'il permet de représenter l'ensemble des situations de coopération appartenant au domaine d'application du système informatique visé par la mise en oeuvre.

Un formalisme de définition est dit **modélisable** lorsqu'il est possible de mettre en oeuvre toutes les situations de coopération qu'il permet de décrire.

Le **processus de développement** se compose de la spécification et de la mise en oeuvre de la situation de coopération.

4.3.2. Le processus de développement propre à WoorKs³

WoorKs supporte les situations de coopération répartie prédéfinie. Ce type de situations étant caractérisé par la stabilité et la répétitivité, la connaissance organisationnelle relative à une situation particulière est importante.

Dès lors, la phase de spécification du processus de développement fournit une définition complète et détaillée de la situation de coopération. Cette définition comprend une description exacte des documents utilisés, de leur flux et des traitements qu'ils subiront tout au long de l'exécution de la procédure (supra **2.1.6. Le modèle des procédures**).

Cette définition est rédigée dans un formalisme compréhensible par toutes les personnes impliquées dans son élaboration et par les personnes qui seront chargées de sa mise en oeuvre.

La phase de mise en oeuvre ne débute que lorsque la phase de spécification est totalement terminée. Elle est à charge des développeurs d'application et consiste à transcrire la définition de la situation de coopération en un ensemble d'objets et de relations qui seront compréhensibles par WoorKs (supra **2.3. Le processus de développement**).

³ supra : **2.3. le processus de développement**

Une fois que le processus de développement est terminé, l'exécution de la procédure peut commencer (figure 4.1.).

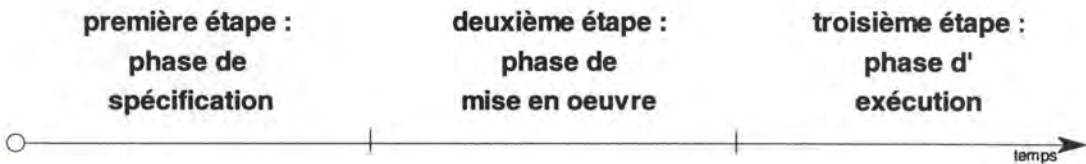


Fig. 4.1. De l'identification à l'exécution d'un workflow

Les trois étapes que sont la spécification, la mise en oeuvre et l'exécution sont distinctes et généralement à charge de personnes différentes. Ainsi, la phase de spécification sera à charge de gestionnaires spécialistes de la formalisation d'une procédure organisationnelle, et la phase de mise en oeuvre sera à charge de spécialistes des systèmes d'information.

Ce 'cloisonnement' des compétences est dû à la complexité des étapes de spécification et de mise en oeuvre. Cette complexité engendre une représentation complète et détaillée des procédures. En se basant sur cette représentation le système offre un grand nombre de fonctionnalités aux acteurs prenant part à la réalisation des procédures. Prenons comme exemple la représentation de la réalisation d'une activité (supra **2.1.4. Le modèle des opérateurs**).

Ex : Une activité dont le but est de créer une fiche de paie avec le traitement de texte Word. La spécification de sa réalisation comprendra la référence à ce traitement de texte plus un ensemble de macro-commandes qui guideront l'acteur tout au long de son travail de réalisation. Le travail de l'acteur réalisateur est simplifié au maximum : il n'a plus qu'à remplir les champs vides.

4.3.3. Le processus de développement propre à ACTA⁴

4.3.3.1. Introduction

⁴ supra 2.3. le processus de développement

ACTA vise les situations de coopération répartie et ad hoc. Ce type de situations étant caractérisé par la nouveauté, l'expérience organisationnelle relative à une situation particulière est faible.

De plus, l'accélération des modifications des marchés (supra **Chapitre 1. La problématique des S.I.A.C.**) exige une grande flexibilité de la part des entreprises désireuses d'assurer leur survie. Dès lors, les situations de coopération ad hoc se multiplient et doivent se mettre rapidement en place.

Compte tenu des caractéristiques précitées, ACTA devra⁵ intégrer un processus de développement simple et rapide pour supporter ce type de situation.

4.3.3.2. Simplicité du processus de développement

Supposons qu'une entreprise se trouve confrontée à l'introduction d'une innovation technologique dans son secteur d'activité. Nous prendrons l'exemple de l'introduction des machines à écrire électroniques sur le marché des machines à écrire : nous sommes un fabricant de machine à écrire mécaniques. Par conséquent, cette innovation technologique équivaut pour nous à l'établissement d'une nouvelle situation de coopération répartie ad hoc dont le but sera d'analyser et de réagir au déséquilibre de notre secteur d'activité.

Nous ne disposons d'aucune expérience organisationnelle concernant la tâche d'analyse et de réaction à l'introduction de machines à écrire électroniques. Donc, la phase de spécification ne saurait donner une description exacte du flux des documents, des traitements qui seront réalisés, etc. A cette absence d'expérience organisationnelle s'ajoute une autre difficulté : les tâches faisant partie de ce type de coordination sont souvent des tâches de gestion, ce qui rend presque impossible toute description de leur réalisation.

Du fait de ces deux caractéristiques, la phase de spécification ne peut fournir a priori une description fiable de la situation de coopération (supra **3.2.2.1. Les activités de construction de la situation de coopération**). Il est donc inutile de vouloir représenter a priori le flux des documents, les traitements qui leur seront appliqués, etc.

Dès lors, durant la phase de spécification du processus de développement d'ACTA on se contentera d'identifier l'objectif de la tâche cible, de la décomposer en sous-objectifs de

⁵ Rappelons que dans son état actuel, ACTA n'est pas un système mais un ensemble de concepts qui devront être appliqués en vue de créer un support pour les situations de coopération répartie et ad hoc.

complexité moindre et d'attribuer la réalisation des sous-tâches, ainsi créées, aux acteurs de l'organisation.

Si l'on compare les objectifs de l'étape de spécification d'ACTA avec les objectifs de l'étape de spécification de WoorKs, on constate que l'information produite par la première est moins précise que celle produite par la seconde (Tab. 4.2.).

	<u>spécification de la réalisation</u>	<u>spécification des entrées/sorties</u>
WoorKs	identification des outils et de la succession des fonctions qui seront utilisées;	une description de la structure des documents en entrées et en sorties ;
ACTA	une description en langage naturel de son objectif ;	une description en langage naturel des données et des résultats ;

Tab.4.2. Différenciation des objectifs de l'étape de spécification de la situation de coopération

Si l'information représentée et prise en compte, par le support de la situation de coopération, est moins complexe, la phase de mise en oeuvre sera moins importante, d'où la simplification du processus de développement.

Notons qu'ACTA n'impose pas la marche à suivre pour réaliser les tâches. Les acteurs ont des obligations de résultats et ils sont libres de choisir la manière d'y arriver. Cette liberté de réalisation leur procure plus de responsabilités. Ces responsabilités engendrent une amélioration de leur satisfaction personnelle⁶.

4.3.3.3. Flexibilité d'ACTA

Comme nous l'avons signalé précédemment, la flexibilité est une nécessité pour les systèmes prétendant supporter les situations de coopération ad hoc. Cette flexibilité est conditionnée par la rapidité de mise en place de la nouvelle situation de coopération: en effet , si la spécification de la situation de coopération destinée à définir la réaction de la

⁶ Il serait très intéressant d'analyser les conséquences psychologiques liées à l'utilisation de tel ou tel système. Malheureusement, nous ne disposons d'aucune étude de ce type à propos d'ACTA et de WoorKs.

firme face à l'introduction de machines électroniques dans son secteur d'activité dure un mois (ce qui est le cas du processus de développement de WoorKs [SRIVA 92]), il est probable qu'elle sera devancée par les concurrents ne disposant pas de support pour les situations de coopération. De ce fait, ces derniers auront gagné un mois dans la redistribution des cartes du secteur d'activité.

Pour assurer la rapidité de mise en place de nouvelles situations de coopération, ACTA devra disposer d'un processus de développement rapide. Dès lors, la construction des plans de réalisation d'ACTA se fera dynamiquement, c'est-à-dire que

- l'ensemble des acteurs aura la possibilité d'y participer ;
- les phases de spécification, de mise en oeuvre et d'exécution ne seront plus différenciées.

4.3.3.3.1. Participation de l'ensemble des acteurs aux processus de développement

La simplification du processus de développement engendre un décloisonnement des compétences, c'est-à-dire que **tous les acteurs prenant part à la situation de coopération sont capables de participer à sa spécification**. Pour ce faire, on remplace l'utilisation du formalisme de définition, qui était le transporteur informationnel liant la spécification et la mise en oeuvre, par un outil de spécification (infra : **5.2. L'outil de spécification et de gestion**). Cet outil permet à chacun des acteurs d'exprimer la partie de la situation de coopération dont il est responsable, en utilisant directement les objets et relations appartenant à ACTA.

Dès lors, il n'y a plus de différenciation entre la personne qui identifie une nouvelle procédure et la personne chargée de sa spécification. Cela engendre une diminution du risque d'incompréhension entre les acteurs lié au passage de la phase de spécification à la phase de mise en oeuvre.

4.3.3.3.2. Superposition du processus de développement et de l'exécution

La **superposition** du processus de développement et de l'exécution permet de commencer l'exécution du plan de réalisation (supra **Chapitre 3 : ACTA**) propre à une situation particulière avant que ce dernier ne soit complètement spécifié.

Cela augmente la vitesse de réaction de l'entreprise face, par exemple, à l'introduction des machines à écrire électroniques.

4.4. Représentations de l'activité

4.4.1. Introduction

Les choix faits au niveau de la représentation de l'activité, à savoir la spécification des éléments de la réalité qui doivent être représentés, semblent être l'élément central d'un système de gestion de la coopération. En effet, ces systèmes s'appuient sur une représentation de l'activité (plus ou moins précise) pour offrir aux acteurs un ensemble plus ou moins grand de fonctionnalités.

Il s'agit en fait de trouver le meilleur compromis entre la complexité du processus de développement et l'ensemble des fonctionnalités que l'on peut offrir en se basant sur la représentation de l'activité qu'il génère.

Au cours de cette partie, nous essayerons de mettre en évidence les principales différences qui existent entre les objets centraux de cette représentation, à savoir les activités (WoorKs) et les tâches (ACTA).

4.4.2. Les cycles de vie

Nous commencerons par rappeler brièvement les cycles de vie associés aux activités, aux procédures (Fig. 3.) (supra 2.1.6.3. **Activité**) et aux tâches (Fig. 2.)(supra 3.3.2.1.2. **Les concepts du modèles**), pour ensuite aborder leurs différences.

4.4.2.1. Rappel

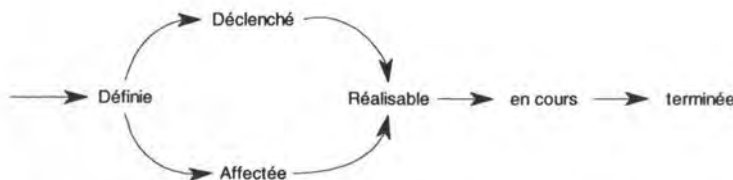


Fig. 4.3. Représentation simplifiée du cycle de vie d'une tâche

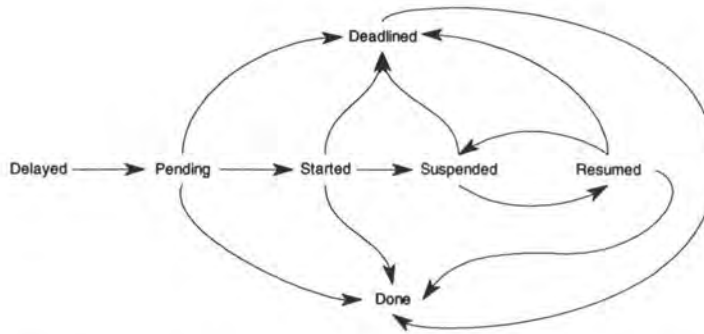


Fig. 4.4. Représentation simplifiée du cycle de vie d'une activité ou d'une procédure

4.4.2.2. Comparaison

Les cycles de vie se composent d'états et d'événements remarquables, ces derniers provoquant le passage d'un état à un autre. Les états et les événements représentés par WoorKs sont exclusivement relatifs à l'exécution des activités ou des procédures. Les états et les événements représentés par ACTA sont relatifs à la fois à la spécification et à l'exécution des tâches.

