



Rapport sur Articles de Recherche

Architectures

Présenté par

Rabih TOUT & Yuqing GUO

M2 Recherche Informatique (RTS)

Sommaire

Article 1	3
1. Introduction.....	4
2. Les modèles classiques.....	5
2.1 Modèle en Cascade (Waterfall).....	5
2.2 Modèle en Spirale.....	6
2.3 Le Modèle en V.....	6
3. Les étapes recherche.....	7
3.1 Décomposition des Fonctions.....	9
3.2 Scénarios.....	9
3.3 Les Principes.....	10
3.4 Requirement.....	10
3.5 Modèle de l'Interaction.....	11
3.6 User-Interface.....	11
3.7 L'Architecture.....	11
3.8 L'Abstraction.....	11
3.9 L'Implémentation de l'infrastructure.....	12
3.10 Le développement d'application.....	12
3.11 L'évaluation.....	12
3.11.1 Inspection.....	12
3.11.2 Simulations.....	12
3.11.3 Problèmes standard.....	12
3.11.4 User studies.....	12
3.11.5 Programming data.....	13
3.11.6 Performance data.....	13
4. La conclusion et l'avenir.....	13
Article 2	15
1. Introduction et motivation.....	16
2. Département de radiologie.....	16
2.1 Définition.....	16
2.2 Fonctionnement.....	16
2.3 Infrastructure.....	18
3. Problématique.....	19
4. Approche proposée.....	19
5. Infrastructure orientée par travail.....	20
Conclusion.....	21

Article 1

An Integrated Approach to Designing and Evaluating Collaborative Applications and Infrastructures

PRASUN DEWAN

Department of Computer Science, University of North Carolina,

1. Introduction

Système coopératif inclut les infrastructures communes et les applications spécifiques pour supporter les travaux coopératifs.

Par exemple, des systèmes des infrastructures coopératives :

XTV (Abdel-Wahab et Jeffay, 1994) : X Teleconferencing and Viewing

Rendezvous : Hill et al. , 1994

GroupKit : Roseman ET Greenberg, 1996

Prospero : Dourish , 1995

MASSIVE : Greenhalgh et Benford, 1995

COLA : Trevor et al., 1994

COAST : Schuckmann et al.,

Trellis : Stotts et Furuta , 1994

COCA : Li et Muntz , 1998

Suite : Dewan et Choudhary 1992

Et des système des applications collaboratives:

GROVE outline editor : Elis et al ., 1991

PREP document editor : Neuwirth et al., 1992

GroupDesign graphics editor : Karsenty et al., 1993

FLECSE collaborative software development environment : Dewan et Riedl , 1993

Scrutiny : Gintell et al ., 1993

EGRET : Johnson , 1992

Ils offrent les infrastructures support système coopératif, on peut construire les systèmes coopératifs facilement.

Il y a beaucoup de recherches concernant le développement et l'évaluation du système coopératif. Identification de processus de développement est très important pour établir un système fort et souple. Il y a des modèles anciens comme modèle en Spirale(Spiral) et modèle en cascade (Waterfall) , mais ils décrivent des étapes du développement des logiciels arbitraux . Ils ne sont pas capables pour le système coopératif. Nous avons établi cette approche intégrée pour développer les applications collaboratives et les infrastructures collaboratives. Elle divise certaine étapes de modèle précédant en plusieurs étapes spéciales qui décompose la fonction du système en plusieurs subfonctions étant rechercher individuellement, choisit un principe général accordant le design et la construction de la fonction, identifie les scénarios collaboratifs en niveau multiple et abstrait, réalise les modèles d'interaction, développe l'architecture logique et l'architecture physique, développe infrastructure abstraite, réalise les applications, etc. Ce n'est pas tous les projets ont besoin de toutes les étapes. En effet, nous ne trouvons pas encore un projet qui a besoin de toutes les étapes.

Ce article est structuré comme suit : En première partie, nous introduisons les modèles en Waterfall et Spiral, ensuite, nous illustrons le modèle de notre approche

intégrée, et les étapes de l'évaluation d'application, enfin nous présentons la conclusion et la recherche de l'avenir.

2. Les modèles classiques

Notre approche a des relations avec Waterfall et Spiral. Ils définissent les façons généralement pour constituer un logiciel.

2.1 Modèle en Cascade (Waterfall)

C'est le modèle classique, il est utilisé pratiquement partout sous forme de nombreuses variantes. Le modèle développe le projet en quelques phases:

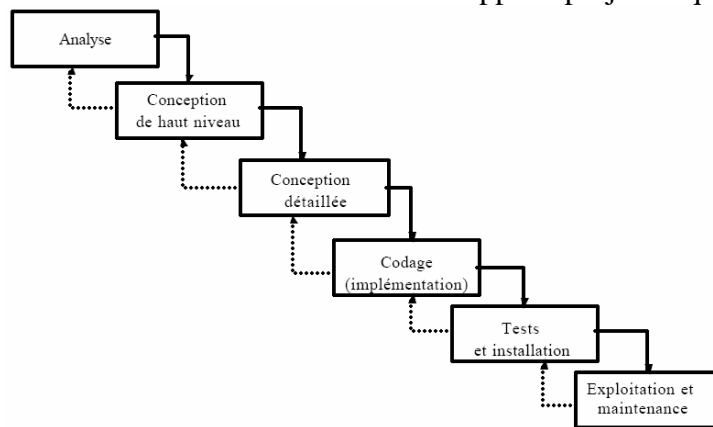


Figure 1 : Les étapes de modèle en cascade.

