



Chapitre 01

**Systèmes coopératifs
& Interopérabilité.**

1) Introduction :

L'évolution des nouvelles technologies de l'information et de la communication et entre autres l'Internet et les systèmes d'informations distribués a permis à un nombre important de cette communauté de partager des sources d'informations hétérogènes et d'une grande masse. La consultation, la mise à jour et la recherche d'informations par les différents utilisateurs nécessitent des outils et des moyens spécifiques se basant sur la standardisation de l'univers de discours. L'interopérabilité des systèmes est une meilleure solution qui propose un ensemble de processus afin de permettre aux différents systèmes de partager des informations et des services.

Aujourd'hui, l'échange de l'information est devenue plus qu'une nécessité vu le contexte économique dans lequel vivent les entreprises et vu le niveau de développement atteint par les technologies de communication. Cependant certains concepts tels que la coopération et l'interopérabilité méritent une définition plus précise.

Dans le cadre de ce chapitre nous présentons la notion de coopération et les systèmes d'information coopératifs ainsi que ses différentes propriétés, ainsi que l'interopérabilité des systèmes d'information et aussi les problèmes d'hétérogénéités qui en découlent. Nous présentons également les notions de base du Travail Coopératif Assisté par Ordinateur(TCAO), ensuite nous expliquons le principe de l'EAI (Entreprise Application Integration), et finalement les middlewares et les techniques d'intégration.

2) Coopération :

Les mutations technologiques de ces dernières décennies ont profondément bouleversé la vie des organisations. Progressivement, l'idée de la coopération dans les systèmes d'information est devenue une nécessité afin de répondre à des nouvelles exigences organisationnelles.

2.1 Définition de Coopération :

Il n'existe pas réellement de consensus sur une définition précise de la coopération. Plusieurs définitions souffrent d'une certaine proximité avec d'autres notions, comme celle de « collaboration » par exemple. Au travers des définitions présentes dans la littérature et relatives à la collaboration et la coopération, il est parfois difficile de différencier ces deux notions.

Nous retiendrons que la coopération est réalisée par la division du travail entre les participants comme une activité où chaque personne est responsable d'une partie de la résolution du problème. La collaboration consiste à l'engagement mutuel des participants dans un effort coordonné pour résoudre le problème ensemble.
[E. Boutigny]

Le terme de coopération peut se définir comme la capacité pour un ensemble de systèmes d'information autonomes, coexistant dans un environnement virtuel commun, à accomplir une tâche. [K.Zarour], Chaque système possède sa propre existence, indépendamment de la tâche globale à résoudre, dont l'accomplissement nécessite plus de ressources que n'en possède chaque système séparément, ou peut être réalisé plus efficacement par la collaboration de plusieurs systèmes.

La coopération est un processus par lequel des individus, des groupes ou des organisations entrent en relations et travaillent ensemble dans un but commun.
[E. Boutigny]

Le travail coopératif « émerge quand plusieurs acteurs engagés dans la réalisation d'une tâche commune, sont mutuellement dépendants dans leurs productions individuelles pour être en mesure d'effectuer la tâche qui leur a été confiée ».

La coopération se caractérise par son aspect pluridisciplinaire puisqu'elle implique non seulement des disciplines techniques telles que l'étude des interactions homme/machine, les réseaux de télécommunication, les théories de l'intelligence artificielle, l'étude des processus industriels, mais aussi des disciplines des sciences humaines telles que la sociologie des organisations, la sociologie scientifique, la psychologie cognitive, la psychologie sociale, l'ergonomie, l'anthropologie, les théories de la décision, les sciences cognitives, l'ethnologie, la linguistique, et l'éducation (M.brahimi 2010).

2.2 Objectifs de la coopération :

L'objectif premier de la coopération est de répondre à un besoin des entreprises. Les projets doivent correspondre à une perspective de bénéfice (financier, humain, matériel...), de limitation de risques ou de baisse des coûts.

Les objectifs de la coopération sont multiples. Nous citons certains d'entre eux :

- **Partage de l'information** : étude de marché commune, recherche de perspectives et d'axes de développement pour une filière, veille économique.
- **Partage de compétence et de moyens** : partage de compétences, de savoir-faire, de personnel (groupements d'employeurs), investissement en commun, groupement d'achat, recours groupée à un prestataire, gestion collective de domaines spécifiques...
- **Actions de formation** : définition en commun des besoins de qualification, partenariats avec les systèmes éducatifs, actions sur l'adéquation formation/métier...
- **Actions de recherche et d'innovation** : intelligence économique et territoriale, recherche et développement sur thématiques communes, partenariats avec acteurs locaux (centres de recherche, universités, laboratoires).

- **Actions commerciales** : promotion organisée en commun (site Internet, salons, publicité, labels), marketing du territoire, travail en commun à l'export, création d'une structure commerciale.
- **Partage des risques et des coûts.**

2.3 Contribution du travail coopératif :

Le travail coopératif a des retombées intéressantes sur les nouvelles organisations :

- **Meilleure rendement** : Le travail coopératif permettra une productivité meilleure et une nette amélioration des performances.
- **Meilleure qualité de services**: Le travail coopératif apportera une amélioration de la qualité du service, avec un meilleur temps de réponse. Une vision claire des tâches à effectuer, et des tâches effectuées par les autres.
- **Visualisation claire des tâches** : Le travail coopératif permettra de connaître en tout temps la situation exact des projets, les coûts, etc. et pour ainsi apporter une meilleure connaissance permettant d'améliorer considérablement l'efficacité.

L'informatisation a conduit au développement, dans les organisations, d'un nombre considérable de systèmes d'information chargés de stocker, d'organiser, d'interroger et de restituer des données utiles à l'utilisateur. Ces systèmes doivent collaborer afin de permettre une coopération à différents niveaux : coopération des informations issues des SI nouveaux ou existants, coopération entre les acteurs de l'organisation ou coopération entre les organisations. Les solutions répondant à ces besoins reposent sur la notion de système d'information coopératif.