Cette différence est une conséquence de la **non-représentation du processus de développement** et de la **séparation du développement et de l'exécution du workflow**.

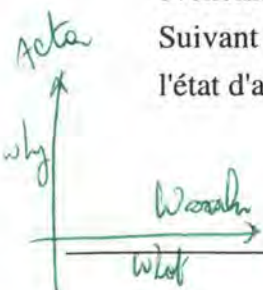
WoorKs ne représente pas le processus de développement. Le développeur doit connaître la méthode à suivre s'il veut définir une procédure.

ACTA offre un outil d'aide à la réalisation qui guide les acteurs lors de la spécification de leur travail.

WoorKs sépare nettement les moments de spécification et d'exécution d'une procédure. On n'exécute celle-ci que lorsqu'elle est entièrement spécifiée.

ACTA permet de commencer la réalisation d'une tâche alors que le plan de réalisation auquel elle appartient n'est pas totalement spécifié. De ce fait, il peut y avoir des interférences entre la suite de la spécification du plan de réalisation et l'exécution des tâches qui le compose.

Prenons comme exemple la destruction d'une tâche composante : c'est un événement remarquable qui correspond à une modification du plan de réalisation. Suivant le contexte dans lequel elle survient, cette destruction aura des effets sur l'état d'avancement de la tâche composée. [PETIT 94].



Par la représentation d'états relatifs au processus de développement, le cycle de vie des tâches d'ACTA englobe le cycle de vie des activités de WoorKs. Néanmoins, les deux systèmes représentent des situations de coopération répartie. Cette similitude permet d'établir une correspondance entre les états remarquables du cycle de vie d'ACTA et du cycle de vie de WoorKs (Tab. 4.5.).

<u>ACTA</u>	<u>WoorKs</u>
définie	
déclenchée	
affectée	delayed
réalisable	pending
en cours	started, suspended, resumed, deadlined
terminée	done

Tab. 4.5. Correspondance entre les états des tâches et des activités

Notons une description plus détaillée de l'état d'avancement de la réalisation des activités de WoorKs. L'état *en cours* pour ACTA correspond aux états *started*, *suspended*, *resumed* et *deadlined* pour WoorKs.

Dans le chapitre suivant, consacré à l'intégration d'ACTA et de WoorKs, nous présenterons et résoudrons les problèmes liés à la présence d'objets dont l'évolution est caractérisée par des cycles de vie différents.

4.5 Représentations de l'organisation

4.5.1. ACTA

Le modèle de l'organisation d'ACTA a été développé comme support de l'activité de délégation[PETIT 94]. Il ne contient que les informations nécessaires à la répartition du travail au sein de l'organisation. Ce modèle ne tient pas compte de l'environnement organisationnel et n'offre qu'une faible représentation de la structure organisationnelle (supra 3.3.2.2. **Le modèle des acteurs**).

Notons la différenciation entre les compétences fonctionnelles, spécifiques aux fonctions assumées par l'acteur au sein de l'organisation, et les compétences personnelles qui sont propres à l'individu et reconnues par l'organisation [PETIT 94]. On peut, par exemple, représenter le fait qu'un programmeur Windows est spécialisé dans le domaine des interfaces graphiques. Sa fonction est 'programmeur Windows' et ses compétences personnelles sont 'création d'interface graphique'.

4.5.2. WoorKs

Le modèle organisationnel de WoorKs est plus qu'un simple support de la répartition des activités au sein de l'organisation.

Il contient des informations relatives à l'environnement organisationnel(supra **2.1.1.4. Représentation de l'environnement**) qui peuvent être utilisées pour représenter les clients, les fournisseurs ou toute autre personne avec qui l'organisation a des contacts.

La représentation de la structure organisationnelle(supra **2.1.1.2. La structure organisationnelle**) est plus complète et offre plus de services que la représentation offerte par ACTA. Par exemple, il existe des méthodes qui permettent de trouver la secrétaire du groupe auquel appartient un acteur.

La représentation des compétences organisationnelles(supra **2.1.1.3. Les compétences organisationnelles**) se limite à la tenue, par un acteur, d'un ou de plusieurs rôles au sein d'un ou plusieurs groupes. Contrairement au modèle d'ACTA, les rôles ne sont pas spécifiés par un ensemble de types de tâches qu'ils sont censés réaliser. Cette définition est implicitement connue par la personne qui définit les procédures.

4.6. Répartition du travail au sein de l'organisation

Les systèmes de gestion de la coopération répartie offrent des fonctionnalités de répartition du travail parmi les membres de l'organisation.

4.6.1. WoorKs

Rappel : le modèle des procédures de WoorKs ne représente pas le processus de développement et n'en tient pas compte..

Il n'existe aucune relation entre le modèle de représentation des procédures et le modèle de l'organisation qui soit relative à la conception du workflow. Les seules relations que l'on trouve sont relatives à l'exécution. Parmi ces relations, on distingue les relations d'initialisation, de responsabilité et de réalisation (supra Fig. 2.20.)

Lors de la définition du workflow, le développeur d'application utilise la notion de rôle pour 'initialiser' les relations présentées ci-dessus. Lorsqu'il définit une activité, il a le choix entre attribuer sa réalisation à un acteur précis ou à un rôle [MOLLE 93]. S'il choisit de l'attribuer à un rôle, la liaison avec un acteur ne se fera que lors de la création de l'activité (supra 2.1.7.4.3. **L'exécution d'un workflow**). Cela permet de combler le décalage qui pourrait exister entre le moment de définition et d'exécution de l'activité.

C'est donc au moment de la définition de la procédure que l'on spécifie l'ensemble des personnes qui pourront être initiatrices, responsables ou réalisatrices d'un des objets de représentation de l'activité.

4.6.2. ACTA

Contrairement à WoorKs, ACTA représente le processus de développement. Les relations entre le modèle de l'organisation et le modèle de représentation de l'activité sont relatives à la conception et à l'exécution du plan de réalisation.

Lors de la définition d'une activité, le concepteur utilise les renseignements offerts par le modèle de l'organisation pour attribuer la réalisation de l'activité à un acteur. Contrairement à WoorKs, tous les employés doivent avoir accès aux informations concernant les compétences de leurs collègues pour initialiser un processus de délégation avec les acteurs censé être compétents pour réaliser la tâche qu'ils ont définie. Il ne sert à rien de vouloir déléguer une tâche de secrétariat au responsable d'un projet.

Le but de la représentation des compétences d'ACTA est différent de la représentation des compétences de WoorKs. Pour ACTA, il s'agit d'offrir aux acteurs concepteurs l'information nécessaire pour qu'ils choisissent l'acteur à qui ils vont déléguer leur tâche. Pour WoorKs, il s'agit de combler la lacune qui existe entre le moment de spécification et celui de la réalisation. En effet, il arrive qu'un acteur présent lors de la spécification ne fasse plus partie de l'organisation lors de l'exécution.

Chapitre 5 :

Intégration d'ACTA et de WoorKs

5.1. Introduction

Dans le premier chapitre, nous avons constaté l'augmentation du nombre de situations de coopération formées d'aspects prédéfinis et ad hoc (supra **1.5. Evolution attendue**). Nous avons analysé et comparé ACTA et WoorKs en vue de les intégrer pour produire un support des situations de coopération répartie et ad hoc.

Il existe une grande différence entre l'état d'avancement des recherches concernant ces deux systèmes: WoorKs est utilisé dans certaine entreprise et ACTA n'est qu'au stade de développement conceptuel. Si l'on veut évaluer les possibilités d'implémentation d'ACTA en réutilisant les objets et relations appartenant aux modèles de WoorKs, il faut amener ACTA au niveau de l'implémentation (infra **5.2. Choix d'implémentation pour ACTA**).

Une fois ACTA amené, rapidement il est vrai, au niveau d'implémentation, nous aborderons la problématique propre à l'élaboration d'un système intégré basé sur les concepts d'ACTA et de WoorKs (infra **5.3. Interfaçage d'ACTA et de WoorKs**).

Suivra une partie consacrée a l'implémentation du système intégré ACTA-WoorKs. On y verra comment implémenter les concepts d'ACTA en utilisant l'environnement de WoorKs (infra **5.4. Implémentation d'ACTA-WoorKs**).

5.2. Choix d'implémentation pour ACTA

5.2.1. Introduction

Comme l'environnement de WoorKs est implémenté, il serait dommage de ne pas en profiter pour implémenter ACTA. C'est pourquoi nous avons fait les choix d'implémentation d'ACTA en essayant de réutiliser l'environnement de WoorKs.

ACTA comprend le modèle d'organisation des tâches, le modèle des acteurs et le modèle des conversations. Les choix d'implémentation déterminent un ensemble d'outils¹ qui permettront d'utiliser la connaissance de chacun de ces modèles.

Parmi ces outils, certains sont propres à l'un ou l'autre de ces modèles mais, dans la majorité des cas, ils ont des relations avec la connaissance relative à plusieurs d'entre eux.

Nous commencerons en présentant la corbeille de l'utilisateur qui est au centre de l'interaction de ce dernier avec le système(**infra 5.2.3. La corbeille**). Ensuite, nous aborderons l'outil de spécification et de gestion des situations de coopération répartie et ad hoc(**infra 5.2.4. L'outil de spécification et de gestion**). Suivra celui de réalisation des tâches (**infra 5.2.5. L'outil de réalisation**) qui permet aux acteurs d'atteindre l'objectif opérationnel des tâches qui leurs ont été affectées. Nous terminerons par l'outil de gestion du modèle de l'organisation (**infra 5.2.6. Le gestionnaire du modèle de l'organisation**) et par le système de communication (**infra 5.2.7. Le système de communication**).

Avant toute chose, il faut spécifier la représentation des informations qui circuleront au sein d'un plan de réalisation. Si nous voulons présenter les choix d'implémentation concernant ACTA et envisager concrètement son intégration avec WoorKs, on ne peut éluder la problématique propre à la gestion des informations.

5.2.2. Représentation des informations

[PETIT 93] n'aborde que très brièvement la problématique de gestion de l'information au sein d'un arbre de réalisation. On peut y lire :

¹Remarquons l'importance de l'interface homme/machine de ces outils qui, pour la plupart, seront utilisés par une grande partie des acteurs de l'organisation; voir [MEIN 91] pour plus de renseignements.

' la charge de coordination interne, c'est-à-dire des acteurs assumant un rôle de réalisation consiste à : [...] la mise à disposition des ressources, en particulier des informations, nécessaires à la réalisation des tâches subséquentes. [...] '.

Le caractère ad hoc engendre une impossibilité de fixer a priori les entrées et les sorties des tâches. Dès lors, les entrées seront spécifiées dynamiquement par le concepteur et les sorties seront spécifiées dynamiquement par le réalisateur.

- La spécification dynamique des entrées d'une tâche se fait avant sa délégation. Le délégrant choisit les documents qu'il désire passer au délégataire.
- La spécification dynamique des sorties d'une tâche se fait avant sa terminaison. Le délégataire choisit les documents qu'il considère être les résultats de sa tâche (en tenant compte de la définition de l'objectif opérationnel).

Nous utiliserons les concepts de folder in, folder out et folder transient. Le folder in contiendra les entrées, le folder out contiendra les sorties et le folder transient les données intermédiaires de la tâche. (Fig. 5.1.)

Un **folder** est un ensemble de références. Il dispose de méthodes qui permettent de le créer, de le remplir, de le vider, de le copier dans un autre folder et de le détruire.

Une **référence** représente l'endroit où se trouve un document et l'outil qui permet de le manipuler.