- Etude des besoins (system engineering): Commence par la perception d'un problème ou d'un besoin. Cherche un ensemble de besoins que le logiciel devrait satisfaire.
- Analyse : Analyse détaillée de toutes les fonctions et autres caractéristiques que le logiciel devra réaliser pour l'utilisateur, telles que vues par l'utilisateur.
- Conception générale (design) : Définition de l'architecture générale du logiciel. Pas de détails sur la manière dont les éléments composant le système seront implantés.
- Conception détaillée (design) : Spécification de la manière dont chacun des composants du logiciel sera réalisé et dont ils interagiront.
- Codage et test unitaire (code) : Fabrication de tous les composants du système et tests de chacun des programmes individuellement.
- Test d'intégration (testing) : Test de l'ensemble du système.
- Exploitation : Montage du système dans son environnement opérationnel, formation des usagers, mise en opération.
- Maintenance (maintenance) : Correction, adaptation et perfection du système.

La phase suivante dépend de la phase précédente, et on revient à la phase précédente par le feedback de la phase suivante. C'est un modèle clair et systématique, il est facile à organiser.

2.2 Modèle en Spirale

Le Spiral est un modèle hiérarchique qu'il développe le logiciel en quelque cercle, la production de chaque cercle est une version de la production finale, et l'objet produit dans un cercle dépend le cercle précédent. Le modèle en cascade ne supporte pas la notion de la constitution basée du cercle précédent.

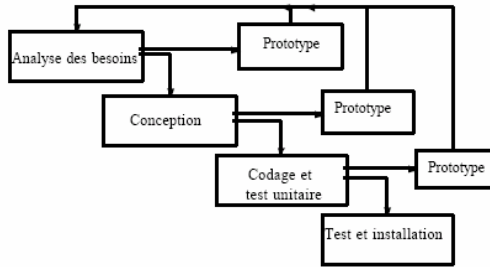


Figure 2

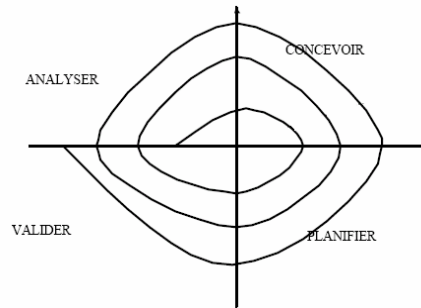


Figure 3

Chaque cercle représente une activité de l'analyse vers le développement. Les cercles sont divisés en plusieurs secteurs : analyse, concevoir, planifier et valider. Des prototypes sont développés pour chaque activité.

2.3 Le Modèle en V

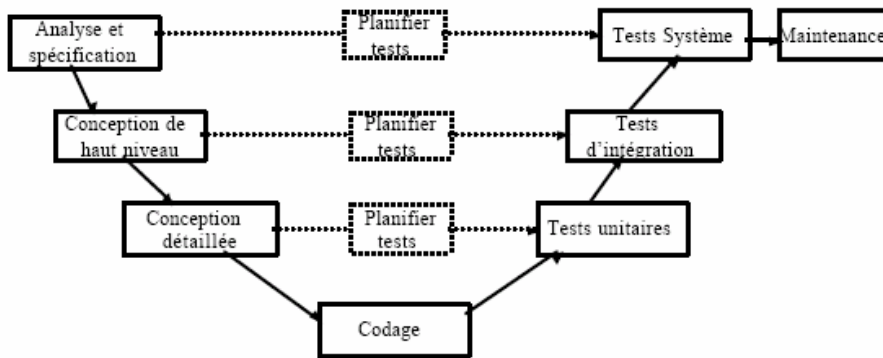


Figure 4

Le modèle en V du cycle de développement est inspiré du modèle en cascade. Il montre non seulement l'enchaînement des phases successives, mais aussi les relations logiques entre phases plus éloignées. L'application stricte du modèle en phases prescrit qu'on complète entièrement une phase avant de passer à la suivante. Dans la pratique, il arrive cependant qu'au cours d'une phase on découvre des erreurs commises dans une phase antérieure ou même l'absence d'éléments essentiels qui auraient dû être fournis par une phase antérieure. Il peut donc être nécessaire de revenir sur une phase précédente. Si tel est le cas, il faut parcourir à nouveau toutes les phases à partir de la phase révisée pour répercuter partout les modifications.

Il améliore du modèle en cascade, il met en évidence la symétrie et la relation qu'il y a entre les phases du début du cycle de vie et celles de fin. Les phases du début doivent être accompagnées d'une planification des phases de fin

On utilise ces modèles pour fabriquer le système coopératif, mais les modèles généraux ne peuvent pas donner les détails des phases. Pour constituer un système coopératif, les modèles ne sont pas assez compétence.

3. Les étapes recherche

Notre recherche veut résoudre les problèmes de l'application et l'infrastructure de système coopératif. Elle est illustrée en figure 5.

La première partie contient les étapes d'analyse des infrastructures et des applications. Ils incluent la décomposition la fonction du système coopératif en des fonctions individuelles. Adaptation des principes communs correspondus le design des fonctions, définition des scénarios qui doivent être supportés, adoption un modèle interactif supportant les besoins, réalisation user-interface concrète, développement des architectures physiques et des architecture logique. Puis on réalise des applications.

Les étapes suivant sont évaluation des infrastructures et des applications. Il inclut test si des designs correspondent aux les services devrait satisfaire.

Une application coopérative aide les usagers réalisent ses tâches coopératives, et une infrastructure exécute automatiquement certains parts de fonction des applications.

L'application et l'infrastructure sont développées sur les logiciels et les infrastructures communes comme une banque de données et un user-interface outil ou un système distribué.