3) systèmes d'information coopératifs :

Le domaine de l'informatique et plus particulièrement celui des systèmes d'information a connu une profonde mutation depuis une dizaine d'années. En particulier, les besoins d'ouverture et communication entre les systèmes d'information (SI) sont devenus importants, Les nouvelles applications telles que les systèmes d'aide à la décision, les systèmes d'aide au commandement, les applications

de simulation et le commerce électronique nécessitent une communication entre systèmes d'information. Les applications ont besoin d'accès coordonnées à différentes source de données afin de synthétiser les informations provenant de ces sources. Le partage de l'information présente des intérêts économiques non négligeables puisqu'il permet de réduire les couts d'acquisition et de mise à jour des bases de données, ce partage nécessite également une communication inter-systèmes. Par conséquent, les SI actuels doivent désormais être coopératifs mais leur répartition et leur hétérogénéité constituent des obstacles majeurs à leur coopération. [M.Saidane,2005].

3.1 Définition de système d'information coopératif .

Un Système d'information coopératif (SIC) se définit comme une collection de systèmes d'information distribués, éventuellement hétérogènes, ayant chacun une activité propre plus ou moins autonome, et travaillant de manière coordonnée à un objectif commun. Cet objectif peut être, à titre d'exemple, de donner un accès uniforme à des sources d'information distribuées et hétérogènes, ou de coordonner plusieurs processus afin de fournir un service ajouté. [A.El Fallah , B.Espinasse , A.Ouksel]

Les SICs sont généralement caractérisés par la grande variété et le grand nombre de sources d'informations. Ces sources d'informations sont hétérogènes, autonomes et distribuées soit sur un réseau local (Intranet) soit sur Internet. De tels systèmes doivent être capables d'exécuter principalement les tâches suivantes : [K.Zarour]

- la découverte des sources : trouver la bonne source de données pour l'interroger.
- la recherche d'informations : identifier les informations non structurées et semi structurées.
- le filtrage des informations : analyser les données et éliminer celles qui sont inutiles. Un système de filtrage de l'information peut assortir l'information qui est appropriée ou importante des énormes quantités de l'information et filtrer dehors l'information non pertinente et sans importance.

3.2 Propriétés des SICs :

Les systèmes d'informations coopératifs ont différents propriétés. Nous nous limiterons à trois propriétés essentielles : distribution, autonomie et hétérogénéité.

3.2.1 Distributions :

Il s'agit de la distribution physique des sources de données et à l'hétérogénéité des systèmes. Depuis déjà quelque année il est naturel que penser les données et les traitements ne soient pas physiquement sur un même lieu ou machine mais, au contraire, répartis sur un réseau. Des techniques comme CORBA de l'OMG, RMI/Java, Web services, etc. Permettent de prendre en charge cette distribution.

3.2.2 Autonomie :

L'autonomie fait référence au contrôle indépendant et séparé des systèmes composants. Ainsi, l'intégration ou l'abandon d'un composant ne doit pas perturber le fonctionnement global du SIC. En général, l'autonomie apparaît sous différentes formes. Autonomie définit trois niveaux :

1. *L'autonomie de conception*: indique que chaque système composant dans le SIC est libre de choisir sa propre conception. Cependant, il doit respecter certaines contraintes comme le modèle de données, le langage de requêtes, la représentation des données.
2. *L'autonomie de communication*: se réfère à la capacité d'un système composant de décider de la façon de communiquer avec les autres systèmes. Un système serait capable de décider quand et comment il pourrait répondre aux requêtes provenant des autres systèmes composants.
3. *L'autonomie d'exécution*: permet à un système d'exécuter ses opérations locales sans interférence avec les opérations soumises par d'autres systèmes composants. Ceci implique qu'un système ne peut pas imposer un ordre d'exécution à un autre système ayant ce type d'autonomie.

3.2.3 Hétérogénéité :

L'hétérogénéité peut se manifester sous plusieurs formes. Elle peut être provoquée par les différences technologiques au niveau des plates-formes matérielles et/ou logicielles (systèmes d'exploitation, réseau de communication, etc.). On peut distinguer également l'hétérogénéité induite par les différences des systèmes d'information composants. Ces différences se situent en général au niveau des modèles de données pour l'expression des structures et des contraintes et au niveau des langages de requêtes. Les différentes formes d'hétérogénéités engendrent plusieurs types de conflits lors de la coopération des systèmes d'information.

4) Interopérabilité des Systèmes d'informations:

De nos jours, Les systèmes d'informations des entreprises disposent d'une variété de données et de services qui évoluent sans cesse et couvrent pratiquement tous les domaines d'activité (Formation, Télécoms, banques, transport, Energie, travaux publics, etc.) Et l'ensemble des processus métiers (Comptabilité, Budget, Ressources Humaines, production, maintenance, Relation client, etc.). Le partage et la combinaison des connaissances disponibles sur différents systèmes constituent un besoin fondamental. De nombreuses entreprises développant de nouveaux produits sont amenées à développer toujours davantage de collaborations avec leurs clients, leurs partenaires et leurs fournisseurs dans le cadre de la chaîne logistique.

Les échanges entre partenaires sont donc devenus de plus en plus nombreux et de plus en plus complexes. De nombreux problèmes d'interopérabilité apparaissent alors et peuvent venir perturber les différents processus de collaboration entre partenaires. L'interopérabilité est considérée comme très importante voire critique dans de nombreux domaines, dont l'informatique, le médical au sens large, les activités ferroviaires, l'électrotechnique, l'aérospatiale, le domaine militaire et l'industrie en général. [G.Vicien et C.Merlo]

4.1 Définition De L'interopérabilité :

Le mot interopérabilité puisse revêtir des significations différentes en fonction du contexte, dans le domaine de l'informatique, il est généralement pris dans le sens de la capacité des produits et services informatiques disparates à échanger et à utiliser des données et des informations (c'est-à-dire "communiquer") en vue de fonctionner ensemble dans un environnement en réseau. [Livre blanc, 2005]

Diverses sources à l'échelle mondiale s'accordent sur cette définition :

- Un livre blanc sur l'interopérabilité daté de juin 2004 et rédigé par l'EICTA, une association d'échange informatique européenne, définit l'interopérabilité comme "la capacité de deux ou plusieurs réseaux, systèmes, périphériques, applications ou composants à échanger des informations et à utiliser les informations ainsi échangées".
- Le Digital Millennium Copyright Act (DMCA) de 1998 définit l'interopérabilité comme "la capacité des programmes informatiques à échanger des informations, et à utiliser mutuellement les informations qui ont été échangées".
- Le dictionnaire des télécommunications de Newton définit l'interopérabilité comme "la capacité à exploiter des logiciels et à échanger des informations dans un réseau hétérogène, c'est-à-dire un grand réseau constitué de plusieurs réseaux locaux différents".