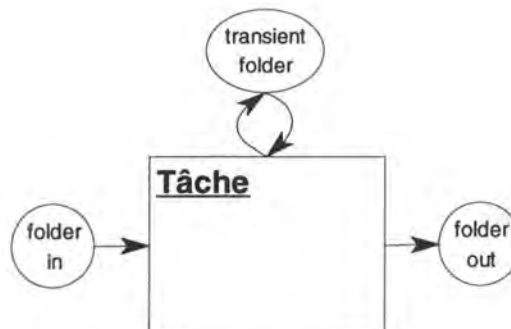


Fig. 5.1. Représentation des informations manipulées par une tâche

Toujours selon [PETIT 93] : 'Cette charge peut être allégée par la définition a priori (notamment lors de la planification) de conventions quant aux caractéristiques de présentation des informations et des procédures de mise à disposition. '

Nous ne nous attarderons pas sur ces mécanismes 'prédéfinis' de circulation et de présentation des documents. Remarquons simplement que :

- la présentation est à charge des outils de manipulation ;
- la circulation se fera conformément aux relations reliant les différentes tâches, par exemple :
 - la transmission des résultats d'une tâche composante vers le folder transient de la tâche décomposée à laquelle elle appartient ;
 - la transmission de la totalité ou d'une partie des résultats d'une tâche origine d'une relation de séquençement vers la tâche cible de cette même relation.

5.2.3. La corbeille

La corbeille est directement inspiré de la corbeille de WoorKs. Ses fonctionnalités diffèrent quelque peu de celles proposées par l'objet dont il a été tiré.

5.2.3.1. Définition

Une **corbeille** appartient à un acteur. Elle lui permet d'interagir avec le système. C'est le premier objet qui est présenté à l'acteur lorsqu'il initialise son poste de travail.

5.2.3.2. Fonctionnalités

La corbeille comprend une référence aux tâches pour lesquelles son propriétaire est délégant ou déléataire. Elle donne accès aux outils de travail parmi lesquels on trouve l'outil de réalisation des tâches (infra **5.2.5. L'outil de réalisation**), l'outil de spécification et de gestion des situations de coopération (infra **5.2.4. L'outil de spécification et de gestion**), l'outil de communication (infra **5.7. Le système de communication**).

L'interface utilisateur sera contextuelle dans le sens où si l'utilisateur sélectionne une tâche, l'ensemble des outils disponibles et des fonctionnalités accessibles seront fonction de la tâche sélectionnée.

Par exemple, s'il sélectionne une tâche et qu'il démarre l'outil de communication, il aura accès à l'ensemble des conversations ayant pour sujet la tâche sélectionnée.

Voici quelques contextes d'appel relatifs aux outils de réalisation et de spécification :

si le propriétaire de la corbeille sélectionne une tâche T :

- si son état est antérieur à *réalisable*,
alors il peut démarrer l'outil de spécification des plans de réalisation qui lui donnera accès aux tâches dont il est soit le concepteur ou le réalisateur et qui appartiennent au plan de réalisation de la tâche sélectionnée ;
- si son état est réalisable et que le propriétaire de la corbeille en est le réalisateur alors il pourra, soit la réaliser en utilisant l'outil de réalisation qui lui permettra de réaliser, soit continuer la spécification du plan de réalisation auquel elle appartient en utilisant l'outil de spécification.

si le propriétaire n'a sélectionné aucune tâche :

- il n'a pas accès à l'outil de réalisation ;
- il peut démarrer l'outil de spécification pour spécifier un nouveau plan de réalisation.

5.2.4. L'outil de spécification et de gestion des situations de coopération

5.2.4.1. Définition

L'outil de spécification et de gestion est appelé à partir de la corbeille utilisateur et supporte d'une part, les activités de coordination externe l'activité de consolidation et d'autre part, l'activité de coordination interne consistant à ce que le délégrant mette à la disposition du délégataire les informations² nécessaires à la réalisation de la tâche à laquelle il est affecté. (supra : **3.2.1. L'objectif**).

² Le délégrant peut mettre des informations à la disposition du délégataire, mais il n'est pas obligé de le faire. S'il ne le fait pas, le délégataire devra rechercher lui-même les informations dont il aura besoin.

5.2.4.2. Fonctionnalités

Les acteurs communiquent avec l'outil de spécification via une interface graphique contextuelle qui met en oeuvre les règles de guidances propres aux concepts d'ACTA.

Ex. : un acteur ne peut décomposer une tâche qui est dans un état postérieur à *en cours*.

Les fonctionnalités auxquelles auront accès les acteurs sont fonction du contexte dans lequel l'outil a été appelé.

- Si une tâche de la corbeille a été sélectionnée, l'outil présente le plan de réalisation auquel la tâche appartient. En fonction du lien qui lie le propriétaire de la corbeille à la tâche objet, et de l'état d'avancement de cette dernière, il offrira tel ou tel sous-ensemble des fonctionnalités.
- Si aucune tâche de la corbeille n'a été sélectionnée, l'outil offre les fonctionnalités qui permettent de créer un nouveau plan de réalisation.

Fonctionnalités de construction d'une situation de coopération

- créer une nouvelle situation de coopération en initialisant un nouvel arbre de réalisation ;
- décomposer une tâche ;
- greffer une tâche ;
- créer une relation de séquencement ;
- initialiser un processus de délégation ;
- détruire une tâche.

Fonctionnalité de gestion d'une situation de coopération

- suivre l'état d'avancement des tâches ;
- modification de l'échéance de réalisation d'une tâche ;

Fonctionnalités liées aux documents

Lorsqu'un acteur décide de déléguer une tâche, il peut³ choisir les objets qui seront les entrées de cette tâche. Il place la référence à ces objets dans le folder in de la tâche, de sorte que les délégataires potentiels pourront consulter les entrées de la tâche avant d'en accepter la réalisation.

5.2.5. L'outil de réalisation des tâches

5.2.5.1. Définition

Lorsque l'utilisateur dispose d'une tâche *réalisable* dans sa corbeille il peut la *prendre en charge* en démarrant l'outil de réalisation. Ce dernier se compose d'un gestionnaire des documents, d'une boîte à outils et d'un système de communication.

5.2.5.2. Fonctionnalités

5.2.5.2.1. Le gestionnaire de documents

Il permet à l'acteur d'utiliser les entrées de sa tâche, de rechercher d'autres informations, et de sélectionner les documents qu'il considère comme étant les résultats de sa tâche.

Utilisation des entrées de la tâche

Le gestionnaire des documents donne accès au folder in de la tâche et permet de 'passer' les documents qui s'y trouvent aux outils de la boîte à outils (**infra 5.2.5.2.2. La boîte à outils**).

Recherche d'information

Pour obtenir un complément d'information en cours de réalisation d'une tâche, l'acteur utilise un browser. Ce dernier lui permet de parcourir les bases de données en utilisant un langage de consultation.

Les informations qui se trouvent dans ces bases de données sont tous les objets persistants de l'organisation. Ces objets forment la 'mémoire informatique' de

³ S'il n'en choisit pas, la tâche n'a pas d'entrée et donc l'acteur qui sera affecté à sa réalisation devra rechercher lui-même les documents qui lui serviront pour atteindre l'objectif de la tâche.

l'organisation. Cette mémoire comprend des informations dont l'accès doit être réglementé. Dès lors, on définit un mécanisme de protection basé sur le *domaine accessible* des utilisateurs.

Le **domaine accessible** de l'utilisateur est l'ensemble des données informatiques auquel il aura accès. Il est déterminé par la fonction de l'utilisateur au sein de l'organisation et par la tâche que ce dernier est en train de réaliser. Notons la possibilité d'élargir le domaine accessible sur demande adressée à son supérieur.

Détermination des résultats

La fonction de détermination des résultats de la tâche permet au réalisateur de choisir les documents du transient folder qui seront placés dans le folder out. Cette fonction correspond à la prise en charge d'une partie de la coordination interne obligeant les acteurs à veiller à la mise à disposition des informations qu'ils produisent.

5.2.5.2.2. La boîte à outils

La boîte à outil comprend les outils utilisés par les acteurs pour réaliser les tâches auxquelles ils sont affectés. Citons un traitement de texte, un tableur, un langage de programmation et son compilateur, etc.

Pour tenir compte des préférences des acteurs en matière d'utilisation d'outils informatiques(M. Dupont préfère utiliser le traitement de texte WordPerfect alors que M. Grégoire préfère utilise le Word), il faut une boîte à outils 'à géométrie variable' et des logiciels de conversions (qui permettrons à monsieur Grégoire de récupérer les documents de monsieur Dupont).

5.2.5.2.3. Le système de communication⁴

Le système de communication sera accessible à partir de l'outil de réalisation et disposera de protocoles de conversation qui serviront à obtenir de l'information concernant la tâche à réaliser, à obtenir des documents, à discuter librement avec un autre acteur, à élargir son domaine d'accessibilité en s'adressant à son supérieur, etc.

5.2.6. Le gestionnaire du modèle de l'organisation

⁴ infra 5.7.7. Le système de communication

La composition d'une organisation varie tout au long de son existence. On peut engager de nouveaux acteurs, un acteur peut changer de poste, la structure de l'organisation peut changer, etc. Il faut désigner un ensemble d'acteurs qui auront accès à un outil de manipulation du modèle de l'organisation. A l'aide de ce dernier, ils veilleront à ce que les informations contenues dans le modèle de l'organisation soient une image fidèle de la réalité.

Le gestionnaire du modèle de l'organisation leur permettra de visualiser, de modifier ou de détruire des objets existants. Par exemple, si monsieur Dupont est remercié, il faut enlever du modèle de l'organisation l'objet qui le représente.

Le gestionnaire leur permettra également de créer de nouveaux objets. Par exemple, si on crée la fonction d'amuseur public, il faut créer l'objet qui la représente au sein du modèle de l'organisation.

5.2.7. Le système de communication [BIGNO 90][AGOS 91]

5.2.7.1. Introduction

ACTA est un support aux situations de coopération répartie et adhoc. Dès lors, il devra supporter les interactions entre les acteurs faisant partie de la situation de coopération. Ces interactions sont modélisées par les mécanismes de délégation de tâche et de demande de report d'échéance. Chacun de ces mécanismes utilise un protocole de conversation [PETIT 94]. En plus de ces contacts formalisés, il faut laisser aux acteurs la possibilité de communiquer librement (cf. un courrier électronique).

Le concept de 'protocole de communication' à été répandu par [WINO 87] dans le cadre de la théorie langage/action. Cette dernière a donné naissance à plusieurs supports de situations de coopération répartie parmi lesquels on trouve notamment COORDINATOR, AMIGO, CONTRACT, UTUCS, etc [BIGNO 90].

Il nous fallait choisir de, soit développer notre propre gestionnaire des conversations, soit réutiliser un système existant en l'adaptant aux concepts propres à ACTA. Dans le cadre de ce travail, nous avons choisi d'évaluer les opportunités de réutilisation d'UTUCS.

Ce choix a été motivé par l'ensemble des fonctionnalités offertes par UTUCS [AGOS 91] et par le souhait de ses concepteurs de l'intégrer à WoorKs. En 1990, nous pouvions lire :

' The UTUCS component has been conceived as an independent application: it aims to serve its users supporting them while they communicate each other during their

activities. It is our firm belief, anyhow, that its integration within the Activity Coordination System of ITHACA offers interesting possibilities of improving its effectiveness within Office environments '[BIGNO 90].

Néanmoins, notre quête est différente, dans le sens où nous voulons intercaler ACTA entre WoorKs et UTUCS. Nous pensons enrichir la paire WoorKs - UTUCS en y ajoutant ACTA. Cet aspect sera abordé dans la partie consacrée à l'interfaçage d'ACTA et de UTUCS (infra **5.3.2.2.4. Greffage d'une activité**). Avant d'aborder cette partie, il nous faut étudier l'interfaçage entre ACTA et UTUC.

5.2.7.2. Interfaçage entre ACTA et UTUCS

5.2.7.2.1. Comparaison d'ACTA et de UTUCS

UTUCS se situe dans le courant théorique langage/action [WINO 87] qui

' looks at the communication as the basis upon which individuals in the office seek to establish mutual commitments and so, to coordinate their activities ' [BIGNO 90]

Il se base sur les quatre types de protocole suivants : *commitment to do*, *commitment to be*, *information handling*, *information providing*.