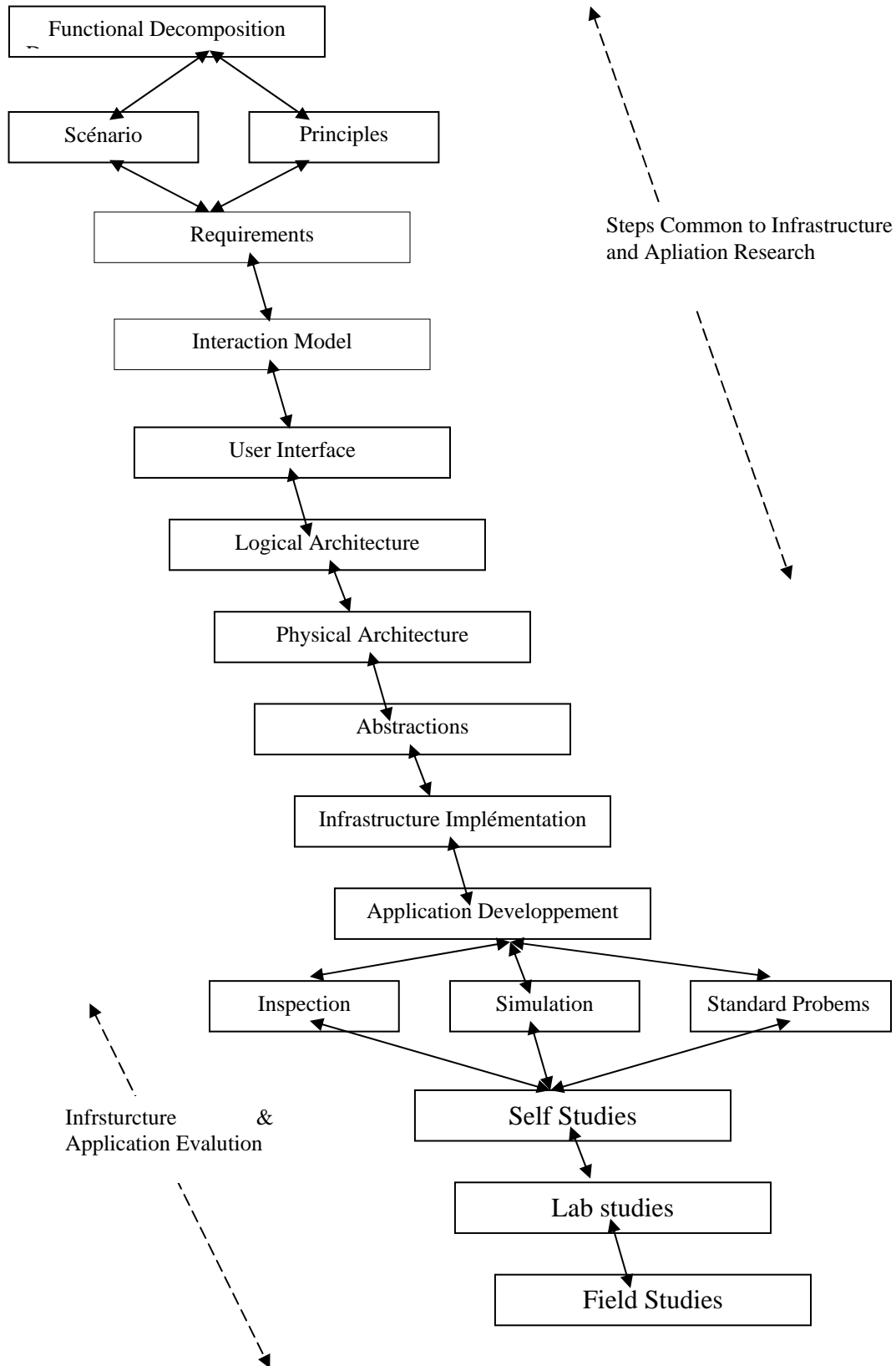


Figure 5. Diagramme of research plan

3.1 Décomposition des Fonctions

La *décomposition des fonctions* du système consiste à exprimer les besoins fonctionnels du système en des unités plus simples. Le résultat de cette activité est une structure appelée *architecture conceptuelle*. Les composants sont des abstractions qui traduisent les requis fonctionnels du système. Par exemple, un système opération est divisé en certaines subfonctions, gestion de processus, communication de processus, gestion du mémoire, et gestion du fils, etc. Chaque fonction est développée individuellement, mais il ne faut pas négliger les relations entre les subfonctions. Par exemple, Le design de gestion de processus dépende l'interface offerte par les composantes de mémoire.

Le système coopératif est plus compliqué que les logiciels traditionnels. A cause de il inclut des designs compliqués des multiple sections, comme le user-interface, la banque de données et système distribué.

Nous propose décomposer la fonction du système coopératif en des fonction dessous :
Interaction individuelle

- Couplage – effets d'action d'un utilisateur sur la présentation des autres
- Faire-défaire(undo)
- Trouver des différences dans les versions indépendantes d'un objet (diffing)
- Combiner les versions indépendantes dans un objet unique (merging)
- Contrôle d'accès : gestion des droits des acteurs (autorisations d'actions)
- Contrôle de concurrence : prévenir l'inconsistance d'actions effectuées simultanément
- Gestion de processus (workflow) du déroulement d'interventions et d'action
- Conscience de l'autre (awareness) : rendre compte des activités des autres pour partager une perception rendant compte de l'évolution globale du travail collectif
- Gestion de session : gestion des arrivées et des sorties des différents acteurs dans une session de travail coopératif.

Chaque subfonction que nous proposons n'est pas obligatoire pour tout les système, par exemple, un Internet-ChatRoom, il n'y a pas de subfonction de Faire-défaire.

Il y a des autres façons de la décomposition de fonction du système. Par exemple, D'après Olson, il divise la fonction en Linking, Awareness et Undo. D'après Ellis, il divise la fonction en Ontological, Coordination, User-interface. Ontological contient l'interaction individuelle, le contrôle d'accès, Coordination contient le contrôle de concurrence et la gestion de processus, User-interface contient l'interaction individuelle et couplage et conscience de l'autre.