Pour parler simplement, l'interopérabilité consiste à faire fonctionner ensemble des systèmes hétérogènes.

Une étude récente a montré que « 40 % des coûts d'un projet TIC dans la majeure partie des industries sont attribués à la résolution des problèmes d'interopérabilité »[R.Ruggaber, 2003]. Cependant, la nature de ces problèmes est bien trop souvent mal identifiée. En effet, l'interopérabilité est encore considérée aujourd'hui comme un problème informatique.

Il convient de distinguer « interopérabilité » et « compatibilité ». Pour être simple, on peut dire que la compatibilité est une notion verticale qui fait qu'un outil peut

fonctionner dans un environnement donné en respectant toutes les caractéristiques et l'interopérabilité est une notion transversale qui permet à divers outils de pouvoir communiquer—quand on sait pourquoi, et comment, ils peuvent fonctionner ensemble.

Actuellement, trois catégories de problèmes d'interopérabilité ont été identifiées. On distingue les problèmes conceptuels, les problèmes technologiques et les problèmes organisationnels.

- ***Les problèmes dits conceptuels.*** Ils concernent les aspects syntaxiques et sémantiques. Ils sont liés principalement à la modélisation des informations à un niveau d'abstraction relativement haut, comme par exemple un modèle d'entreprise, ainsi qu'à la façon de structurer les informations en vue d'un échange.
- ***Les problèmes dits technologiques.*** Ils concernent l'utilisation de matériels informatiques pour communiquer et échanger des informations. Ces problèmes sont relatifs, par exemple, à des incompatibilités au niveau des plates-formes, des infrastructures, du système d'exploitation, etc.
D'un point de vue purement technique, ces problèmes concernent les normes pour présenter, stocker, échanger, traiter et communiquer des données et des informations grâce à l'utilisation de systèmes logiciels.
- ***Les problèmes dits organisationnels.*** Ils sont liés aux structures organisationnelles et aux techniques de management utilisées dans des entreprises différentes. Plus précisément, ces problèmes concernent les aspects de responsabilité, d'autorisation, de confiance, légaux, de propriété intellectuelle, ...

L'interopérabilité amène deux problèmes majeurs, les conflits syntaxiques et les conflits sémantiques. Le premier type de conflit résulte de l'utilisation de modèles de données distincts entre systèmes. Par exemple, des modèles de représentation différents sont utilisés pour structurer un même concept (relation dans le modèle relationnel, classe dans le modèle objet, XML, etc). Le second type de conflit est issu des différences de compréhension et d'interprétation entre les informations provenant de divers domaines d'application. [Fabrice.J]

5) Travail Coopératifs Assisté par Ordinateur (TCAO) :

La notion de Travail Coopératif Assisté par Ordinateur est apparue vers le début des années 90 et est définie comme étant le domaine qui étudie la conception, la construction et l'utilisation des systèmes coopératifs. Connu aussi sous le sigle anglo-saxon de CSCW (Computer Supported for Cooperative Work), Il englobe l'aspect sciences humaines (travail coopératif) et technologique (les nouvelles technologies de l'information et de la communication NTIC). Il permet à un groupe d'individus de travailler ensemble via une infrastructure informatique en adaptant la technologie de l'information à leurs besoins.

Le travail coopératif suppose l'utilisation des TIC (technologies de l'information et de la communication) et la connaissance d'un certain nombre de notions techniques.

5.1 Workflow : Un workflow est un flux d'informations au sein d'une organisation, comme par exemple la transmission automatique de documents entre des personnes. On appelle « workflow » (traduisez littéralement « flux de travail ») la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier (aussi appelé processus opérationnel ou bien procédure d'entreprise). Le terme de « workflow » pourrait donc être traduit en français par « gestion électronique des processus métier ». De façon plus pratique, le workflow décrit le circuit de validation, les tâches à accomplir entre les différents acteurs d'un processus, les délais, les modes de validation, et fournit à chacun des acteurs les informations nécessaires pour la réalisation de sa tâche. Pour un processus de publication en ligne par exemple, il s'agit de la modélisation des tâches de l'ensemble de la chaîne éditoriale. Il permet généralement un suivi et identifie les acteurs en précisant leur rôle et la manière de le remplir au mieux.

5.2 Collecticiel (GROUPWARE).

Traduction du terme anglo-saxon GROUPWARE, nous retiendrons, en accord avec Favier (1998), que le collecticiel est « l'ensemble des techniques et des méthodes qui contribuent à la réalisation d'un objectif commun à plusieurs acteurs, séparés ou réunis par le temps et l'espace, à l'aide d'un dispositif interactif faisant appel à l'informatique, aux télécommunications et aux méthodes de conduite de groupe ».

5.3 Les espaces fonctionnels :

La coopération assistée par ordinateur, est décrite par Salber à travers un modèle conceptuel reposant sur la combinaison de trois espaces fonctionnels. [G.Gronier, C.Sagot]

Le schéma suivant présente les différents critères appartenant à chaque espace composant le système d'une plateforme TCAO en reprenant les trois sphères du trèfle d'Ellis. [m.leitzelman, h.dou, j.kister]



Figure .I.01: Les espaces fonctionnels.

- **L'espace de production (ou espace de données partagées)** : identifie tout objet résultant de l'espace d'activité du système collectif ainsi que les fonctionnalités de production d'objets partagés. La gestion des accès aux objets produits est assurée à ce niveau. Cet espace offre une vision statique du système.
- **L'espace de communication** : englobe toutes les fonctionnalités qui assurent l'échange d'informations entre les acteurs du collectif. Ce qui permet de définir la communication Homme/Homme médiatisée, il existe plusieurs types de communications : par exemple Les systèmes de visioconférence, la vidéo, la téléphonie sur IP, sont les modes de communication les plus utilisés.
- **L'espace de coordination** : représente tous les moyens de coordination dans l'espace de travail, qui se basent principalement sur les tâches et les activités, les acteurs et leurs habilités de modérer leur production.

