

Module Système d'Information – Chapitre Informatisation du Système d'Information

TD : Infrastructures logicielles des Systèmes d'Information

L'arrière boutique (*back-office*) d'une entreprise peut se gérer avec un seul Progiciel de Gestion Intégrée (PGI, ou ERP en anglais) qui intègre la gestion de toutes les activités de l'entreprise dans un seul logiciel. Dans le cas contraire, le Système d'Information de l'entreprise sera une intégration de plusieurs progiciels de gestion, identifiés comme étant chacun le meilleur de sa catégorie (*best-of-bread*), pour gérer un besoin particulier de l'entreprise. Ce TD se concentre sur le second cas.

Exercice 1 : Les activités de l'entreprise

Question 1 : Rappelez les activités classiques de l'entreprise et les besoins de gestion associés.

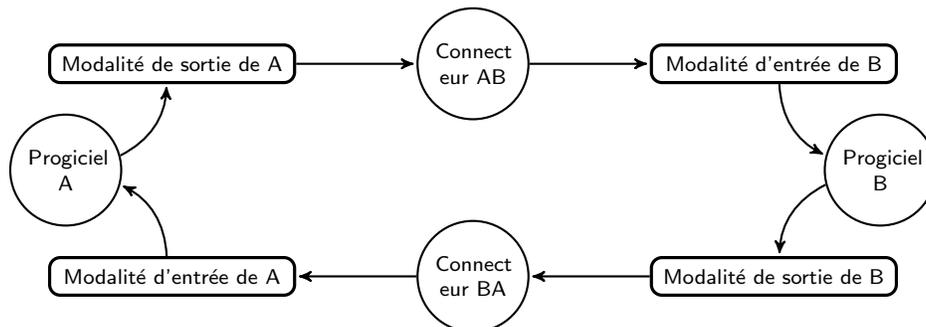
Question 2 : Pour les progiciels de la liste ci-après, associez le ou les besoin(s) de gestion qu'il permet de gérer parmi ceux évoqués précédemment. Attention de consacrer au plus 1 heure à ce travail, ensuite vous passerez aux exercices suivants.

Liste (non-exhaustive!) de logiciels de gestion pour l'entreprise :

- | | | | |
|----------------|---------------|--------------------|-----------------------------|
| 1. Ciel | 10. Eurecia | 19. Cogivea | |
| 2. Concurrive | 11. Windchill | 20. Acteos | 28. Adequasy Allegro |
| 3. Clipper | 12. Sage 100 | 21. Evoliz | 29. Creasoft Stocknet |
| 4. Lascom | 13. Audros | 22. vTiger | 30. Yourcegid Retail |
| 5. Prestashop | 14. Fletesia | 23. Magento | 31. ABW C-FIRST |
| 6. SugarCRM | 15. GLPI | 24. GCS TMS | 32. Apisoft Optimaint |
| 7. EBP | 16. Sellsy | 25. Divalto Idylis | 33. Siveco Group Cosswin |
| 8. Solune Alta | 17. Pentaho | 26. Qubes XXL | 34. Elfa-systèmes Clarrisse |
| 9. SpagoBI | 18. Akanea | 27. Hardis Reflex | 35. SolidWorks PDM |

Exercice 2 : Connecter deux logiciels

Le but est de permettre l'échange bidirectionnel de données entre deux progiciels qui ne sont pas prévus pour fonctionner ensemble. Pour ce faire, des programmes de conversion de données (fichiers, base de données, flux, ...) supplémentaires sont créés pour rendre ces échanges possibles. Ces programmes sont appelés des connecteurs. Ces petits programmes sont souvent des scripts BATCH, SHELL, BASH, Python, ...ou encore PHP, qui vont commencer par lire/extraire/recevoir (selon la modalité) une donnée provenant du logiciel source, puis l'analyser, la traiter, la convertir, et enfin écrire/charger/envoyer (selon la modalité) le résultat de la conversion à destination de l'autre logiciel.



Les principales modalités d'entrée et de sortie qui peuvent être couramment rencontrées sont :

- Lecture / Ecriture d'un fichier (csv, xml, ...)

- Extraction / Chargement dans une base de données
- Envoi / Réception d'un message via un webservice (XML-RPC, ...)

Le déclenchement d'un connecteur peut se faire :

- Ponctuellement et manuellement par l'utilisateur
- Ponctuellement et automatiquement par un programme (qui, par exemple, surveille l'apparition d'un évènement, procédure stockée (*trigger*), ...)
- Périodiquement et automatiquement, toutes les X secondes, chaque heure, chaque nuit, ... par le système d'exploitation (et le service de planification de tâches CRON, par exemple) ou par programmation interne dans le connecteur (qui boucle à l'infini).

Question 1 : En considérant uniquement les trois modalités ci-avant, quels sont les 9 cas possibles de connecteur AB entre deux logiciels ? Représentez-les schématiquement et Indiquez à chaque fois par un verbe le travail effectué par le connecteur sur la ressource intermédiaire entre lui et le logiciel.

Question 2 : En considérant uniquement les trois modalités ci-avant, combien de configurations sont possibles pour connecter deux applications A et B ?