3.2 Scénarios

Un scénario est un séquence de l'interaction entre l'utilisateur et l'application. On cherche les scénarios pour définir l'application et l'infrastructure.

Il y a des scénarios d'application-spécifique et des scénarios d'application-indépendante. Le scénario d'application-indépendante décrit les interactions communes et succinctes, et le scénario d'application-spécifique décrit les interactions spécifiques.

Par exemple, dans un système coopératif de l'hôpital, des docteurs font la conférence pour discuter les radios des patients, en cour de conférence, le système permet Alice et Luc faire un conversation privée, quant ils commencent le conversation, ils se échangent les radios. C'est un scénario spécifique, et il y a des autre scénarios spécifiques comme échanger des rapports, échanger des images. Un scénario indépendante peut décrit un ensemble de interactions similaires, échanger des document.

On peut classe des scénarios en scénario segmental et scénario complet. Un scénario segmental peut être inclure seulement une action de couplage. Et un scénario complet peut inclure les autre action comme défaire etc. Le scénario segmental aide designer la fonction indépendante, par exemple, le scénario couplage aide définir le modèle de couplage, le scénario accès aide définir le modèle d'accès. Des scénarios complets sont importants pour évaluer le système coopératif. Nous l'expliquons plus tard.

Comment définir des scénarios du système ? On peut analyser les interactions actuelles pour conclure des scénarios. Par exemple, dans la conférence de l'hôpital, on distribue les radios à chaque docteur. Dans la conférence virtuelle, radios sont déployés sur l'écran d'ordinateur de chaque docteur, les docteurs peuvent être dans un même lieu ou à distance. Quelque fois, si nécessaire on peut imaginer des scénarios hypothétique, après il est fabriqué, le test est très important pour assurer si le scénario hypothétique est se changé en scénario actuelle.

Il y a deux opinons du système coopératif : on considère la recherche du système coopératif est le problème de science de l'informatique, en contre, les autre personne considèrent la recherche doit concerner les chercheurs d'autre scientifique domaine comme psychologie, sociologie, etc. Selon nous, des chercheurs dans les autre domaine peut amener expériences valables accordant comment les gens faire les tâches coopératives dans la vie actuelle, il est très important pour le phase de scénario.

3.3 Les Principes

Les principes de système coopératif sont des règle correspondant le design et implémentation des fonctions.

Maintenant, il n y a pas de standardisation du principe qui est accepté par tout le monde. Chaque projet se établit son principe spécial.

3.4 Requirement

Un logiciel traditionnel corresponde trois requirement : l'efficacité, la flexibilité et ease-of-specification. L'efficacité exprime la consommation de la mémoire et le

temps de interaction. Il est surtout important pour les systèmes opérations et les systèmes de compilateur. La flexibilité exprime que les utilisateurs trouvent que les services du système sont ce qu'ils désirent. Pour l'application, la flexibilité exprime que l'application doit supporter tous les scénarios demandés par les utilisateurs finals. Pour l'infrastructure, la flexibilité indique qu'il faut support les programmeurs implémenter les applications nécessaires.

Hormis les trois requirement, il y a les autre requirement à considérer, notamment la réutilisation du Class de personnalité. Ca veut dire l'application de personnalité peut être partager par multiple personnel, mais sans besoin de la modifier.

3.5 Modèle de l'Interaction

Un modèle de l'interaction décrit un user-interface. On peut dire qu'un modèle est un user-interface abstrait, pour l'infrastructure, un modèle est une interface abstraite entre l'application et l'infrastructure.

3.6 User-Interface

L'étape suivant est implémenter le user-interface concret. Les programmeurs le réalisent par l'outil de user-interface.

3.7 L'architecture

L'architecture contient l'architecture logique et l'architecture physique.

Pour l'architecture logique, nous proposons diviser le système en des modules, des couches, des fils, des processus. Pour l'architecture physique, nous divisons le système en répliquas et host.

En général, l'architecture peut influencer la réutilisation, l'efficacité, et la complication de système coopératif. Par exemple, on peut décrire le système en un grand module, ou le séparer en un module qui charge implémenter la fonction d'application et un module qui charge le contrôle de utilisateur qui utilise l'application. Un module peut s'exécuter dans un host ou s'exécuter dans des hosts des clients (Replicating). Technologie de Replicating diminue le temps de réponse, mais il exige la synchronie.

3.8 L'Abstractions

L'abstraction décrit l'infrastructure pour des programmeurs. Il permet des programmeurs implémenter des modèles d'interaction et l'architecture sur l'infrastructure.

Effectivement, concepteur développe une abstraction d'un utilisateur, et puis l'améliore l'abstraction à une abstraction pour multiple utilisateur.

3.9 L'Implémentation de l'Infrastructure

Dans cette étape, l'infrastructure est développée. On peut le développer sur un infrastructure existé, par exemple, XTV(Abdel Wahab and Jeffay ,1994) utilise X Server .

3.10 Le développement de application

Dans cette étape, on établit l'application, si on utilise l'infrastructure, on considérer des paramètre offert par l'infrastructure, il faut déposer les paramètres défauts. L'application qui est développé dans cette étape est un prototype qui offre les fonctions correspondant les scénario et requirement.

3.11 L'évaluation

Une architecture n'est jamais intrinsèquement bonne ou mauvaise, les requis techniques, comme la portabilité et l'efficacité, ne suffisent pas à expliquer une architecture : le contexte culturel de l'entreprise, les délais imposés de production, et de manière générale, tous les intervenants, dans le processus de conception et de mise en œuvre d'un logiciel, ont une influence sur les structures architecturales retenues.

Dans cette étape, on examine commun le système marche, si la fonction marche bien, et on acquiers les information d'évaluation.

3.11.1 Inspection

Le plus simple façon de évaluation est inspection pour assurer si le système réalise complément les fonctions.