L'affectation des tâches et des rôles aux différents acteurs dans l'espace collaboratif permet la coordination des différents acteurs pour réaliser une œuvre commune. Cet espace offre une vision dynamique du système.

6) EAI (Intégration des Application d'Entreprise) :

L'« integration business » est devenu un enjeu majeur pour beaucoup d'entreprises. Il y a de nombreuses raisons à ce surcroît d'intérêt : la croissance explosive d'Internet, les pressions de la compétition, de nouveaux modèles de business et le besoin d'homogénéiser les processus entre l'approche interne et celle avec les partenaires. Un nouveau secteur de marché a émergé pour répondre à ce besoin, dédié aux logiciels qui permettent une meilleure intégration du business. L'un de ces solutions est l'EAI. [M.chevassus,2004]

L'Intégration d'applications d'entreprise ou IAE (en anglais Enterprise Application Integration, EAI) est une architecture intergicielle permettant à des applications hétérogènes de gérer leurs échanges. On la place dans la catégorie des technologies informatiques d'intégration métier (Business Integration). Sa particularité est d'échanger les données en pseudo temps réel. Par extension, l'acronyme EAI désigne un système informatique permettant de réaliser cette architecture en implémentant les flux inter-applicatifs du système d'information. Ainsi l'EAI est la

création de solutions d'affaires robustes et élégantes en combinant les applications qui utilisent le middleware commune et d'autres technologies viables. Avec ces réalisations, l'EAI a été créée par les analystes de l'industrie pour aider les organisations informatiques à comprendre l'émergence d'un type de logiciel qui facilite les douleurs de l'intégration.

L'objet de l'EAI est l'interopérabilité et l'organisation de la circulation de l'information entre des applications hétérogènes, c'est-à-dire faire communiquer les différentes applications constituant le système d'information de l'entreprise, voire même celles des clients, des partenaires ou des fournisseurs.

6.1 Pourquoi EAI?

Dans le passé, les architectures des systèmes d'entreprise ont été mal planifiées. De nombreuses organisations construisent des systèmes basés sur la technologie fraîche de la journée sans se rendre compte comment ces systèmes seraient en quelque sorte, un jour de partager l'information. Il y a un certain nombre d'organisations équipées de différents types de systèmes ouverts et propriétaires. Chacun avec son développement, base de données propre, le réseautage et le système d'exploitation. Ainsi entraînant un environnement hétérogène. Pour surmonter ces maux qui affectent les entreprises, il y eut un certain nombre de technologies ad hoc et des méthodes. Mais rien ne semblait être une solution parfaite compte tenu de la complexité associée à ces systèmes. EAI est venu comme une aubaine pour les architectes d'entreprise de mettre tout à droite, qui font face aujourd'hui les entreprises. EAI est devenu un ensemble sophistiqué de procédures avec les technologies nouvellement raffinés, tels que des middlewares et des courtiers de messages qui permettent aux utilisateurs de lier les systèmes entre eux.

Comme les organisations commencent à comprendre la nécessité de l'interconnexion des systèmes disparates, afin de répondre aux besoins de l'entreprise, l'importance des technologies d'intégration est à la fois sentir. Ainsi, il devient un jalon très important pour chaque entreprise de manière efficace, l'architecte, la conception, un développement de systèmes basés sur la technologie EAI. Comme il y a un certain nombre de technologies EAI excellentes, il a besoin

d'une courbe d'apprentissage abrupte d'acquérir les connaissances et la sagesse afin de déterminer quand appliquer la technologie EAI, choisissant la technologie appropriée, l'architecte d'une solution, et ils l'appliquent avec succès au problème.

6.2 Types d'EAI?

Un système d'entreprise est composé de plusieurs processus d'affaires et des données. Alors, quand un expert en informatique envisage d'utiliser la technologie EAI, il doit d'abord comprendre comment ces processus d'affaires sont automatisés et l'importance de tous les processus d'affaires. Cette compréhension fera ressortir un grand nombre de conseils utiles pour déterminer la quantité de travail nécessaire, combien de temps cela prendra, quels processus d'affaires et les données doivent être intégrées, etc. En dehors de cette tâche initiale et la première de l'exploration, la connaissance primaire nécessaire est à ce niveau, le processus d'intégration doit être réalisé dans une application d'entreprise car il y a principalement quatre niveaux, tels que le niveau des données, au niveau interface de l'application, niveau de la méthode, et niveau de l'interface utilisateur dans une application.

1. Données au niveau EAI est le processus et les techniques et la technologie de transfert de données entre les magasins de données. Cela peut être décrit comme l'extraction d'informations d'une base de données, si besoin, traiter ces informations, et mise à jour le même dans une autre base de données. L'avantage des données de niveau EAI est le coût d'utilisation de cette approche. Parce qu'il ne peut y avoir de changements dans le code de l'application et, par conséquent il n'est pas nécessaire pour tester et déployer l'application résultant dans un peu de dépenses. De plus, les technologies fournissant des mécanismes pour déplacer des données entre bases de données, ainsi que reformate que les informations sont relativement peu coûteux compte tenu des niveaux d'autres EAI et de leurs technologies habilitantes applicables.

2. Demande interface de niveau EAI se réfère à la mobilisation des interfaces exposées par des applications personnalisées ou emballés. Développeurs faire usage de ces interfaces pour accéder aux processus d'affaires et de simples informations. L'utilisation de ces interfaces, les développeurs sont en mesure d'apporter de nombreuses applications ensemble, leur permettant de partager le logique métier et de l'information. Les seules limites que les développeurs sont confrontés sont les caractéristiques spécifiques et les fonctions des interfaces d'application.
3. Niveau de la méthode EAI est le partage du logique métier qui peut exister au sein de l'entreprise. Les applications peuvent accéder à des méthodes sur n'importe quelle autre application. Les mécanismes de partage des méthodes entre les applications sont nombreux, y compris les objets distribués, serveurs applicatifs. Un ORB peut prendre l'appel d'une application des méthodes stockées dans d'autres applications. Un serveur d'application peut être un serveur physique partagé pour un ensemble partagé de serveurs d'applications. La plupart de l'intégration ont été passe à ce niveau car il y a un certain nombre de technologies robustes pour accomplir ce type.
4. L'interface utilisateur au niveau EAI est une approche plus primitive. Les architectes et développeurs sont en mesure de regrouper des applications en utilisant leurs interfaces utilisateur comme point commun d'intégration. Par exemple, les applications mainframe qui n'ont pas de base de données ou des processus d'affaires au niveau d'accès peuvent être accessibles via l'interface utilisateur de l'application. Ce type n'est pas un préféré, même si à de nombreuses reprises, c'est la seule façon d'aborder la tâche d'intégration.