ACTA offre plus. Son modèle est réellement une représentation des tâches et des relations qui caractérisent une situation ad hoc. Il est plus riche car il ne se contente pas de quatre protocoles génériques de communications, mais ajoute une sémantique propre aux tâches (supra **3.3.2.1. Le modèle d'organisation des tâches**). En utilisant le concept de conversation couplé au modèle de représentation de l'activité, il représente plus finement la réalité organisationnelle. En se basant sur cette représentation plus fine de la réalité, il offre des fonctionnalités de gestion de la collaboration qui dépassent ce que le courant langage/action peut offrir (Fig.5.2.).



Fig. 5.2. Comparaison des couvertures sémantiques d'ACTA et de UTUCS

5.2.7.2.2. Réutilisation de UTUCS

Le modèle des conversations d'ACTA et UTUCS sont basés sur le concept de conversation présenté dans [WINO 87]. Les différences entre ACTA et UTUCS sont les suivantes :

- ACTA n'utilise pas les mêmes protocoles de conversation;
- ACTA crée un lien étroit entre les conversations et le module de représentation de l'activité. Ex : pour demander un report d'échéance de réalisation d'une tâche à laquelle il est affecté, l'acteur initialise un protocole de négociation de report d'échéance. Si le report d'échéance est accepté, l'échéance de réalisation de la tâche objet de la demande devra automatiquement être modifiée.

Nous n'avons pas eu accès aux spécifications d'UTUCS. Nous sommes donc incapables de dire s'il est possible d'implémenter les types de conversations d'ACTA et de les relier avec le modèle de représentation de l'activité d'ACTA.

Néanmoins, eu égard aux fonctionnalités offertes par UTUCS, il semblerait être un outil de manipulation approprié au modèle des conversations d'ACTA. Voici les principales fonctionnalités d'UTUCS [BIGNO 90] :

- " • opening, closing, visualizing a conversation and defining its type;
- choosing the next move within a conversation among a limited set of alternatives;
- retrieving a conversation according to some requirement;
- classifying the current conversation according to some individual criterion;
- soliciting the user to give the required answers and to do the required actions;

- handling the deadlines of his own commitments with the other member of the network as well of the commitments that they have assumed with him. "

De plus, UTUCS offre des fonctionnalités propres aux situations de coopération collégiale (supra **1.4.1. La coopération**). Il offre, par exemple, un support aux réunions de groupe [AGOS 91].

5.3. Interfaçage d'ACTA et de WoorKs

L'intégration d'ACTA et de WoorKs comprend deux aspects distincts. Le premier est la définition et l'implémentation des relations qui existent entre ces deux approches de représentation de l'activité. Le second consiste à réutiliser l'environnement de WoorKs pour implémenter ACTA: il s'agit de réutiliser des modèles périphériques⁵ tels que la gestion du temps et des événements et des accessoires tels que la corbeille utilisateur, les agendas, etc.

La première composante sera étudiée dans cette partie, alors que la seconde le sera dans la partie intitulée 'Utilisation de l'environnement WoorKs pour implémenter ACTA '.

5.3.1. Introduction

ACTA et WoorKs représentent des sémantiques différentes des situations de coopération (supra **1.5. Evolution attendue** et **Chapitre 4 : comparaison d'ACTA et de WoorKs**). Si l'on veut offrir un support au processus de coopération, il faut tenir compte de ces deux sémantiques, de ces deux représentations d'aspects différents de la réalité. On ne peut cependant les envisager comme deux sous-systèmes indépendants car les domaines qu'ils représentent sont liés. Les situations réelles de coopération comprennent des éléments prédéfinis et ad hoc qui concourent à la réalisation de l'objectif du groupe(Fig. 5.3.).

⁵ Par rapport à la représentation de l'activité, ce modèle est un accessoire certes indispensable, mais pas central.

Situation de coopération

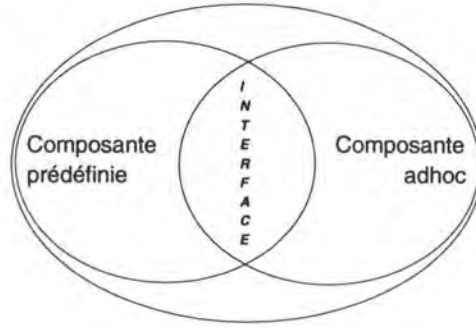


Fig. 5.3. Représentation des composantes d'une situation de coopération

Au cours de cette partie nous étudierons les relations qui lient les composantes prédéfinies et ad hoc. Les relations identifiées seront abordées sous l'angle d'extensions de WoorKs, ces dernières 'ouvrant' le support prédéfini à la représentation et à la prise en compte des situations de coopération ad hoc.

5.3.2. Extension de WoorKs

Nous allons voir comment étendre WoorKs de manière à ce qu'il cohabite avec ACTA pour former un environnement intégré de support aux situations de coopération répartie prédéfinie et ad hoc.

5.3.2.1. Approche générale

Nous allons mettre en évidence les faiblesses de WoorKs face aux situations ad hoc en utilisant un exemple abstrait de procédure⁶.

La procédure est définie par :

- une entrée : le document A dont la structure est définie ;
- une sortie : le document Z dont la structure est définie ;
- ses activités (1,2) et leur ordonnancement et pour chacune d'elles, la définition exacte de leur réalisation.

⁶ La définition de cette procédure a été simplifiée en vue d'en faciliter la compréhension.

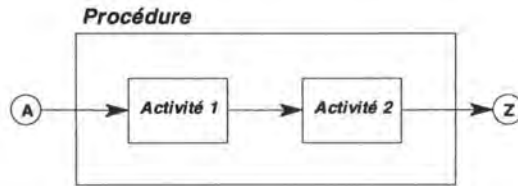


Fig.5.4. Représentation d'une procédure

Cette procédure représente une situation de coopération prédéfinie. Lorsque ses conditions de déclenchement sont vérifiées, elle est créée et elle s'exécute. Elle se place dans la corbeille de son responsable et place l'activité 1 dans la corbeille de l'acteur qui devra la réaliser,

Malheureusement, il pourrait s'avérer que la définition de la procédure faite par la phase d'identification du processus de développement ne permette pas de prendre en compte la situation réelle à laquelle doit faire face son responsable au moment de son exécution. S'il n'existe aucun moyen de l'adapter dynamiquement à cette nouvelle situation, son responsable doit l'abandonner complètement et l'effectuer en dehors du cadre prédéfini par WoorKs.

Ce que nous voulons, c'est permettre au responsable d'adapter la définition de la procédure (qu'il est en train d'exécuter) aux modalités de la situation réelle, non 'prévues' par la phase d'identification.

Pour ce faire, nous allons analyser les extensions suivantes : la *redéfinition d'une procédure*, les *activités libres*, la *redéfinition de la réalisation d'une activité* et le *greffage d'une activité*.

5.3.2.1.1. La redéfinition d'une procédure

Elle consiste à adapter le méta-graphe de la procédure à la situation particulière qu'elle est censée supporter. Le méta-graphe est une représentation de la structure de la procédure (la Fig. 5.4. est un exemple simplifié). L'adaptation consisterait à :

- ajouter ou supprimer une activité ;
- modifier les entrées ou les sorties d'une activité en ajoutant ou en supprimant un document ;
- modifier la spécification de la réalisation d'une activité ;
- modifier l'ensemble des entrées/sorties de la procédure ;
- etc.

Cette redéfinition se ferait, soit juste avant le démarrage de l'exécution de la procédure, soit durant son exécution; elle serait à charge du responsable de la procédure.

Prenons un exemple: à la fin de la réalisation de l'activité 1, le responsable de la procédure estime qu'il serait préférable de modifier la suite de l'exécution de la procédure. Il voudrait dupliquer la deuxième activité pour la faire exécuter par deux acteurs différents. Il crée l'activité 3 à l'image de l'activité 2. Ensuite, il ajoute une quatrième activité dont le but sera de regrouper les résultats des activités 2 et 3 (Fig. 5.5.)

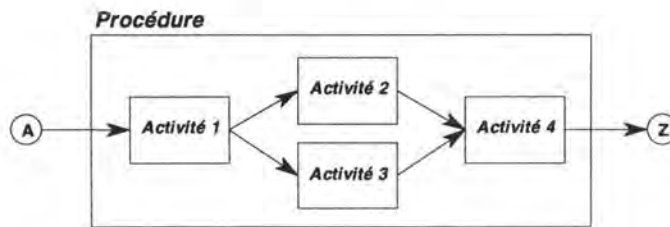


Fig. 5.5. Résultats de l'adaptation du méta-graphe de la fig. 3.

La complexité de la réalisation de cet outil de manipulation du méta-graphe des procédures (infra **5.3.2.2.1. Redéfinition de la procédure**) nous détournera de son élaboration. Nous nous tournerons vers des extensions moins ambitieuses qui porteront essentiellement sur la définition de la réalisation d'une activité. Ces extensions sont *l'activité libre*, *la redéfinition de la réalisation d'une activité* et *le greffage d'une activité*.

5.3.2.1.2. L'activité libre

Elle consiste à laisser des zones d'incertitudes lors de la spécification de la procédure. De cette manière on évite de définir des choses qui n'ont que très peu de chance d'être utilisées au moment de l'exécution. Les zones d'incertitude concernent la spécification de la réalisation des activités. Ce choix est motivé par l'existence de procédures pour lesquelles la décomposition en activités et le flux des documents sont clairement définis mais dont la réalisation de certaines activités ne peut être définie a priori.

Reprenons l'exemple de notre procédure abstraite. Dans la version actuelle de WoorKs, les réalisations des activités 1 et 2 doivent être définies lors de sa spécification. Supposons qu'il soit impossible de spécifier a priori la réalisation de l'activité 1, alors on la définit comme étant indéterminée.

On laisse ainsi la possibilité à l'acteur qui sera affecté à la réalisation de l'activité de faire comme bon lui semble. Néanmoins, il devra remplir les obligations de résultats, de sorte qu'en sortie de cette activité on trouve le(s) document(s) attendu(s).

5.3.2.1.3. Redéfinition de la réalisation d'une activité

La phase de spécification de la procédure détermine, pour chacune de ses activités, la manière dont elles devront être réalisées (supra **2.1.4. Le modèle des opérateurs**). Il peut s'avérer qu'au moment de l'exécution, l'acteur chargé de la réalisation d'une activité se rend compte de l'inadéquation de cette dernière au contexte dans lequel elle s'exécute.

Prenons l'exemple de l'activité de remplissage des formules budgétaires de la procédure d'élaboration du budget. Sa réalisation consiste à présenter le formulaire au doyen de chaque faculté pour qu'il le remplisse champ par champ. Cette réalisation suppose que le doyen ait la connaissance nécessaire pour remplir chaque champ en lui attribuant une valeur. Il pourrait s'avérer qu'un doyen ne dispose pas de cette connaissance, et qu'il désire pouvoir recontacter ses collègues, demander des rapports, etc. Ce qu'il voudrait, c'est initialiser une nouvelle situation de coopération qui aurait comme entrées/sorties les entrées et les sorties de l'activité.

Dès lors, il faudrait permettre aux acteurs chargés de la réalisation d'une activité de la redéfinir complètement en initialisant une situation de coopération ad hoc.

5.3.2.1.4. Greffage d'une activité

Elle est motivée de la même manière que la redéfinition de la réalisation de la procédure, mais son objectif est différent. Plutôt que d'oublier la spécification de la réalisation de la tâche, on la complète en greffant un sous-objectif opérationnel complémentaire.

5.3.2.2. Présentation approfondie

5.3.2.2.1. Redéfinition de la procédure

Chaque type de procédure de WoorKs est représenté par un méta-graphe qui définit sa décomposition en activités et le flux des documents entre chacune d'elles. Ces types de procédures seront instanciés chaque fois que cette procédure devra s'effectuer.

L'idée serait de considérer le méta-graphe de la procédure comme un modèle à partir duquel les acteurs pourraient élaborer la procédure particulière qui correspondrait à leurs

besoins du moment [SCHM 91]. Cette actualisation du méta-graphe serait la solution aux différences de perception de la procédure qui existent entre les moments de spécification et de réalisation.

Il faudrait disposer d'un outil de spécification de la procédure accessible à tous les responsables des procédures; accessible dans le sens où il serait relativement facile à utiliser. Malheureusement, les exigences de spécification de la structure des documents en entrées et en sorties, ainsi que celles de la réalisation des activités rendent impossible la réalisation de ce type d'outil sur base de la représentation de l'activité des modèles de WoorKs.