3.11.2 Simulations

Une autre façon est constatée si le système peut simuler les mêmes fonctions que un l'autre système. C'est pour assurer les deux système avoir la même flexibilité.

3.11.3 Standard problèmes

On peut utiliser le système exécuter le problème standard et on juge le système basé le solution que le système offert. Il est particulier propre quant il y a beaucoup de systèmes à comparer.

3.11.4 User Studies

L'expérience des utilisateurs peut examiner si il y a des scénarios désiré n'est pas établi, et si les scénarios hypothèque est satisfait les user final.

On collecte les données d'usage en trois étapes :

- Self studies : les développeur de système utilisent le système
- Lab studies : les gens hormis les développeur utilisent le système pour résoudre quelque problèmes représentable.
- Field Studies : les utilisateurs finals utilisent le système pour résoudre les problème concrets.

3.11.5 Programming data

- Self Prototyping : les développeurs de l'infrastructure développent prototype d'application démonstration.
- Self Implémentation : les développeurs de l'infrastructure développent Lab-application ,ça veut dire données de l'application sont collecté .
- Lab prototyping : Les autre programmeurs hormis les développeurs construisent le prototype d'application.
- Lab Implementation : Les autre programmeurs hormis les développeurs construisent le Lab-application.
- Field Implementation : les programmeurs final construisent les application concrète.

3.11.6 Performance data

La performance du système est difficile à mesurer, on peut demander directement les utilisateurs si ils trouvent le système est satisfaisant, ou évaluer le performance par benchmarks. Le flexibilité de système est évalué par l'inspection, la simulation, et le problème standard. L'efficacité de système est mesurée par données de la performance. Les scénarios hypothèque sont évalués par donnée usage.

4. La Conclusion et l'avenir

La recherche du système coopératif est de plus en plus important, selon nous, nous avons donné les opinons dessous:

- La recherche est décomposé en certain phase : décomposer la fonction de l'application en petites fonctions qui peuvent être rechercher plus où mois indépendamment. Adopter un principe général accordant le design de la fonction, identifier les scénarios collaboratifs en niveau multiple abstrait, identifier les requirement , réaliser le modèle d'interaction, développer l'architecture logique et l'architecture physique, développer l'infrastructure abstraite, réaliser les applications. Evaluer les applications et les infrastructures.
- Pour supporter les tâches coopératives, l'infrastructure et l'application doivent être designer ensemble.
- Les scénarios spécifiques et les scénarios indépendants doivent être définir.
- Les Requirements importants est dynamique, flexibilité, efficacité et réutilisation des logiciels existés. Il y a des collisions entre eux.
- L'inspection, la simulation, le problème standard peut aider les gens à évaluer le système coopératif.
- Expérimentation de l'usage concernent des programmeurs et des utilisateurs finals. Expérimentation de utilisateurs consistent Self Studies, Field Studies, self prototype. Expérimentation de utilisateurs consistent self implémentation, Lab prototype, Lab implémentation, et Field implémentation.

Plus part des phases sont optionnels. C'est un exemple de système qui suit les phases.

	Suit	Massive	Group Kit	XTV
Func.Decomp	Yes	No	Yes	No
Principe	Yes	No	Yes	No
Scénario	Yes	Yes	Yes	Yes
Requirement	Yes	Yes	Yes	Yes
Inter . Modèle	Yes	Yes	Yes	Yes
User Interface	Yes	Yes	Yes	Yes
Archi. logique	Yes	Yes	Yes	No
Archi.physique	Yes	Yes	Yes	Yes
Abstraction	Yes	Yes	Yes	Yes
Infra.Impl	Yes	Yes	Yes	Yes
Application	Yes	Yes	Yes	Yes
Inspection	Yes	Yes	Yes	Yes
Simulation	Yes	No	No	No
Standard Prob	Yes	No	Yes	No
Self Studies	Yes	Yes	Yes	Yes
Lab Studies	Yes	Yes	Yes	Yes
Field Studies	No	Yes	Yes	Yes

Pour conclusion, la recherche des approches de système coopératif est très importante, notre travail le débute, et nous espérons le meilleur d'avenir.

Article 2

Designing Work Oriented Infrastructures

OLE HANSETH

Department of informatics, University of Oslo, Norway

NINA LUNDBERG

Department of Informatics, University of Göteborg, Sweden

1. Introduction et motivation

La technologie de l'information concernant la médecine et la santé devient de plus en plus importante. L'introduction de cette dernière dans les hôpitaux a été lente et problématique grâce à divers problèmes.

L'objectif principal des auteurs est de nous montrer que dans certains cas, les infrastructures sont complexes donc utiliser les infrastructures classiques existantes n'est pas du tout efficace. Pour bien focaliser cette idée, les auteurs vont faire une analyse détaillée du fonctionnement d'un département de radiologie, ce qui va les aider à mieux comprendre ses infrastructures ainsi que les défis qui se posent devant l'introduction de l'IT dans les hôpitaux. En connaissant ces derniers, la création de nouvelles infrastructures qui entraînera à une bonne implémentation des PACS (Picture Archiving and Communication Systems) et des RIS (Radiological Information Systems) sera plus claire et plus facile à réaliser.

2. Département de radiologie

2.1 Définition

Le service de radiologie est une unité de service pour des départements cliniques à l'intérieur de l'hôpital, pour d'autres hôpitaux, et pour les unités primaires de soin (médecins généralistes). Les services fournis sont des examens radiologiques et des rapports basés sur le rayon X et d'autres types d'images radiologiques. Les services radiologiques sont les "outils" importants pour le diagnostic, le traitement, et l'intervention des patients.