6.3 Technologies EAI?

La tâche de l'intégration peut être accomplie avec succès à différents niveaux d'une application. Ainsi, il existe différentes technologies compactes et élégant pour remplir ces objectifs. Il y a différents middlewares Object Request Broker, les courtiers de message, la technologie Web comme XML, etc. Il devient obligatoire pour un architecte d'avoir une solide compréhension de ces technologies, les mérites et les inconvénients de chaque technologie, où chaque brille et est flexible et évolutive avant lancer dans la tâche grandiose de l'intégration.

Il y a le point-à-point middleware, tels que les appels de procédure distante (RPC) et le middleware orienté message (MOM), base de données orientée middleware, et le middleware transactionnel, y compris les serveurs d'applications.

En outre, il y a un battage médiatique au sujet de la grande nouvellement couronné technologie Web XML qui peut faire des merveilles en tant que mécanisme d'intégration standard. XML fournit format commun d'échange de données, en encapsulant les deux métadonnées et de données. Ceci permet aux applications et bases de données pour échanger des informations sans avoir à comprendre quoi que ce soit les uns des autres. Le principal facteur en utilisant la technologie XML pour l'EAI, c'est que les données XML est portable. Enfin, le facteur de Java dans l'EAI est également de faire des vagues parmi les architectes EAI.

7) Les middlewares & les techniques (CORBA, DCOM, RMI) :

En architecture informatique, un middleware ou intergiciel est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques. Le réseau est mis en œuvre par l'utilisation d'une même technique d'échange d'informations dans toutes les applications impliquées à l'aide de composants logiciels.

Les composants logiciels du middleware assurent la communication entre les applications quels que soient les ordinateurs impliqués et quelles que soient les caractéristiques matérielles et logicielles des réseaux informatiques, des protocoles réseau, des systèmes d'exploitation impliqués. Les techniques les plus courantes d'échange d'informations sont l'échange de messages, l'appel de procédures à distance et la manipulation d'objets à distance. Les middlewares sont typiquement utilisés comme ciment pour relier des applications informatiques disparates des systèmes d'information des entreprises et des institutions.

7.1 Remote Méthode Invocation (RMI):

RMI (Remote Method Invocation) est une API Java permettant de manipuler des objets distants (c'est-à-dire un objet instancié sur une autre machine virtuelle, éventuellement sur une autre machine du réseau) de manière transparente pour l'utilisateur, c'est-à-dire de la même façon que si l'objet était sur la machine virtuelle (JVM) de la machine locale.

Ainsi un serveur permet à un client d'invoquer des méthodes à distance sur un objet qu'il instancie. Deux machines virtuelles sont donc nécessaires (une sur le serveur et une sur le client) et l'ensemble des communications se fait en Java. On dit généralement que RMI est une solution "tout Java".

7.1.1 Structure des couches RMI :

Les connexions et les transferts de données dans RMI sont effectués par Java sur TCP/IP grâce à un protocole propriétaire (JRMP, Java Remote Method Protocol) sur le port 1099.

A partir de Java 2 version 1.3, les communications entre client et serveur s'effectuent grâce au protocole RMI-IIOP (Internet Inter-Orb Protocol), un protocole normalisé par l'OMG (Object Management Group).

La transmission de données se fait à travers un système de couches, basées sur le modèle OSI afin de garantir une interopérabilité entre les programmes et les versions de Java.

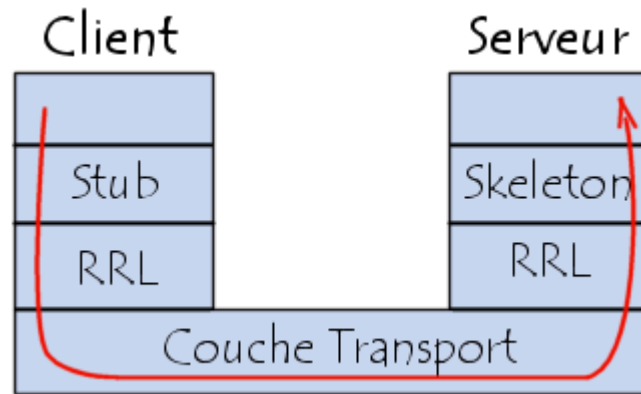


Figure .I.02 : Structure des couches RMI.

Le **stub** (traduisez souche) et le **skeleton** (traduisez squelette), respectivement sur le client et le serveur, assurent la conversion des communications avec l'objet distant.

La **couche de référence RRL** (remote Reference Layer) est chargée du système de localisation afin de fournir un moyen aux objets d'obtenir une référence à l'objet distant. Elle est assurée par le package `java.rmi.Naming`. On l'appelle généralement registre RMI car elle référence les objets.

La **couche de transport** permet d'écouter les appels entrants ainsi que d'établir les connexions et le transport des données sur le réseau par l'intermédiaire du protocole TCP. Les packages `java.net.Socket` et `java.net.SocketServer` assurent implicitement cette fonction.

7.1.2 Architecture de RMI :

L'architecture de RMI est schématisée ci-dessous :

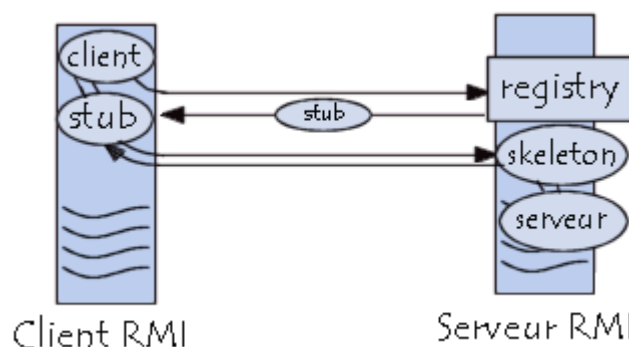


Figure .I.03 : Architecture de RMI.