Le développement de cet outil se situe bien au-delà des prétentions de ce travail. Il demanderait de tels changements dans la structure même de WoorKs que nous nous interrogeons sur sa faisabilité.

5.3.2.2.2. L'activité libre

La définition des procédures de WoorKs se divise en deux parties : la première est la définition de la décomposition de la procédure en activités et la spécification du flux de documents entre ces dernières; la seconde est la définition de la réalisation de chacune des activités.

Il arrive souvent que la seconde partie de la spécification soit incomplète. C'est-à-dire qu'il existe une ou plusieurs activités pour lesquelles on ne peut définir a priori leur réalisation.

Ex. : Le cas d'élaboration d'un budget pour un institut universitaire [PETIT 93] est bien structuré mais il comprend des étapes pour lesquelles on ne peut prédéfinir la façon dont elles seront réalisées. C'est notamment le cas pour la phase d'élaboration des propositions qui sera réalisée par l'administrateur délégué. Ses entrées et ses sorties sont bien spécifiées: il s'agit d'une analyse des demandes budgétaires des doyens (comme entrée) et d'une proposition budgétaire (comme sortie). La manière dont l'administrateur va réaliser la proposition en fonction de l'analyse ne sera définie implicitement que lorsqu'il la réalisera.

Il faudrait, d'une part, permettre au développeur d'application de ne spécifier que les entrées et sorties de l'activité et, d'autre part, accorder la liberté de réalisation à l'acteur affecté à sa réalisation en lui offrant un outil d'aide à la réalisation.

Permettre au développeur de ne pas spécifier la réalisation d'une activité

Le développeur spécifie la structure et la réalisation de la procédure en utilisant le langage A.D.L. (activity description language). Il faudra y ajouter le concept d'activité libre pour lequel il ne devra pas spécifier de réalisation.

Accorder la liberté de réalisation

Il s'agit d'ouvrir le monde prédéfini au monde ad hoc. Les activités libres sont affectées à un acteur qui sera chargé de leur réalisation. Si l'on veut réellement permettre à cet acteur de définir la réalisation des activités libres, il faut lui permettre de les décomposer, d'en déléguer une partie de la réalisation, etc. C'est-à-dire qu'il faut lui permettre de définir une situation de coopération dont l'objectif serait de fournir les documents résultats de l'activité libre.

Cette ouverture se divise en deux parties : la première consiste à offrir aux acteurs un outil de réalisation dynamique(supra **5.2.5. L'outil de réalisation des tâches**) ; la seconde consiste à leur offrir un outil de spécification de situation de coopérations ad hoc(supra **5.2.4. L'outil de spécification et de gestion**). Ainsi, les acteurs ont le choix de réaliser eux-mêmes la totalité de l'activité ou d'en déléguer la réalisation partiellement ou totalement.

Implémentation

Introduction

Nous commencerons par spécifier l'objet activité libre. Ensuite, nous étudierons les problèmes posés par l'insertion d'un plan de réalisation au sein d'une procédure et nous terminerons par un scénario validant les choix d'implémentation.

L'activité libre

Relation avec les corbeilles d'acteur

La corbeille de WoorKs est construite autour de concepts liés à l'exécution des activités et des procédures.

L'activité libre permet à l'utilisateur de spécifier une nouvelle situation de coopération. La corbeille étant l'accessoire d'interaction entre l'activité de

l'organisation et les acteurs, il faudra la doter de fonctionnalités de spécification des activités libres(supra **5.2.3. La corbeille**). Dès lors, une partie de la corbeille sera destinée à accueillir les tâches qui pourront être spécifiées.

Se pose alors le problème de savoir quand va se placer une tâche faisant partie d'une activité libre dans la partie spécification de la corbeille de l'acteur chargé de sa réalisation. Autrement dit, va-t-on lui permettre de spécifier sa tâche dès le début de la procédure à laquelle elle appartient ou dès la création de l'activité libre dont elle fait partie.

Si on choisit de créer la tâche lors de la création de l'activité libre, on fait perdre du temps à l'acteur chargé de sa réalisation car il aurait pu commencer sa spécification avant que la procédure ne crée l'activité libre.

Si on choisit de créer la tâche dès le début de la procédure, il faut attribuer la réalisation de l'activité libre dès le début de la procédure pour savoir à qui attribuer la tâche. De cette façon on permet à l'acteur de commencer la spécification de la tâche avant que l'activité dont elle fait partie ne soit déclenchée. Or, au début de la procédure, on ne peut être sûr de la création de cette activité libre, dans ce cas l'acteur aurait commencé une spécification qui ne servira jamais⁷.

Nous pensons qu'il faudra permettre au développeur de spécifier le moment de création de la tâche. Ce moment sera le début de la procédure, le moment de la création d'une activité en amont de la tâche, ou celui de la création de l'activité libre.

La gestion des documents (Fig. 5.6.)

La tâche dispose d'une folder in, d'un transient folder et d'un folder out. L'activité dispose d'attributs qui contiennent les documents en entrées et les documents résultats. L'activité libre dispose de méthodes qui lui permette :

- de transférer ses entrées au folder in de la tâche ;
- de choisir son alternative de sortie en fonction des documents se trouvant dans le folder out de la tâche;

⁷ Notons que le fait de préparer une tâche qui ne se réalisera jamais peut se produire dans le cadre de situations réelles.

- de transférer les documents du folder out vers ses attributs résultats en fonction de l'alternative de sortie qui a été déterminée.

Les relations entre le monde prédéfini et le monde ad hoc, telles qu'elles ont été présentées dans le cas de l'activité libre, limitent la liberté de réalisation de l'acteur. Il s'agit d'une limitation au niveau des types de documents qu'il devra manipuler et fournir lors de la terminaison de la réalisation. Ces contraintes sont nécessaires pour assurer la liaison entre l'activité libre et la suite de la procédure, du fait de la spécification de la structure des documents qui seront utilisés par les activités en aval de l'activité libre.

Il faut donc ajouter une méthode qui informera le réalisateur des types et des structures des documents qui sont attendus comme résultats de la tâche. Cette information doit être contraignante dans le sens où elle devra être respectée en vue de permettre l'utilisation de la méthode de l'activité qui consiste à évaluer le contenu du folder out pour déterminer l'alternative de sortie de l'activité. Cette méthode fait partie de l'activité libre et a été construite lors de la compilation du texte A.D.L.. Cette méthode est donc prédéfinie et ne pourra s'exécuter que si le type des documents qu'elle trouvera dans le folder out correspond à ce qu'elle attend.

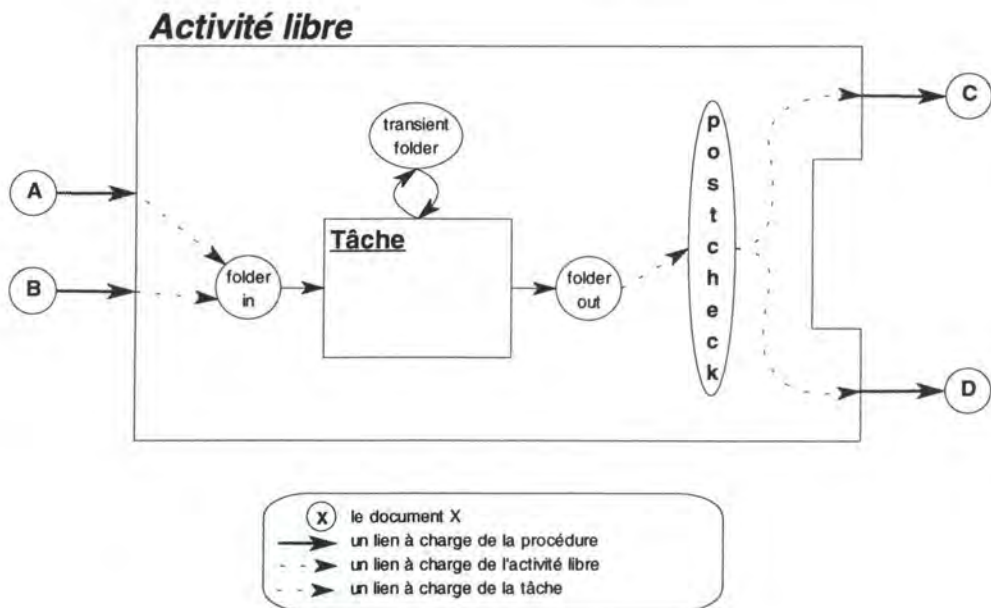


Fig. 5.6. Représentation d'une activité libre et de ses relations avec la procédure et la tâche

Interfaçage entre la procédure et le plan de réalisation d'une de ses activités libres

Nous illustrons notre propos par le figure X. Elle représente une activité libre qui s'exécute par un arbre de réalisation. Cet arbre a été spécifié à partir de la tâche A qui a été placée dans la corbeille du responsable de l'activité libre dès le début de la procédure.

Condition de déclenchement de la tâche A

La tâche A fait partie de l'activité libre. Cette dernière est caractérisée par une condition de déclenchement similaire aux conditions de déclenchement des activités classiques. Rappel : la condition de déclenchement d'une activité est l'activation de ses liens d'entrées et , éventuellement, le déclenchement d'un événement de retard.

La condition de déclenchement de la tâche A se divise en deux : d'une part, l'activation des liens d'entrées de l'activité et d'autre part, la survenance d'un événement de déclenchement propre au plan de réalisation et/ou la survenance d'un événement de retard de l'activité.

On ne peut déclencher le plan de réalisation sans le déclenchement de l'activité libre car c'est elle qui fournira les informations d'entrées sans lesquelles on ne peut commencer la réalisation du plan de réalisation. Donc, les événements de déclenchement en provenance du plan de réalisation ne seront pris en compte que si l'activité libre a été déclenchée.

Il existe donc trois possibilités de définition de la condition déclenchement pour l'activité libre. L'extension A.D.L. devra permettre de la spécifier par :

- *un événement de retard* ;
- *rien*, lorsque l'activation des liens d'entrées de l'activité libre déclenche la tâche ;
- *ACTA*, lorsque le déclenchement de la tâche est provoqué par l'activation du lien d'entrée et par la survenance d'un événement de déclenchement en provenance de l'arbre de réalisation⁸.

⁸ Il s'agit d'une synchronisation d'événements. Il faut donc disposer d'un mécanisme de mémorisation.

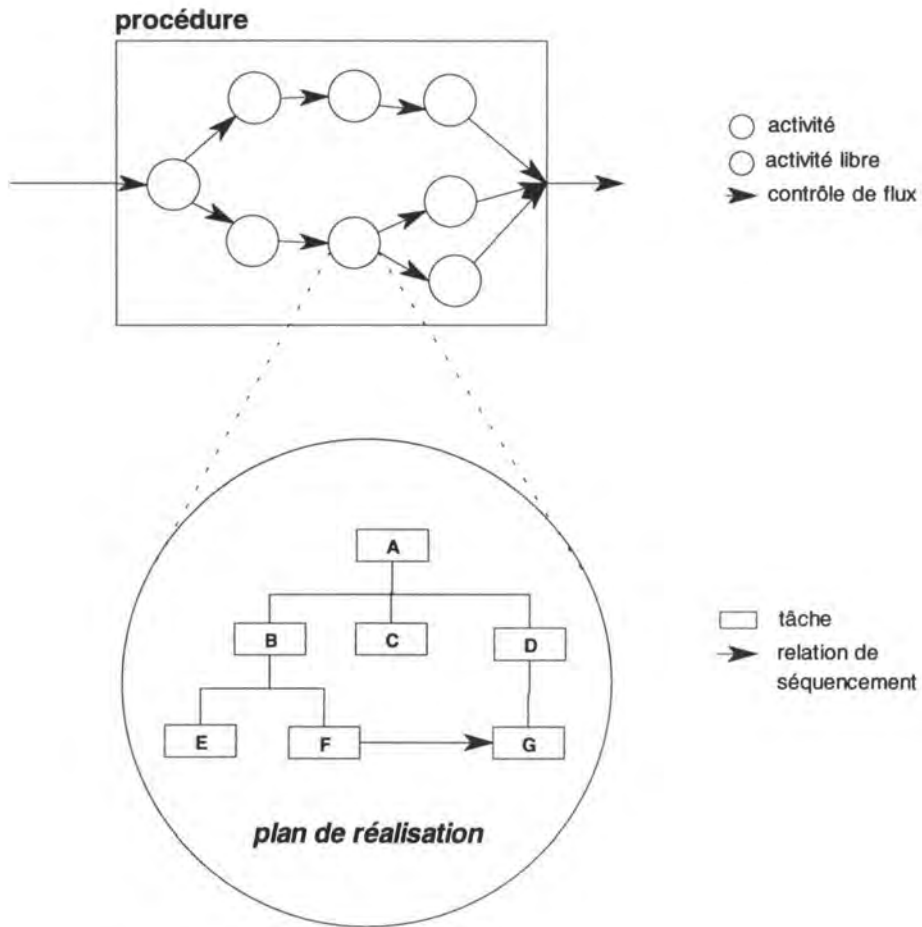


Fig. 5.7. Représentation d'une activité libre comprenant un plan de réalisation⁹

Echéance de réalisation de la tâche A

Lorsque le développeur d'application spécifie l'activité libre à l'aide du langage A.D.L., il peut spécifier une échéance de réalisation. Lors de la création de l'activité libre, le module de coordination créera un événement d'échéance. Si cet événement se déclenche avant la fin de la terminaison de l'activité libre, il aura comme effet la mise en urgence(supra 2.1.6.3. **Caractéristiques et spécification d'un activité**) de cette dernière. Il faudra répercuter cela sur la tâche A et sur le plan de réalisation¹⁰ dont elle fait partie.