2.2 Fonctionnement

De l'extérieur, le fonctionnement du département de radiologie paraît très simple (Pour commander un examen, un client envoie un ordre au département de radiologie. L'ordre identifie l'examen, spécifie la date et le médecin. Après l'examen, le médecin ajoute quelques lignes sur l'ordre, concernant le résultat de l'examen. Cet ordre revient enfin au clinicien pour être analysé). A l'intérieur du département, le fonctionnement est beaucoup plus compliqué. Quand un ordre arrive, le réceptionniste cherche dans le RIS pour voir si le malade a été examiné avant ou non afin de préparer les films anciens qui pourront être nécessaires pendant l'examen. Une date d'examen est spécifiée et l'ordre est placé comme tout les autres demandes dans un cahier de reliure placé sur les étagères en ordre selon les dates d'examen. Le jour de l'examen, le réceptionniste prépare les anciens films qui pourront être nécessaires pendant l'analyse des nouvelles images. Les nouvelles images sont stockées dans le PACS. Le rapport est placé sur une table située dans le corridor. En se déplaçant entre les différentes chambres, le radiologiste prend les rapports pour les examiner. S'il y a des images anciennes qui conviennent, alors ces dernières sont utilisées pour faire une

comparaison en utilisant un projecteur ou deux écrans d'ordinateurs pour plus de détail. Chaque jour, cinq ou six radiologistes sont assignés pour l'interprétation des images. Un rapport est imprimé et mis avec l'ordre hors de l'étagère pour être envoyé au clinicien par un service postal. En cas d'un long rapport, un dactylo l'imprime et le met dans l'étagère personnelle du radiologiste qui à son tour va le réviser, le signer et le mettre hors les étagères. Des photos sont parfois envoyées avec le rapport. Etc...

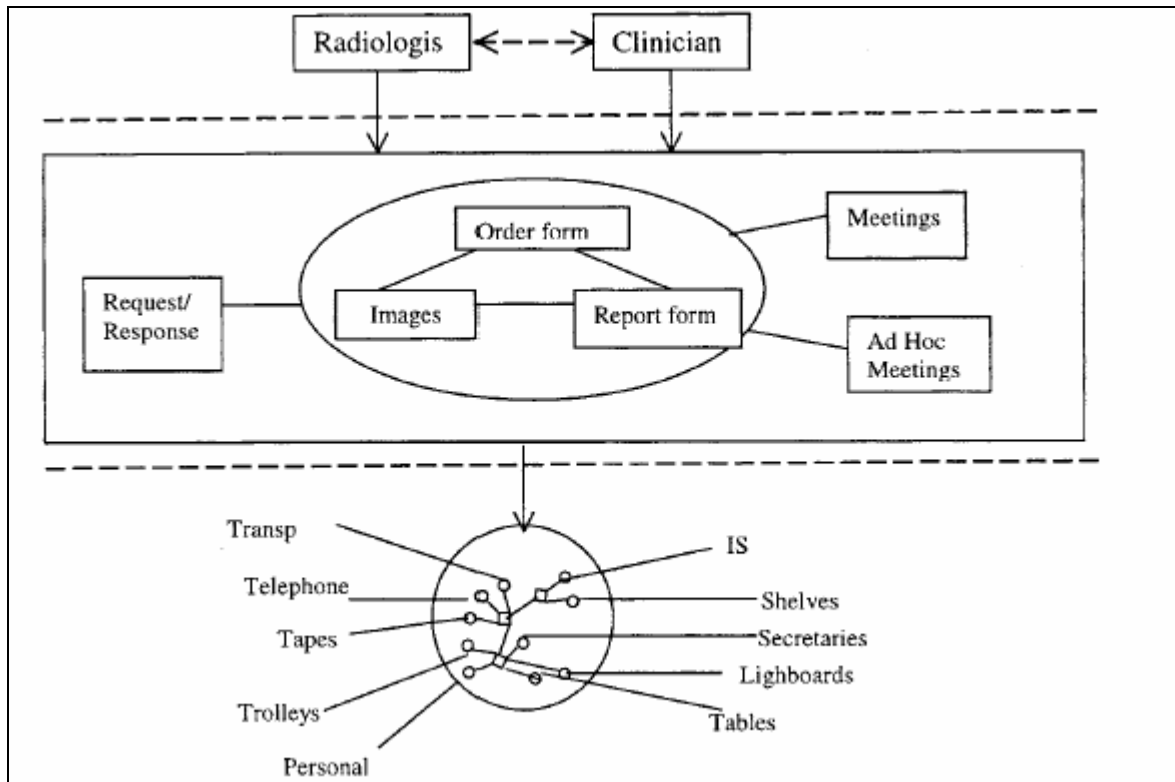
Donc plusieurs personnes, groupes et objets participent dans le fonctionnement à l'intérieur du département de radiologie et parfois une personne ou un objet joue plusieurs rôles. Le rôle initial des radiologistes par exemple est d'interpréter et analyser les images des malades pour faire un rapport de leur état. Alors que cela n'est qu'une de plusieurs tâches. Leur travail quotidien est encore divisé entre examens, réunions, patients, secteurs de réception, discussion avec d'autres médecins dans les couloirs, etc...

Le rôle d'une secrétaire est divisé entre organisation des dates, des rapports et des commandes, imprimer et envoyer les rapports, placer les ordres et les rapports dans les chariots et sur les étagères, préparer les disques médicaux pour une réunion, etc...

Les transporteurs qui ont pour rôle initial de transporter les malades d'un appartement à l'autre, en ce déplaçant entre les appartements, transportent parfois les rapports avec eux. Les rapports sont transportés parfois par les malades eux même, par leur parents ou par le service postal ordinaire. Même la table placée dans le couloir a plusieurs rôles dont l'un, est de donner au radiologiste une idée sur le travail qui vient tout simplement quand ce dernier jette un coup d'oeil sur les dossiers placer sur elle.