Lorsqu'un objet instancié sur une machine cliente désire accéder à des méthodes d'un objet distant, il effectue les opérations suivantes :

1. il localise l'objet distant grâce à un service de désignation : le registre RMI.
2. il obtient dynamiquement une image virtuelle de l'objet distant (appelée stub ou souche en français). Le stub possède exactement la même interface que l'objet distant.

Le stub transforme l'appel de la méthode distante en une suite d'octets, c'est ce que l'on appelle la sérialisation, puis les transmet au serveur instanciant l'objet sous forme de flot de données. On dit que le stub "marshalise" les arguments de la méthode distante.

3. Le squelette instancié sur le serveur "désérialise" les données envoyées par le stub (on dit qu'il les "démarshalise"), puis appelle la méthode en local.
4. Le squelette récupère les données renvoyées par la méthode (type de base, objet ou exception) puis les marshalise.
5. le stub démarshalise les données provenant du squelette et les transmet à l'objet faisant l'appel de méthode à distance.

7.2 DCOM (Distributed Component Model) :

Distributed Component Object Model (DCOM) est une technique propriétaire de Microsoft qui permet la communication entre des composants logiciels distribués au sein d'un réseau informatique.

L'histoire a commencé en 1993 avec OLE 2.0 (Object Linking and Embedding), l'ensemble des mécanismes permettant aux utilisateurs d'un produit Microsoft d'intégrer des composants construits avec un autre produit Microsoft. Le fondement d'OLE était COM (Component Object Model), un standard de compatibilité binaire entre objets, et qui unifie l'architecture des différents services OLE. OLE n'a pas été créée à l'origine pour la programmation répartie. Depuis 1996 est apparu DCOM qui désigne les composants Windows qui peuvent être distribués sur différentes plates-formes (Windows) à travers Internet. Ces composants sont basés sur une extension du modèle objet COM aux environnements distribués, appelé DCOM.

L'usage initial des composants DCOM était la mise en œuvre de documents composés. Un document est composé de plusieurs composants répartis (tableaux, textes, graphiques).

L'intégration des concepts d'objet et de répartition, à la fois suivant une approche intégrée et appliquée. L'approche intégrée se matérialise par le fait que les objets sont des unités de répartition qui communiquent en utilisant des invocations à distance, et l'approche appliquée par le fait de développer des services de répartition sous forme de bibliothèques de classes. [C.Voisin]

7.2.1 Qu'est-ce qu'OLE ?

OLE est l'abréviation d'Object Linking and Embedding, ce qui signifie Intégration d'objets et Lien sur des objets. OLE est une technologie qui a été développée par Microsoft, initialement dans le but de permettre la programmation d'objets capables d'être insérés dans des applications réceptacles soit par intégration complète, soit par référence (ce que l'on appelle une liaison). Les objets intégrés ou liés sont capables de s'afficher dans l'application qui les contient. Ils sont également capables de fournir un certain nombre de services standards permettant leur manipulation (ces services peuvent être la sauvegarde de leur état, la capacité à être édité, etc...).

7.2.2 Qu'est-ce que COM ?

Component Object Model est une technique de composants logiciel (comme les DLL) créée par Microsoft. COM est utilisé en programmation pour permettre le dialogue entre programmes. Bien qu'il ait été mis en place sur de nombreuses plates-formes, il est toujours majoritairement utilisé sur Microsoft Windows. Son prédécesseur fut Object Linking and Embedding (OLE). COM est également sur le point d'être remplacé, en 2009, par le Framework .NET de Microsoft. Les technologies COM regroupent également COM+, Distributed COM et ActiveX.

7.2.3 Qu'est-ce que DCOM ?

Le modèle DCOM est un ensemble d'extensions à l'environnement COM et a été développé et livré après le système COM a été déployé dans la place du marché DCOM fournit des communications de composants distribués. L'un des objectifs majeurs de l'architecture DCOM est d'intercepter tous les messages locaux de communication interprocessus et les remplacer par un protocole de réseau basé sur la Distributed Computing Environment (DCE). En outre, le protocole pourrait également être responsable de l'acheminement de la demande dans un environnement réseau.

Les principaux avantages de DCOM tels que décrits dans Microsoft et Digital Equipment Corporation est la capacité de fournir un système de gestion de la mémoire partagée qui peut être utilisé entre les composants, la manière dont il interagit avec le réseau en ce qui concerne l'interopérabilité des réseaux de communication et de transparence, la capacité d'être en mesure de fournir le chargement dynamique et le déchargement des composants que nécessaire et en fournissant une erreur riche et de l'environnement de rapports d'état pour les développeurs.

7.2.4 Les services fournis par DCOM :

Du point de vue du programmeur, DCOM spécifie la manière dont les composants peuvent être utilisés. Du point de vue du système, DCOM spécifie les services que ce dernier doit fournir aux programmeurs.

Ce dernier aspect est moins intéressant (ou du moins il n'intéresse que Microsoft), parce que ces spécifications ne concernent que les éditeurs de logiciels qui désirent implémenter DCOM. On ne décrit donc DCOM que du point de vue des programmeurs.

DCOM regroupe les fonctionnalités des composants par interfaces. Une interface est un jeu de fonctions plus ou moins liées sémantiquement, qui permettent d'utiliser un composant. Un composant peut implémenter plusieurs interfaces.

DCOM spécifie la forme des interfaces au niveau binaire. Les interfaces sont simplement des classes abstraites pures. Cependant, comme cette structure peut être recréée dans d'autres langages. DCOM est donc indépendant du langage.

DCOM est sécurisé par les droits d'accès des utilisateurs. Ceci signifie qu'il est nécessaire de disposer d'un serveur de domaine pour donner ces droits (en pratique, ce serveur est un poste Windows Server). DCOM gère également les licences d'utilisation pour les composants commerciaux.

DCOM gère la durée de vie des objets OLE par compte de références sur ces objets. Les objets ne sont détruits que lorsque tous les clients de ces objets ont relâché leur référence qu'ils détenaient sur eux.