⁹ supra **Chap. 3. : ACTA** pour plus d'information concernant les plans de réalisation

¹⁰ Cet exemple illustre l'existence de problèmes de coordination entre les tâches et les activités libres dû aux différences caractérisant leurs cycles de vie. La solution serait de les fusionner pour créer un cycle de vie commun.

Scénario

Nous allons mettre en mouvement l'ensemble des choix d'implémentation présentés ci-dessus en vue d'en vérifier la faisabilité. Il ne s'agit pas de donner une description exacte de la nature et de l'ordonnancement des processus qui seront mis en oeuvre¹¹, mais d'en donner une idée.

Exécution d'une procédure contenant une activité libre

La procédure a été démarrée par un initialisateur potentiel à partir de sa corbeille. Supposons que l'ensemble des activités en amont de l'activité libre soit terminé et que la prochaine activité soit une activité libre.

Le module de coordination (*supra* **2.1.7.4.3. L'exécution du workflow**) :

- crée l'activité libre ;
- prend les données de l'activité libre dans le contexte de la procédure et les places dans le contexte de l'activité libre ;
- regarde s'il doit créer la tâche contenue par l'activité¹²;
- s'il l'a créée, il la place dans la partie spécification de la corbeille de l'acteur affecté à sa réalisation ;
- vérifie la condition de déclenchement de la tâche :
 - si c'est un *événement de retard* : il crée l'événement de retard ;
 - si c'est *rien* : il utilise les méthodes de l'activité libre pour transférer les données de l'activité libre vers le folder in de la tâche. Ensuite il déclenche la tâche et la place dans la partie réalisation de la corbeille de l'acteur affecté à sa réalisation¹³ et rend actif les événements propres au plan de réalisation ;
 - si c'est *ACTA* : il rend actif les événements propres au plan de réalisation ;
- lorsque la tâche se termine, le module de coordination:

¹¹ Cf. Les ingénieurs de la connaissance de T.A.O. sont en train de développer une nouvelle version du module de coordination. Il nous est impossible de décrire exactement les processus d'interfaçage car nous ne disposons d'aucunes de ses spécifications.

¹² Cf. définition par le développeur d'application du moment de création de la tâche, à savoir : au début de la procédure, lors de la création d'une certaine activité ou lors de la création de l'activité libre.

¹³ Remarquons que la tâche se trouve à la fois dans la partie spécification et réalisation de la corbeille du réalisateur. Cela est dû au fait que la tâche peut être soit exécutée, soit réalisée.

- prévient l'activité libre de la terminaison de la tâche ;
 - effectue la postcheck de l'activité libre pour évaluer les éléments qui se trouvent dans le folder out en vue de déterminer son alternative de sortie. Si les documents du folder out ne correspondent pas à ce qui a été prédéfini, il faut retourner à la tâche pour demander à l'utilisateur de les transformer ou de les compléter ;
 - enlever la tâche de la corbeille de son réalisateur ;
 - détruire la tâche ;
 - prévenir la procédure de la terminaison de l'activité libre en lui passant l'identifiant de son alternative de sortie ;
 - en fonction de l'alternative de sortie, faire passer les résultats dans le contexte de la procédure ;
 - destruction de l'activité libre ;
-
- crée les activités en aval ;
 - etc.

5.3.2.2.3. Redéfinition de la réalisation d'une activité

Les procédures de WoorKs étant prédéfinies, il existe une séparation significative entre les moments de spécification et de réalisation. De ce fait, lorsqu'un acteur est assigné à la réalisation d'une activité, c'est-à-dire lorsqu'il la reçoit dans sa corbeille, il est probable que la définition de la réalisation qui lui est imposée soit inadéquate par rapport au contexte actuel. Il faudrait lui permettre de redéfinir la réalisation de l'activité à laquelle il est affecté en vue de l'adapter au contexte du moment.

Inadéquation de la réalisation imposée

Comment l'acteur va-t-il évaluer l'inadéquation de la réalisation qui lui est proposée?

L'acteur ne reçoit pas une description de la réalisation qu'il devra accomplir, il est mené par le système qui lui présente les outils et les données à utiliser. La seule connaissance de la réalisation qu'il peut avoir est une connaissance acquise par expérience de l'activité à laquelle il est affecté. Donc, s'il n'est pas habitué à l'activité qui se présente dans sa corbeille, il ne peut savoir s'il est préférable d'en redéfinir la réalisation.

Le moment de redéfinition

Va-t-il redéfinir avant de commencer la réalisation ou pendant la réalisation ?

Le manque d'information concernant la réalisation d'une activité (supra **Innadéquation de la réalisation imposée**) engendre des situations au cours desquelles l'acteur se rend compte de l'inadéquation de la réalisation après la prise en charge de l'activité. Se pose alors la question de savoir s'il souhaite récupérer le travail accompli et s'il en a la possibilité.

Nous avons choisi de lui permettre de récupérer son travail s'il le désire.

Permettre à l'acteur de redéfinir la réalisation de la tâche à laquelle il est affecté

Il s'agit d'une situation analogue à celle d'une activité libre pour laquelle le moment de création de la tâche est équivalent au moment de création de l'activité libre.

Implémentation

Introduction

L'idée consiste à remplacer l'activité à redéfinir par une activité libre (Fig. 5.8. et Fig. 5.9.). Une fois que l'utilisateur a fait son choix parmi les éléments qu'il désire réutiliser, la situation est similaire au cas de l'activité libre.

Fonctionnalités offertes à l'acteur

Pour permettre à l'acteur de manifester son choix de redéfinition, il faut lui offrir un bouton poussoir qui lui serait présenté tout au long de la réalisation de l'activité. S'il sélectionne ce bouton, il démarre le processus de redéfinition.

Le processus de redéfinition commence par le choix des éléments qu'il désire garder pour entamer l'exécution de la tâche. Ces documents seront placés dans le transient folder de la tâche. Il faut avoir des méthodes propres à l'activité qui permettent de faire passer ses entrées initiales dans le folder in et les entrées modifiées plus les variables locales dans le transient folder. Ces méthodes seront utilisées par un accessoire interactif qui permettra à l'utilisateur de faire son choix.

Une fois le choix réalisé, il faut informer le réalisateur des types de documents que l'on attend comme sortie de la procédure. Il faut aussi avoir des méthodes propres à la postcheck (Fig. 5.9.) qui permettent de choisir l'alternative de sortie de l'activité libre en fonction des types de documents se trouvant dans le folder out de la tâche.

Les relations avec la corbeille de l'utilisateur sont :

- l'enlèvement de l'activité de la partie réalisation ;
- le placement de l'activité libre dans les parties réalisation et spécifications.

Représentation

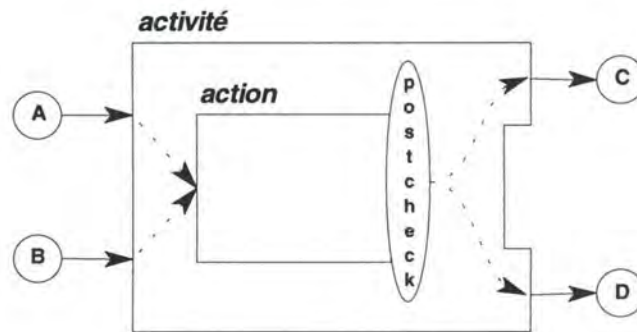


Fig. 5.8. Représentation d'une activité dont la réalisation est définie par une action

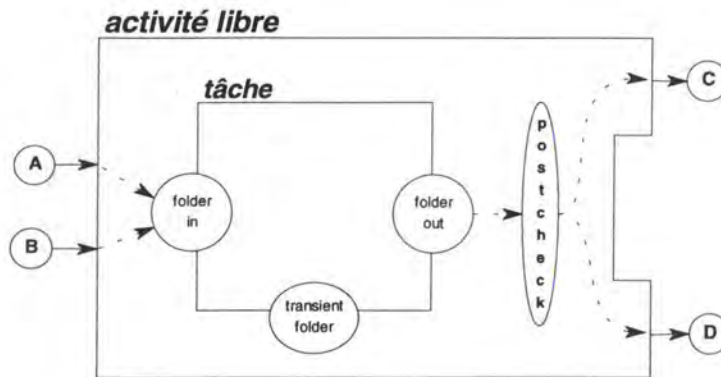


Fig. 5.9. Représentation d'une activité dont la réalisation est libre

Enrichissement de la sémantique du langage A.D.L.

Pour offrir l'ensemble des fonctionnalités citées ci-dessus, il faut que le langage A.D.L. permette de définir la possibilité de redéfinition de la réalisation des activités. La compilation des activités dont la réalisation est redéfinissable créera, en plus des caractéristiques propres aux activités,

- pour chaque entrée et variable intermédiaire, une méthode qui permette de les copier dans le folder in ou dans le transient folder de la tâche ;
- pour chaque entrée, des attributs de sauvegarde destinés à restaurer les valeurs initiales de l'activité en cas de redéfinition pour laquelle l'utilisateur ne désirerait garder aucune valeur intermédiaire ;
- une méthode d'information de l'utilisateur qui, en fonction de la structure de la procédure, informerait l'acteur du type de documents que l'on attend à la sortie de l'activité libre ;
- une méthode de postcheck qui, en fonction des éléments du folder out, choisira l'alternative de sortie de l'activité.

N'oublions pas les événements propres à l'activité: il faut les modifier pour qu'ils fassent référence à l'activité libre au lieu de l'activité. Par exemple, si l'activité à une échéance de réalisation, il faut la répercuter sur l'activité libre et sur la tâche qui compose cette dernière.

5.3.2.2.4. Greffage d'une activité

L'idée serait de permettre à un acteur de compléter la réalisation de l'activité qu'il est en train de réaliser. Cette idée n'est pas neuve; on peut lire dans [AGOS 93] :

' 3. WoorKs & UTUCS: Coupling a workflow management system with a communication support system

A work process is a two-facet object: that is, when workflow follows the previously defined plans, it is made of activities to be done by various actors (the routine dimension of the work process is emphasized); and when a breakdown blocks the planned workflow, it becomes a subject of conversation among the same group of people (the creative dimension of work processes is emphasized) '.

[AGOS 93] annonce un système intégré de support aux situations de coopération répartie prédéfinie et ad hoc. Nous verrons qu'il n'en est rien en montrant qu'il ne s'agit que de l'utilisation des relations de greffage simplifiées.

Cet article est illustré par une procédure de gestion d'une demande de crédit. Un client se présente à la banque, remplit des formulaires de demande de crédit qui seront traités en vue d'accorder ou de refuser ce qui a été demandé.