2.3 Infrastructure

Du point de vue des auteurs, plusieurs objets participant à l'infrastructure de la radiologie comme les rapports, les étagères, les commandes et les réunions, ne peuvent pas être considéré exactement comme infrastructure. Ces objets la ont le caractéristique d'une infrastructure.



La couche contenant les images, les rapports et les ordres n'a un rôle que lorsqu'elle est soutenue par une autre couche. Donc cette infrastructure joue le rôle d'une infrastructure de support. Un autre exemple est lorsque les radiologistes appellent les secrétaires durant leurs conversations avec les cliniciens pour demander des films de l'archive. Dans ce cas, les secrétaires, l'archive, les appareils de téléphone, etc. sont inclus dans l'infrastructure de support.

Quelques objets sont partagés entre plusieurs groupes. Par exemple, le rapport qui est un petit objet, joue un rôle crucial en tant qu'une infrastructure partagée. En fin chacun de ces objets ou groupe d'objets est conçu pour adapter les besoins d'autres groupes. Aucun objet n'est isolé, d'où l'ensemble des groupes et des objets constitue un réseau énorme qui à son rôle est formés d'un grand nombre de sous réseaux.

3. Problématique

Pour qu'un grand ensemble de réseaux fonctionnent ensemble d'une bonne manière il faut que leur fonctionnement soit basé sur des standards bien définis. Cela veut dire que l'implémentation d'une bonne infrastructure, exige une définition de standards. Le premier problème est qu'une infrastructure est constituée d'une infinité de réseaux chacun peut contenir des objets ainsi que d'êtres humains, d'où une autre difficulté est que les activités des êtres humains ne peuvent pas être conçues.

Dans un grand réseau, tous les sous réseaux communicants utilisent les mêmes standards donc le changement de ces derniers d'un seul coup est tout à fait difficile et presque impossible grâce à de grands problèmes qui auront lieu comme l'incompatibilité entre les systèmes existants et les nouveaux systèmes implémentés. Un tel changement prendra beaucoup de temps et il faut que chaque modification soit compatible avec les cas existants. Un bon exemple de ce phénomène est la nouvelle version de IP (IP V6) dont les études ont commencé il y a quelques années et prendront encore plusieurs années grâce aux problèmes d'incompatibilité avec la version ancienne.

4. Approche proposée

Le changement d'un grand réseau ne peut être réalisé qu'en considérant le réseau existant comme point de départ. Le changement peut seulement être effectué en remplaçant des petites parties du réseaux en de nouvelles parties qui fonctionnent sans problèmes avec le réseau entier. Cela demande parfois l'utilisation de passerelles. Par exemple, après que la technologie de PACS était mise en service, les cliniciens et les radiologistes ont voulu améliorer le système avec des fonctions permettant aux cliniciens d'accéder aux images de leurs postes aux départements cliniques. Comme la technologie de PACS fonctionnait sur des postes UNIX et non pas sur Windows, une passerelle (gateway) était développée pour transformer les images en format lisible par des navigateurs de Web. La première étape sera donc d'identifier le sous réseau qui doit être changer. Cela dépend du coût et de la complexité. Donc ça sera plus facile de commencer par les parties les plus simples.