Exercice 3 : Connexion de plusieurs applications en mode point-à-point

Nous allons étudier le nombre de connecteurs nécessaires pour que plusieurs applications puissent toutes communiquer entre elles. Pour cela, nous appelons la théorie des graphes pour modéliser le problème :

- L'infrastructure logicielle est représentée par un graphe orienté, où les sommets du graphe sont les logiciels de l'entreprise et les arêtes du graphe sont les connecteurs (permettant l'échange unidirectionnel d'information d'une application vers une autre).

Nous considérons que le graphe de l'infrastructure logicielle est un graphe complet (ou encore « *plat de spaghettis* »).

Rappel : un graphe complet est un graphe dont tous les sommets sont reliés entre eux.

Question 1 : Dessinez les graphes complets modélisant les cas de 2, 3 et 4 logiciels.

Question 2 : Combien d'arêtes comptent chacun de ces graphes ?

Question 3 : Redessinez le cas de 4 logiciels et ajoutez-y un cinquième logiciel. Combien de nouvelles arêtes doivent être ajoutées pour que le graphe reste un graphe complet ?

Question 4 : Déduisez (de la question précédente) la règle de calcul du nombre de nouveaux connecteurs à créer/installer/maintenir à chaque ajout ou changement de logiciel ?

Question 5 : Pour un Système d'Information de N logiciels, combien de connecteurs doivent être créés / installés / maintenus pour que chacun de ces logiciels puissent communiquer avec tous les autres ?

Exercice 4 : Enterprise Application Integration (EAI)

Dans une approche EAI (qui fait partie intégrante des démarches d'urbanisation des Systèmes d'Information), un logiciel supplémentaire sert de concentrateur (hub). Chaque logiciel communique alors avec les autres en passant obligatoirement et uniquement par le logiciel concentrateur. Ainsi, chaque logiciel doit être connecté au logiciel concentrateur. Nous parlons alors d'infrastructure logicielle « *en étoile* ».

Question 1 : Dessinez les graphes modélisant les cas de 2, 3 et 4 logiciels.

Question 2 : Combien d'arêtes comptent chacun de ces graphes ?

Question 3 : Redessinez le cas de 4 logiciels et ajoutez-y un cinquième logiciel. Combien de nouvelles arêtes

doivent être ajoutées ?

Question 4 : Déduisez (de la question précédente) la règle de calcul du nombre de nouveaux connecteurs à créer / installer / maintenir à chaque ajout ou changement de logiciel ?

Question 5 : Pour un Système d'Information de N logiciels, combien de connecteurs doivent être créés / installés / maintenus pour que chacun de ces logiciels puissent communiquer avec tous les autres ?

Question 6 : Une infrastructure logicielle EAI est un projet complexe à mettre en œuvre et demande un gros investissement de départ. Pensez-vous que ce type d'infrastructure logicielle puisse être mise en place dans de petits systèmes d'information avec peu de logiciels ?

Exercice 5 : Cas d'application

Consignes : Pour ce travail, associez-vous en binômes (au besoin, une seule équipe de 3 sera acceptée). Lors de la dernière séance de TD, plusieurs binômes seront sollicités au hasard pour venir présenter le résultat de leur travail au groupe et discuter ensemble les choix effectués.

Imaginez une entreprise de votre choix (création d'entreprise, entreprise que vous connaissez (stage, alternance) ou entreprise purement fictive). Listez les départements, activités et besoins de gestion que vous imaginez pour le fonctionnement boutique (*front-office*) et arrière-boutique (*back-office*) de cette entreprise. Identifiez les besoins logiciels : progiciels de gestion ou autres.

Trouvez les meilleurs progiciels de gestion (attention ERP interdits!!!) pour chaque activité et besoin de gestion. La notion de « meilleur » logiciel est à discuter en fonction de différents critères : depuis la couverture fonctionnelle pour répondre au besoin (développements supplémentaires nécessaires ?), jusqu'à l'interopérabilité (avec ses propres logiciels, mais aussi avec ceux des partenaires/clients), en passant pas les tarifs.

Pour chaque logiciel, créez et essayez de renseigner la fiche suivante :

	Logiciel 1	Logiciel 2	Logiciel 3	...
Nom du logiciel				
Domaine d'application				
Editeur				
Licence				
Langues				
Tarifs (peut s'exprimer en fonction du nombre d'utilisateurs, des options choisies, ...)				
Systèmes d'exploitation supportés ou hébergement mutualisé (Saas)				
Base de données utilisée (le cas échéant)				
Interfaces d'échange en sortie (formats d'export de fichier, webservices, ...)				
Interfaces d'échange en entrée (formats d'import de fichier, webservices, ...)				
Interopérabilité et compatibilité avec d'autres progiciels de gestion et logiciels d'EAI				

Enfin, vous devez trouver des solutions pour faire communiquer tous ces logiciels entre eux. Vous dresserez

une cartographie avec les différents logiciels et les connecteurs qui permettent les communications. Pour chaque connecteur, vous identifierez :

- Le type de connecteur (son entrée et sa sortie)
- S'il s'agit d'une compatibilité entre les deux logiciels, d'un nouveau développement de code à effectuer ou encore s'il s'agit d'une offre disponible dans le commerce.