L'article annonce une prise en compte des exceptions par UTUCS. Parmi les exceptions présentées, on trouve :

- la possibilité de demander un complément d'informations ;
- la possibilité de contacter une tierce personne ;
- etc.

Il s'agit en fait d'initialiser une conversation [WINO 87] qui a pour objet une activité de WoorKs. [AGOS 93] présente UTUCS comme le gestionnaire des exceptions en disant que, dès qu'une exception survient, on ouvre une communication qui, par exemple, fournira l'information manquante.

Or, nous avons vu (supra **5.2. Choix d'implémentation pour ACTA**) que UTUCS ne représente qu'une faible partie de la situation de coopération ad hoc. Il ne représente pas l'activité des acteurs qui participent à une conversation.

Ex. : si on demande à un acteur si on peut accorder un prêt qui est une exception, on lui passera les informations nécessaires et on attendra une réponse positive ou négative.

L'acteur qui devra donner son avis dans le cadre de l'exemple ci-dessus ne dispose d'aucune aide pour effectuer la nouvelle activité qui se présente à lui. Il ne dispose d'aucun outil de réalisation ou de spécification de la nouvelle situation ad hoc. Tous ce qu'il peut faire s'est initialiser de nouvelles conversations avec d'autres acteurs.

Or, WoorKs dispose d'un ensemble de petits outils représentés par des opérateurs. Parmi ces outils, certains pourraient être réutilisés par l'acteur qui devra effectuer une activité ad hoc (infra **5.2.4.2.4. Utilisation du modèle des opérateurs**).

En utilisant ACTA comme troisième module (en plus de WoorKs et UTUCS), on offre cette boîte à outils et une représentation plus complète de la situation de coopération ad hoc.

Reprenons l'exemple de la procédure d'évaluation d'une demande de prêt présenté dans [AGOS 93]. Supposons qu'un client se présente avec des documents qu'il dit être

équivalents à ceux demandés pour effectuer la demande de prêt. La personne qui enregistre cette demande n'est pas sûre de la légalité de la substitution des documents proposée par le client.

Dès lors, elle crée une nouvelle tâche qui aura comme objectif d'obtenir un supplément d'information. Elle choisit, soit de déléguer cette tâche à un membre du département légal, soit de la réaliser elle-même en consultant une base de données contenant le règlement.

Si la personne préposée choisit de réaliser elle-même la tâche, elle démarre l'outil de réalisation et parcourt les bases de données grâce au browser du gestionnaire des documents.

Par contre, si elle choisit de déléguer la tâche, un acteur du département légal va la recevoir dans sa corbeille. Cet acteur aura le choix soit de la réaliser, soit de continuer la spécification de la situation de coopération en complétant le plan de réalisation auquel la tâche appartient (par exemple en la divisant).

Une fois que l'on a identifié et réalisé l'objectif complémentaire, il faut savoir comment réutiliser les documents résultats dans le cadre prédéfini.

Notre système ainsi que [AGOS 93], ne permet pas de d'inclure dans le flux prédéfini les documents obtenus de manière ad hoc. Ils ne servent qu'en consultation. C'est-à-dire qu'un acteur demandant un supplément d'informations doit faire le lien mentalement entre ces informations et la suite de l'exécution de l'activité. Il ne peut l'utiliser en tant qu'objet informatique dans la spécification prédéfinie de la réalisation.

Il faut néanmoins lier la tâche greffon et l'activité greffée, de manière à respecter l'inclusion du cycle de vie de la tâche greffon dans celui de l'activité greffée.

5.4. Implémentation d'ACTA-WoorKs

5.4.1. Introduction

Nous allons voir comment réutiliser l'environnement logiciel de WoorKs pour implémenter le système intégré ACTA-WoorKs supportant les situations de coopération répartie prédéfinie et ad hoc.

Nous allons revisiter les modèles de WoorKs de manière à identifier les éléments qui seront réutilisés ou modifiés en vue d'implémenter ACTA-WoorKs. Pour faciliter l'approche du lecteur, nous les reprendrons dans un ordre identique à celui de leur présentation (supra **2.1. La connaissance générique de WoorKs**).

Ensuite nous aborderons les modèles qui devront être ajoutés au système pour tenir compte des spécificités d'ACTA.

Nous terminerons cette introduction en regrettant de ne pas disposer de plus de temps pour préparer la présentation de l'utilisation des modèles d'ACTA-WoorKs. La présentation de WoorKs étant relativement complète, il est dommage d'escamoter la présentation de son utilisation car, en se basant sur les éléments de cette présentation (supra **Chapitre 2 : WoorKs**), il est possible de parler très concrètement de l'environnement ACTA-WoorKs. Malheureusement, le manque de temps nous contraint à restreindre nos objectifs.

5.4.2. Utilisation des modèles de WoorKs

5.4.2.1. Utilisation du modèle organisationnel

Le modèle organisationnel est plus complet que le modèle des acteurs d'ACTA. Il représente la structure de l'entreprise de manière plus complète. Il tient compte de l'environnement organisationnel de l'entreprise; cela permet de représenter les informations relatives aux clients, aux fournisseurs ou à tout autre personne avec laquelle l'entreprise est en contact.

Pour toutes ces raisons nous pensons qu'il serait judicieux de réutiliser le modèle de WoorKs presque tel quel, en sachant qu'il serait nécessaire d'effectuer de petites modifications en vue de supporter l'activité de délégation propre à ACTA. Par exemple, il faudrait développer la notion de rôle en y ajoutant une explication qui permettrait au déléguant de choisir la (ou les) personne(s) à qui s'adresser pour déléguer la réalisation d'une tâche d'un certain type.

Dès lors, il faut identifier des méthodes qui permettront de lier la phase initiale du processus de délégation avec la représentation des compétences organisationnelles.

De plus, il faut pouvoir initialiser un protocole de communication avec le concepteur d'une tâche, avec le réalisateur d'une tâche. Donc, il faut que les acteurs ou le groupe d'acteurs soient accessibles par le biais d'une adresse mail qui sera automatiquement dérivée du modèle de l'organisation lorsque l'on voudra les contacter.

5.4.2.2. Utilisation du modèle des références

Rappelons :

- une référence est le lien par lequel WoorKs peut accéder à un objet externe;
- un objet externe est un objet qui est manipulable par une application externe à WoorKs.

Les objets d'informations manipulés par ACTA sont des objets externes qui sont créés par les outils de la boîte à outils de l'outil de réalisation. Donc, ces objets seront référencés soit par un objet référence locale, soit par un objet référence IMAGEWork. Ce dernier permettant l'utilisation de l'outil de manipulation de documents multimédia IMAGEWORK.

5.4.2.3. Utilisation du modèle de l'information

Il représente l'ensemble des objets d'information qui circulent au sein de la représentation d'une situation de coopération prédéfinie et ad hoc.

Les objets d'information manipulés par WoorKs sont, soit des objets externes, soit des objets internes. Les objets externes sont référencés et manipulés par des applications externes. Les objets internes sont des objets du modèle de l'information que l'on manipule sans avoir recours à une application externe.

L'intégration d'ACTA et de WoorKs a comme conséquence l'obligation, pour ACTA, d'être capable de manipuler des documents internes tels que des bons de commandes, des fiches de paie, etc.

5.4.2.4. Utilisation du modèle des opérateurs

WoorKs utilise les actions pour spécifier la réalisation des activités. Ces actions sont des opérateurs; elles sont caractérisées par des entrées, des sorties et des règles de transformation des premières en secondes. Les actions se décomposent en opérateurs interactifs, externe, de documents, etc. Chacun d'eux a un objectif qu'il atteint en transformant ses entrées en sorties.

L'idée serait de pouvoir réutiliser cet ensemble d'opérateurs de manière à les mettre à la disposition des acteurs dans le cadre de la réalisation de tâches ad hoc. Ils feraient partie

de la boîte à outil de l'outil de réalisation; ils pourraient être utilisés par l'acteur qui choisirait les entrées de l'opérateur dans le folder in de sa tâche et retrouverait les sorties dans le transient folder. Pour cela, il faut que les acteurs soient informés de l'objectif de ces opérateurs et des documents attendus en entrées et produits en sorties.

De plus, WoorKs utilise les opérateurs externes pour avoir accès aux applications externes. De la même manière, la boîte à outils de l'outil de réalisation devra avoir accès aux applications externes.

5.4.2.5. Utilisation du modèle du temps

Parce qu'ACTA et WoorKs ont un même domaine d'application, à savoir les situations de coopération répartie, il existe des éléments communs. C'est le cas pour la représentation du temps. On ne peut concevoir un système de gestion de la coopération censé organiser le travail d'une organisation sans se baser sur une représentation du temps.

Le modèle du temps de WoorKs offre plus que ce dont a besoin ACTA, ce dernier n'ayant besoin d'une gestion d'événements que pour représenter les conditions déclenchement temporelles et les échéances de réalisation des tâches.

En plus de la gestion des événements, WoorKs offre un calendrier général et un agenda personnel. Le premier représente l'emploi du temps de l'ensemble de l'organisation, à savoir : les horaires de travail, les jours fériés, etc. Le second représente l'emploi du temps de chaque acteur de l'organisation.

La gestion des événements offre des événements directs qui représenteront les conditions de déclenchement et les échéances de réalisation des tâches. En plus, le mécanisme de monitoring assure le suivi d'avancement d'un plan de réalisation; par exemple, le concepteur d'une tâche peut demander à être prévenu deux jours avant l'échéance de la réalisation de la tâche qu'il a déléguée (il pourra contacter le délégataire pour l'inciter à respecter l'échéance de réalisation de cette tâche).

5.4.2.6. Utilisation du modèle des procédures

Le modèle des procédures représente les workflows. Pour construire ACTA-WoorKs nous avons identifié un ensemble de relations qui assurent le passage du monde prédéfini au monde ad hoc. Pour les mettre en oeuvre, il faut modifier le modèle de procédure. Illustrons une partie de ces modifications en reconsidérant le cas de l'activité libre.

5.4.2.6.1. L'activité libre

Elle fait partie d'une procédure prédéfinie. Cette procédure dispose de méthodes qui lui permettent de créer les activités qui en font partie. La création de l'activité libre doit se faire différemment. Il ne faut pas la placer dans la corbeille de l'utilisateur. Il faut créer la tâche qu'elle contient et la placer dans la corbeille de l'utilisateur si cela n'a pas encore été fait (cf. choix pour le développeur de spécifier le moment de création de la tâche).

Si la tâche a été créée avant l'activité libre, alors son responsable peut avoir entamé la spécification d'un plan de réalisation. Les tâches faisant partie de ce plan ont des conditions de déclenchement et des échéances de réalisation qui sont représentées par des événements temporels. Or, le déclenchement de la tâche ne peut avoir lieu avant le déclenchement de l'activité. C'est pourquoi tous les événements des tâches du plan de réalisation sont inactifs jusqu'au déclenchement de l'activité libre. Au déclenchement de l'activité libre, on consulte la condition de déclenchement de la tâche étant le sommet du nouveau plan de réalisation :

- si sa condition de déclenchement est un événement de retard, il faut le créer ;
- si sa condition de déclenchement est vide, il faut déclencher la tâche et rendre actif l'ensemble des événements propres au plan de réalisation dont elle est le sommet;
- si sa condition de déclenchement est 'ACTA', on ne la déclenche pas et on rend actif l'ensemble des événements propres au plan de réalisation dont elle est le sommet;

Notons une indétermination quant à la condition de déclenchement de la tâche sommet, du fait de l'indétermination du moment de déclenchement de l'activité libre. Certaines règles de contrôle d'ACTA ne pourront être vérifiées lors des étapes de construction du plan de réalisation précédent le déclenchement de l'activité libre. Par exemple, si la condition de déclenchement de la tâche contenue par l'activité libre est vide, son déclenchement sera provoqué par le déclenchement de l'activité libre. Or, si cette tâche a été créée avant le déclenchement de l'activité libre, l'acteur concepteur a pu la décomposer et spécifier des conditions de déclenchement pour les sous-tâches. Comme on ne connaît pas la condition de déclenchement de la tâche sommet, on ne peut vérifier si les conditions de déclenchement des sous-tâches correspondent à des moments postérieurs au moment de déclenchement de la tâche décomposée.