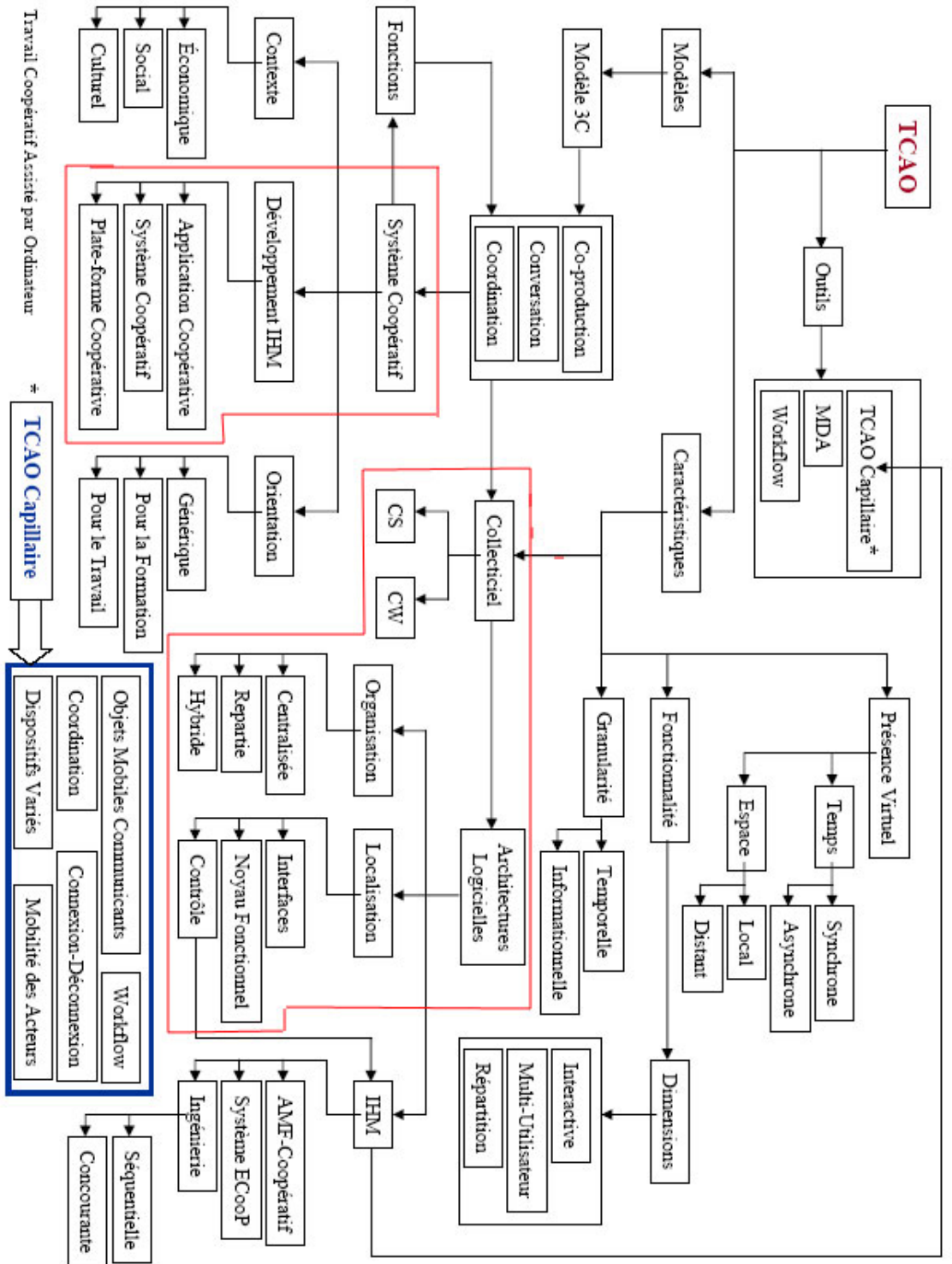
5. Infrastructure orientée par travail

L'infrastructure de la radiologie décrite dans cet article a quelques dispositifs qu'ils lui ont été attribués afin de soutenir des communautés spécifiques dans leur travail pratique. Dans ce cas-ci nous parlons des pratiques fortement complexes et spécialisées dont les propriétés sont en grande partie cachées pour ceux qui ne sont pas des membres de ces communautés (et qui également les membres sont inconscient environ). Une telle infrastructure est appelée « Work Oriented Infrastructure » ou infrastructure orientée par travail. Cette infrastructure est tout à fait différente de l'infrastructure classique qui est simple par son interface d'usage et présente ses services pour tout le monde dans une même égalité, par exemple l'infrastructure de l'électricité et de la télécommunication. Ces dernières doivent être conçues principalement par des ingénieurs tandis que l'infrastructure orientée par travail doit être conçue principalement par ses utilisateurs. L'infrastructure décrite dans cet article n'est pas tout simplement conçue par définition de quelques standards, mais elle est construite étape par étape durant une longue période. Par exemple, aucun ingénieur n'a proposé au radiologiste que les images doivent être déposées sur une table dans le couloir et que la hauteur puisse être utilisée comme protocole de communication et de coordination. Personne n'a proposée l'idée que les transporteurs peuvent transporter les rapports. Les ingénieurs peuvent concevoir des solutions, qui peuvent soutenir des pratiques existantes d'une meilleure façon, mais seulement jusqu'à un degré limité. Pendant plus de quinze années on a essayé d'établir des normes pour l'échange des gros morceaux d'information comme les ordres et les rapports de laboratoire (ordres radiologiques y compris et rapports). Les propositions qui ont été établies, ont été extrêmement complexes et en conséquence très chères de mettre en application et employer et très dur pour changer.

Conclusion

Le premier article présente plusieurs modèles du système coopératif et les étapes pour résoudre les divers problèmes de l'infrastructure. La fig 5 (Diagram of research plan) contient selon le premier article toutes les étapes nécessaires pour résoudre les problèmes d'infrastructure alors que dans le deuxième article, on voit un problème dans une infrastructure qui déjà existe mais qui plusieurs difficultés. Et d'après le premier article, quand l'infrastructure existe, alors on utilise le sens inverse du schéma qui est de bas en haut. Dans l'analyse du département de radiologie, on suit ce sens mais on ne passe pas par toutes les étapes. Les étapes utilisées sont :

- 1- Field Studies : A partir des analyses détaillée du département (40 heures de vidéo, 45 heures d'observation, 22 interviews, etc...)
- 2- User Interface : A partir du traitement des images en utilisant des ordinateurs.
- 3- Interaction Model : L'utilisation du rapport et des images entre les radiologistes, les médecins et les cliniciens.
- 4- Requirements : L'infrastructure de la radiologie a été changée plusieurs fois, et chaque version était modifiée selon les besoins.
- 5- Scénarios : À partir conversation, des téléphones et des réunions entre les medecins, les radiologistes et les cliniciens. Dans chacun des cas une ressource est partagée (Images, Rapport, etc...) entre les participant pour en discuter. Divers types de scénarios existe, soit entre deux personnes, soit entre des groupes de personnes.
- 6- Functional Decomposition : Chacun des gens participants au département de radiologie a sa fonction initiale. En plus il arrive parfois qu'une personne ait plusieurs roles.



Bibliographie

- 'A Framework to Integrate Synchronous and Asynchronous Collaboration', by S. F. Li and A. Hopper.
- 'Designing object-oriented synchronous groupware with COAST' by Christian chuckmann, Lutz Kirchner, Jan Schümmer, Jörg M. Haake
- 'Building Groupware with GroupKit' by Mark Roseman and Saul Greenberg
- 'Software Infrastructures' by PAUL DOURISH
- 'Architectures for Collaborative Applications' by PRASUN DEWAN
- Berg, M. (1999): Accumulating and Coordinating: Occasions for Information Technologies in Medical Work. *Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing*, vol. 8, no. 4, pp. 373–401.
- Berg, M. and C. Bowker (1997): The Multiple Bodies of the Medical Record. *The Sociological Quarterly*, vol. 38, pp. 511–535.
- Ciborra, C. (1996): Improvisation and Information Technology in organizations. In J.I. DeGross, S. Jarvenpaa, A. Srinivasan (eds.): *Proceedings of the Seventh International Conference on Information Systems* (December).
- Hanseth, O. and E. Monteiro (1996): Changing Irreversible Networks: Institutionalisation and Infrastructure. *Proceedings of the Information Systems Research Seminar in Scandinavia IRIS20*.
- www.cs.unc.edu/~dewan