DCOM permet de stocker la description des interfaces dans ce qu'on appelle des types librairies. Ces bibliothèques stockent la description des interfaces, des fonctions des interfaces, de leurs paramètres et de leurs rôles. Elles permettent donc à un tout le monde d'utiliser un composant même sans en avoir la documentation.

DCOM définit les interfaces qui sont nécessaires pour l'utiliser. En particulier, il est possible d'utiliser les interfaces donnant accès aux fonctionnalités du marshaling, à la gestion de la mémoire de manière uniforme entre les objets OLE, à la gestion du multithreading, à la gestion des types librairies et à la connectivité des objets OLE entre eux.

CORBA est-il un middleware ? Bien sûr, CORBA est même un middleware "objet". Il permet à des systèmes hétérogènes de communiquer via un langage commun et un ensemble d'outils standardisés.

7.3 Qu'est-ce que CORBA?

L'architecture Common Object Request Broker (CORBA) est un standard développé par l'Object Management Group (OMG) pour assurer l'interopérabilité entre les objets distribués. CORBA est une solution dans le monde du middleware leader permettant l'échange d'informations, indépendante des plates-formes matérielles, les langages de programmation et systèmes d'exploitation. CORBA est essentiellement une spécification de conception pour un Object Request Broker

(ORB), où un ORB fournit le mécanisme nécessaire pour les objets distribués de communiquer avec un de l'autre, que ce soit localement ou sur des périphériques distants, écrites dans des langages différents, ou à des endroits différents sur un réseau.

OMG (92): CORBA est l'acronyme de Common Object Request Broker Architecture, ce fournisseur indépendant offre une architecture et une infrastructure permettant aux applications informatiques de travailler ensemble au travers du réseau. CORBA utilise le protocole standard IIOP qui permet l'interopérabilité des composants pouvant provenir de serveurs, de systèmes d'exploitation, de réseaux ou de langages de programmation similaires ou différents.

La première version de CORBA (version 1.1) date de 1992. CORBA a toujours été en pleine évolution pour prendre en compte de plus en plus de besoins d'utilisateurs. La version actuelle, CORBA 3.0, a été adoptée fin 2002 et éditée en 2004.

CORBA a été conçue pour développer des applications de la façon la plus simple possible, même si développer des applications distribuées n'est pas toujours facile. Transparence, portabilité et interopérabilité sont les principes de base de la spécification de CORBA. Quand le client fait une requête, il ne voit qu'un appel local, tous les mécanismes impliqués dans l'appel sont transparents au client. Toute machine, abritant un ORB (Object Request Broker), doit pouvoir exécuter des applications CORBA. Les applications doivent être portables sur des machines avec différentes architectures.

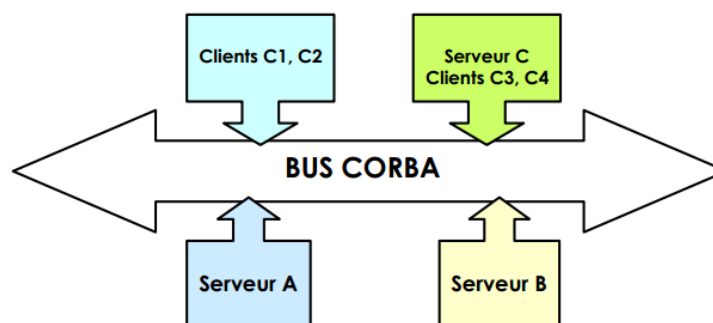


Figure .I.04 : Bus CORBA.

7.3.1 L'OMG :

L'Object Management Group (OMG) est un consortium international créé en 1989 et regroupant actuellement plus de 850 acteurs du monde informatique des constructeurs (IBM, Sun), des producteurs de logiciel (Netscape, Inprise ou ex-Borland/Visigenic, IONA Tech.), des utilisateurs (Boeing, Alcatel) et des institutionnels et universités (NASA, INRIA, LIFL). L'objectif de ce groupe est de faire émerger des standards pour l'intégration d'applications distribuées hétérogènes à partir des technologies orientées objet. Ainsi les concepts-clés mis en avant sont la réutilisabilité, l'interopérabilité et la portabilité de composants logiciels.

L'élément-clé de la vision de l'OMG est CORBA (Common Object Request Broker Architecture) un « middleware » orienté objet. Ce bus d'objets répartis offre un support d'exécution masquant les couches techniques d'un système réparti (système d'exploitation, processeur et réseau) et il prend en charge les communications entre les composants logiciels formant les applications réparties hétérogènes. [J.Geib, C.Gransart, P.Merle]

7.3.2 Object Request Broker (ORB) :

L'ORB fournit les mécanismes par lesquels des objets font des requêtes et reçoivent des réponses, et ce de manière transparente. Il fournit également l'interopérabilité entre des applications sur différentes machines dans des environnements distribués hétérogènes et il interconnecte sans coutures de multiples systèmes objets. D'une façon simplifiée, on peut définir l'ORB comme une entité qui fournit des mécanismes d'interrogations permettant de récupérer des objets, des procédures qui constituent une application.

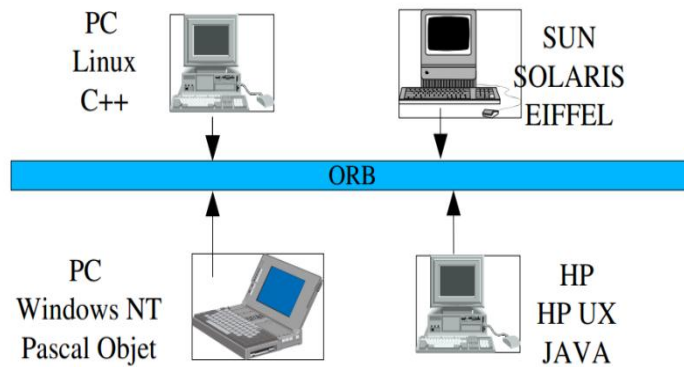


Figure .I.05: Bus Object Request Broker.