5.4.2.7. Utilisation des objets T.O.I

Rappelons que T.O.I. est un framework de génération d'interfaces utilisateur. Il offre des objets tels que des boutons poussoirs, des menus, des boîtes de dialogue, etc. Les ingénieurs d'application de WoorKs ont développé l'interface des accessoires utilisateurs à l'aide de ces objets. Parmi ces accessoires, on trouve notamment la corbeille[PONS 92 (4)] et l'agenda des utilisateurs[KUMA 90].

Les interfaces sont implémentées sous la forme d'objets de commande (supra **2.1.8.1.1. T.O.I.**).

Nous réutiliserons ces objets pour développer les objets de commandes qui assureront l'interface utilisateur de la nouvelle corbeille, de l'outil de réalisation des tâches et de l'outil de spécification et de gestion des plans de réalisation.

5.4.2.8. Utilisation du méta-modèle

Rappelons que le méta-modèle offre des mécanismes génériques de manipulation des documents tels que l'affichage, la modification, la création, etc.

On peut étendre ces mécanismes génériques au système intégré ACTA-WoorKs. On pourrait, par exemple, en faire bénéficier le browser du gestionnaire de document qui parcourt les bases de données et visualise les documents.

Evidemment, parce que les documents qui sont manipulés par ACTA sont définis de manière ad hoc, il faut envisager certains mécanismes qui les prendront en charge. Par exemple, si on introduit dans ACTA la manipulation d'objet 'interne' (supra **2.1.3. Le modèle de l'information**) alors, il faut ajouter une fonctionnalité à l'outil de spécification qui permettra aux utilisateurs de définir dynamiquement la structure des documents manipulés. Dès lors, il faudrait ajouter la génération automatique d'un objet de présentation propre (supra **2.1.8.3.1. Les objets nommés**) et la mise à jour automatique de l'objet interface (supra **2.1.8.1.2. La représentation des types**)

5.4.3. Les nouveaux modèles

5.4.3.1. Le modèle des plans de réalisation

Il faudra créer un modèle de représentation des plans de réalisation qui sera chargé de la mise en place des règles de contrôle et de guidances. Il mettra en oeuvre les mécanismes d'induction et prendra en charge les processus de délégation, de report d'échéances, etc.

5.4.3.2. Le système de communication

Jusqu'a présent, nous l'avons évoqué assez rapidement (Fig 5.10.). Il faudra préciser l'ensemble des fonctionnalités qu'il devra remplir et spécifier son interfaçage d'une part, avec ACTA et d'autre part avec UTUCS.

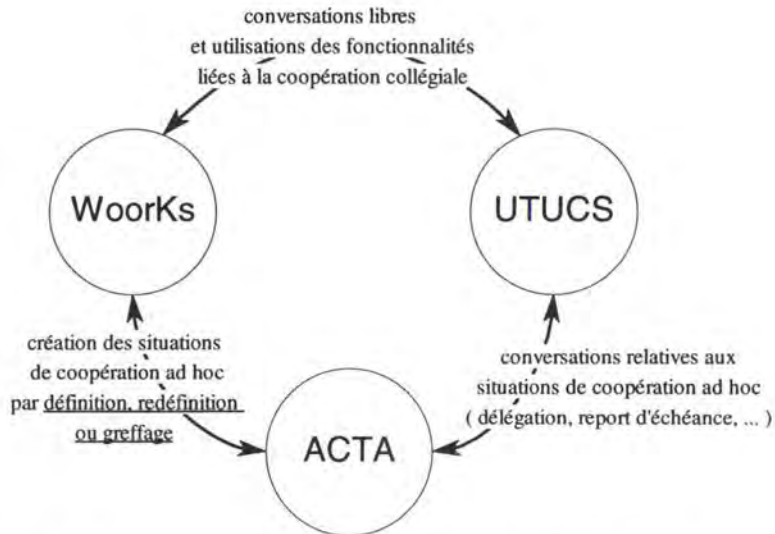


Fig. 5.10. relation entre WoorKs, ACTA et UTUCS

Conclusion

Au terme de ce travail, il nous reste à nous demander dans quelles mesures nous avons atteint notre objectif initial.

Dans le but de mettre sur pied un système intégré supportant les situations de coopération répartie prédéfinie et ad hoc, nous avons adopté une démarche qui s'est voulue la plus concrète possible. On a rapidement constaté que cette démarche avait des exigences dépassant de loin le cadre d'un travail de fin d'études.

C'est pourquoi, bien que nous ayons posé un certain nombre de bases au développement du système intégré ACTA-WoorKs, il subsiste de nombreuses zones d'incertitudes. Parmi celles-ci, notons la partie consacrée aux choix d'implémentation d'ACTA et celle consacrée à l'implémentation du système ACTA-WoorKs. Ces aspects restés en friche devront faire l'objet d'études plus approfondies.

Par ailleurs il faut noter que WoorKs est un système en perpétuelle évolution : ainsi, la connaissance générique est l'objet d'aménagements perpétuels; en outre, certains chercheurs (T.A.O.) sont en train de transférer l'entièreté de l'environnement du système UNIX vers le système P.C. Dès lors, certains de nos choix s'avéreront inadéquats au regard des modalités actuelles d'implémentation de WoorKs. Ainsi, par exemple, la construction de l'interface graphique à l'aide des objets T.O.I. se révèlera certainement obsolète vis-à-vis des possibilités offertes par les environnements de développement PC tels que le *Visual C++*.

orehe
dyhe

Bibliographie

- [WINO 87] Terry Winograd, "A Language/Action Perspective on the Design of Cooperative Work.", *Human-Computer Interaction*, vol. 3, no. 1., 1987/88, pp. 3-30.
- [WINO 88] Terry Winograd, "Where the Action Is", *Byte* décembre 1988, pp. 256-261.
- [BODA 89] François Bodart, Yves Pigneur, *Conception Assistée des systèmes d'information*, deuxième édition, Masson.
- [ADER 90] Martin Ader, Gang Lu, Sudarshan Murthy, Patrick Pons, Kumar Venkataram, WoorKS-1, the first prototype of an object oriented workflow system for offices.
- [BIGNO 90] Celsina Bignoli, Giorgio De Michelis, Carla Simone, Renata Tinini, *User-to-user communication support : Updating the state of the art and specifications of the system*, Department of Computer Science (University of Milan).
- [KUMA 90] Kumar, Cool source file : cmdAgenda.t , Copyright Bull S.A., 1991, 1992, this software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [MALO 90] Malone Th. W. and CROWSTON K. : What is Coordination Theory and How Can It Help Design Cooperative Work Systems ?, in *CSCW90 Proceedings*.
- [VENKA 90] Kumar Venkataramen, Cool source file: timeObj.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [ADER 91] Martin Ader, Sudarshan Murthy, Narayana Murthy, Raghunandan, *Meta Model Reference Manual*.
- [AGOS 91] Alessandro Agostini, Celsina Bignoli, Giorgio De Michelis, Stefano Patriarca, Renata Tinini, *UTUCS - User Manual*, Dipartimento di Scienze dell'Informazione (Università di Milano).
- [MEIN 91] J.P. Meinadier, *L'interface utilisateur : Pour une informatique plus conviviale*, Dunod 1991
- [SCHM 91] Riding a tiger, or Computer Supported Cooperative Work. In Bannon, L.; Robinson, M; Schmidt, K. (eds). *Proceedings of the Secon European Conference on Computer-Supported Cooperative Work*. Kluwer Academic Publisher, Dordrecht/ Boston/ London: 1-16.

- [ADER 92 (1)] Martin Ader, Sudarshan Murthy, Narayana Murthy, Raghunandan R.S, Carlos Bergnes, Josep Monguio, Organisation model reference manual.
- [ADER 92 (2)] Martin Ader, cool source file name : op.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [ADER 92 (3)] Martin Ader, cool source file name : startOp.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [ADER 92 (5)] Martin Ader, Patrick Pons, Initial cool prototype of sample operators.
- [ADER 92 (6)] Marin Ader, Patrick Pons, WoorKS-2, Protorype of the context modul.
- [ADER 92 (7)] Martin Ader, Patrick Pons, WooRKS's information modul.
- [ADER 92 (8)] Martin Ader, Patrick Pons, Initial Cool prototype using IMAGEWorks (Bull Multimedia product).
- [ADER 92 (9)] Martin Ader, Patrick Pons, Initial Cool prototype of IMAGEWorks interface.
- [ADER 92 (10)] Martin Adre, Kadaba SRIVATHS, WooRKS programmers'guide.
- [CALD 92 (1)] Angel Calderon, Josep Mongui'o, cool source file name : proc.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [CALD 92 (2)] Angel Calderon, Josep Mongui'o, cool source file name : copprocX.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [CALD 92 (3)] Angel Calderon, cool source file name : copactSX.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [CALD 92 (4)] Angel Calderon, cool source file name : copactVX.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [CALD 92 (5)] Angel Calderon, cool source file name : copactX.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705). [COOL 92 (1)] Siemens Nixdorf Information Systeme AG 1992, Cool (SINIX), Programmer's Reference Guide. April 1992
- [COOL 92 (2)] Siemens Nixdorf Information Systeme AG 1992, Cool (SINIX), Language Reference Manual. April 1992
- [DI GI 92] Raffaele Di Giovanni, Stefano Patriarca, *Workflow management for financial service : functional specifications for the credits on bank account workflow*, Datamont R&D 1992.
- [LU 92] Gang Lu, Developer Manual of TOI (Version 1.2).
- [PONS 92 (1)] Patrick Pons, cool source file name : mail.t.
- [PONS 92 (2)] Patrick pons, cool source file name : otherops.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [PONS 92 (2)] Patrick Pons, cool source file name : action.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).

-
- [PONS 92 (3)] Patrick Pons, cool source file name : wfBasket.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [PONS 92 (4)] Patrick Pons, cool source file name : cmdBasket.t. This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [PONS 92 (5)] Patrick Pons, cool source file name : context.t . This software was produced as part of the ESPRIT project ITHACA-2 (2705).
- [SRIVA 92] Kadaba Srivaths and Martin Ader, WoorKS Programmers'Guide, Bull S.A. 1992.
- [SRIVA 92 (2)] Kadaba Srivaths, Martin Ader, Kumar Venkataraman, An Object-Oriented Time Representation - Programmer's Reference
- [SRIVA 92 (3)] Kadaba Srivaths, Martin Ader, An Object-Oriented Time Representation - Programmer's Guide
- [AGOS 93] A. Agostini, G. De Mechilis, M.A. Grasso, S. Patriarca, *Reengineering a Business Process with an Innovative Workflow Management System : a Case Study*. COOCS'93-11/93/CA, USA
- [BOGI 93] Douglas P. Bogia, William J. Tolone, Simon M. Kaplan and Eric de la Tribouille, Supporting Dunamic Interdependencies among Collaborative Activities, ACM 1993.
- [LU 93] Gang Lu and Martin Ader, Building workflow application on top of WoorKS.
- [MOLLE 93] Joaquim Mollera, Josep Mongui'o, ADL and its Compiler revised version.
- [PETIT 93 (1)] Thérèse Petitjean, Elaboration du budget dans une institution universitaire, Etude de cas d'analyse d'un processus de prise de décision.
- [PETIT 93 (2)] Thèrèse PetitJean, Situation d'affectation multiple : la relation de duplication multiple. Working paper, 1993.
- [SCHA 93] Tomas Schael, Supporting Cooperative Processes With Workflow Management Technology. Tutorial notes COOCS '93.
- [SWEN 93] Keith D. Swenson, Visual Support for Reengineereing Work Processes. ACM 1993.
- [LESSU 94] Roland Lessuise, Cours de théorie des organisation, matières approfondies. (donné dans le cadre de 3ème licence en informatique, filière système d'information)
- [PETIT 94] Thérèse Petitjean, Contribution à la spécification de situations de coopération ad hoc et à leur prise en compte dans les systèmes de workflow. Facultés universitaires N.D. de la Paix, Institut d'Informatique 1994.
-