7.3.3 Protocoles inter ORBs :

CORBA propose un ensemble de spécifications, mais aucune technologie n'est imposée pour l'implémentation de l'ORB elle-même. Pour que différentes implémentations puissent interagir, un ensemble de règles de communication entre implémentations du bus sont définies, notamment :

- **GIOP**: Général Inter Orb Protocol fonctionne comme une surcouche de n'importe quel protocole de transport. GIOP contient plusieurs formats de messages pour supporter l'ensemble des sémantiques possibles des appels et des réponses des méthodes distantes. GIOP utilise un format de représentation des données universel et transportable à travers les réseaux.
- **IIOP** : Internet Intr Orb Protocol (instanciation de GIOP sur TCP/IP).
- **ESIOP** : protocoles pour les environnements spécifiques (DCE).

7.3.4 Interface Definition Language (IDL) :

IDL est un langage spécialisé pour la définition des interfaces, et c'est ce qui facilite la notion d'objets qui interagissent, si centrale à CORBA. L'IDL est le moyen par lequel les objets dire à leurs clients potentiels ce que les opérations sont disponibles et comment elles devraient être invoquée. La définition IDL définit les types d'objets, leurs attributs, les méthodes qu'ils exportent et les paramètres de la méthode. D'après les définitions IDL, il est possible de mapper des objets CORBA dans les langages de programmation ou des systèmes d'objets.

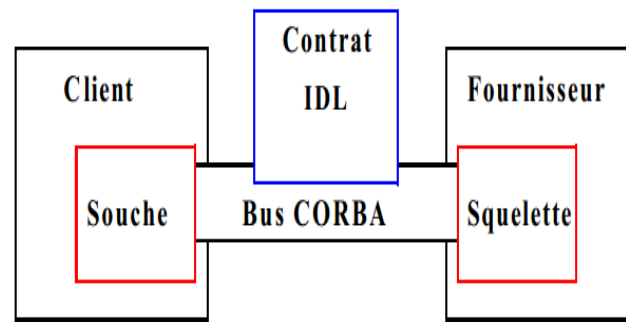


Figure .I.06 : contrat Interface Definition Language.

Le langage IDL (Interface Definition Language) permet d'exprimer, sous la forme de contrats IDL [Geib], la coopération entre les fournisseurs et les utilisateurs de services, en séparant l'interface de l'implantation des objets et en masquant les divers problèmes liés à l'interopérabilité, l'hétérogénéité et la localisation de ceux-ci. Un contrat IDL spécifie les types manipulés par un ensemble d'applications réparties, c'est-à-dire les types d'objets (ou interfaces IDL) et les types de données échangés entre les objets. Le contrat IDL isole ainsi les clients et fournisseurs de l'infrastructure logicielle et matérielle les mettant en relation à travers le bus CORBA.

Les contrats IDL sont projetés en souches IDL (ou interface d'invocations statiques SII) dans l'environnement de programmation du client et en squelettes IDL (ou interface de squelettes statiques SSI) dans l'environnement de programmation du fournisseur. Le client invoque localement les souches pour accéder aux objets. Les souches IDL construisent des requêtes, qui vont être transportées par le bus, puis délivrées par celui-ci aux squelettes IDL qui les délégueront aux objets. Ainsi le langage OMG-IDL est la clé de voûte du bus d'objets répartis CORBA. [J.Geib, C.Gransart, P.Merle]

7.3.5 Invocations statiques et dynamiques des méthodes :

La communication inter-objet se fait par invocation de méthodes d'objets éventuellement distants. Les invocations sont transparentes, il n'y a pas de différent visible entre invocations locales et distantes. Le langage dans lequel les objets ont été implémentés n'est pas apparent. Ces invocations peuvent être statiques ou dynamiques. [G.Linares]

- **Statiques** : l'invocation est spécifiée à la compilation, les informations de type, la nature des opérations sont spécifiées et vérifiées lors de la compilation. Les spécifications sont écrites en IDL, puis traduites par un compilateur IDL en interfaces dans un langage de programmation (par exemple c++, Java..).
- **Dynamiques** : les interfaces de communication sont spécifiées pendant l'exécution, par l'intermédiaire d'un référentiel des interfaces. Cet intermédiaire contient des métadonnées sur les interfaces d'objets, des descriptions de descripteur d'objets que sont les interfaces. De cette façon, toute application peut interroger le bus pour connaître la définition des objets.

7.4 Web services :

Les web services représentent un mécanisme de communication entre les applications distantes à travers le réseau internet ou intranet indépendant de tout langage de programmation et de toute plate-forme d'exécution :

- Utilisant le protocole HTTP comme moyen de transport. Ainsi, les communications s'effectuent sur un support universel, maîtrisé et généralement non filtré par les pare-feu.
- Employant une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées.
- Organisant les mécanismes d'appel et de réponse.

Grâce aux Web services, les applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés.

Dans le chapitre suivant nous allons présenterons les détails de web services et ses techniques.

8) Comparaison entre Les Normes : RMI, DCOM, CORBA, WEB SERVICE :

La comparaison entre les normes RMI, DCOM, CORBA, WEB SERVICE peut être synthèse dans le tableau suivant :

Caractéristiques	CORBA	DCOM	RMI	WEB SERVICES
Plates-formes	Multi	Windows principalement	Toutes les plates-formes JAVA	Multi
la liaison de données	IDL	MS IDL	Primitive, Les objets sérialisés	XML
Langage de définition d'interface	IDL	MS IDL	JAVA	WSDL
Protocole	IOP	RPC/DCE	JRMP	SOAP
Type de donnée	binaire	binaire	binaire	SOAP
Découverte de services	CORBA nommage	non	Rmi registry	UDDI
interopérabilité	Oui (via IOP)	non	non	Oui (via SOAP)
Fournisseurs	OMG	Microsoft	Sun	W3C
Approche objet	oui	oui	oui	oui

Tableau .I.1 : Comparaison entre CORBA, DCOM et RMI et Web Services.

9) Conclusion :

Ce chapitre a montré que le besoin d'échanger et de partager les informations est plus qu'une nécessité aujourd'hui puisque le marché exige des entreprises compétitives et réactives avec une grande capacité de communication. Comme le système d'information constitue l'élément central de l'entreprise pour communiquer en interne et en externe, une bonne coopération des entreprises passe par une coopération de leur système d'informations. Techniquement, l'interopérabilité est un moyen qui permettra de développer une nouvelle génération d'applications qui exploitent des sources de données implantées sur des systèmes géographiquement distants.

Dans ce qui suit nous présenterons la solution de Web services et ces techniques qui représente le cœur de notre travail.