

La discipline “systèmes”

État et évolution

Sacha Krakowiak
Université Joseph Fourier
Projet Sardes (INRIA et IMAG-LSR)
<http://sardes.inrialpes.fr/people/krakowia>

Systeme : un terme surchargé

■ Dans la vie courante

- ◆ système politique, système éducatif, système de défense, système solaire, système métrique, système nerveux, système philosophique, esprit de système, système D, “le système”, ...

■ En informatique

- ◆ système d'exploitation, système d'information, système réparti, système de gestion de bases de données, système temps réel, système critique, système embarqué, système de transitions, système logique, système de communication, système expert, micro-système, système de preuves, ingénieur système, ...

■ Quoi de commun ?

Ensemble composé d'éléments qui interagissent pour remplir une fonction dans un environnement donné

“Systèmes” en informatique : vers une définition

■ Tout ce qui est entre le “matériel” et les applications ?

- ◆ Première approximation ...
- ◆ ... mais trop large, inclut compilateurs, SGBD, IHM, ...

■ Idem, mais réduit à la logistique non spécialisée

- ◆ Gestion de ressources matérielles
 - ❖ mémoire, processeurs, disques, entrées-sorties, ...
- ◆ Fourniture de services génériques
 - ❖ communications, gestion de l'information, ...

Définition maintenant trop restreinte

“Systèmes” : définition par les fonctions

■ Transformation d'interface

- ◆ Fournir aux applications une “machine virtuelle” qui cache la complexité et les limitations du matériel sous-jacent

■ Gestion de ressources (physiques et logiques)

- ◆ Allouer, répartir, partager, protéger les ressources physiques et l'information

■ Services génériques et outils pour la construction de services

- ◆ Communication, nommage, liaison et composition, protection, ...

■ Garanties de “qualité de service”

- ◆ Performances, sécurité, tolérance aux fautes

Ces fonctions s'appliquent à différents niveaux d'abstraction et à différentes échelles (machine, organe spécialisé, grappe, grappe de grappes, réseau, ...)

Évolution des systèmes

Le domaine traditionnel du système d'exploitation s'est étendu sous deux influences

■ Le développement des réseaux et de l'informatique répartie

- ◆ Apparition d'un domaine nouveau : l'intergiciel (*middleware*)
 - ❖ Fournir des interfaces d'utilisation commode
 - ❖ Gérer la communication, la répartition, et l'hétérogénéité
- ◆ Le système d'exploitation n'est plus qu'un composant parmi d'autres

■ Le développement des systèmes embarqués et de l'informatique ubiquitaire

- ◆ Effacement de la distinction entre système et application
- ◆ Le système d'exploitation "déconstruit" est intégré à l'application

Systèmes : essai de taxonomie

Systèmes = infrastructures logicielles

■ De bas niveau

- ◆ Au contact du matériel
- ◆ Systèmes d'exploitation, noyaux spécialisés

■ De niveau intermédiaire

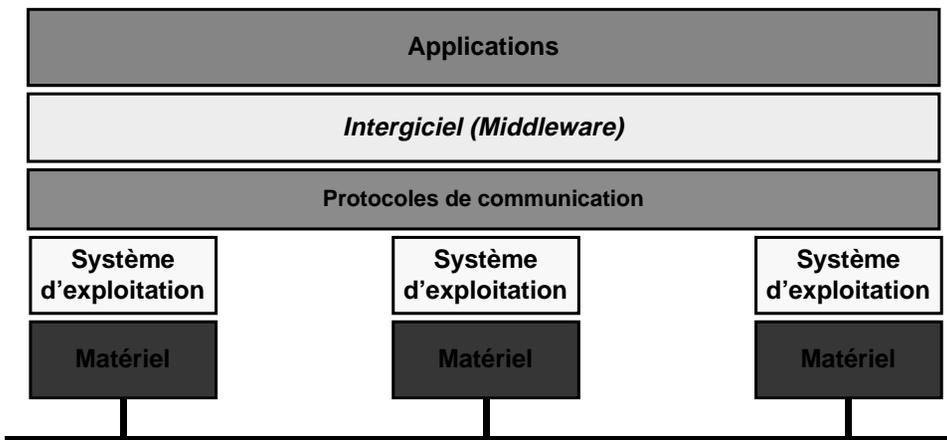
- ◆ Intergiciel (*middleware*)
- ◆ Au-dessus du système d'exploitation et des couches de communication
- ◆ Intégration "horizontale" d'applications (ou de systèmes)

■ Intégré à l'application

- ◆ Intégration "verticale"
- ◆ Systèmes embarqués



Évolution des systèmes (2)

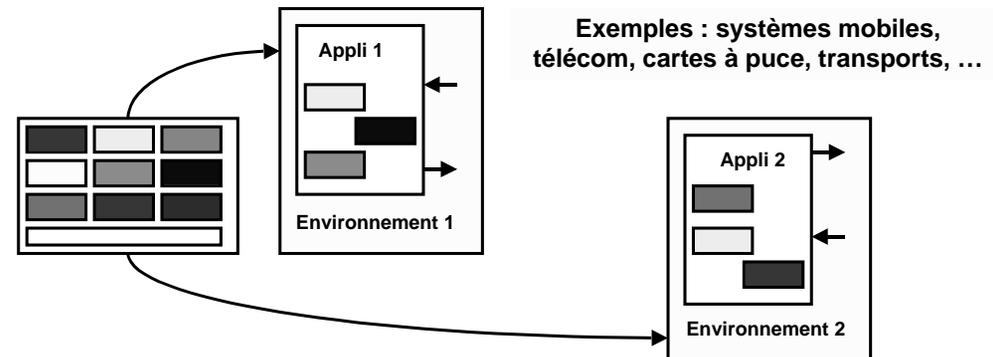


Exemples : *Enterprise Java Beans, Message Oriented Middleware, ...*

Évolution des systèmes (3)

■ "Déconstruction" du système d'exploitation

- ◆ Le système est décomposé en composants élémentaires
- ◆ Des systèmes ou applications sont construits "à la carte" en intégrant les composants nécessaires



La démarche en systèmes

■ Interaction

- ◆ Matériel-logiciel
- ◆ Système-applications-environnement
- ◆ Théorie-expérience
- ◆ Avec les domaines connexes (matériel, réseaux, langages, BD, GL, CHM)

■ Architecture et composition

- ◆ Vision globale, structure, principes d'organisation

■ Intérêt pour l'évaluation

- ◆ Quantitative (mesures, modélisation)
- ◆ Qualitative (collecte et formalisation de l'expérience acquise)
 - ❖ patrons & canevas

■ Évolution et adaptation

Systemes : les bases conceptuelles

■ Programmation concurrente et algorithmique parallèle

- ◆ Processus, synchronisation
- ◆ Parallélisation d'algorithmes

■ Gestion des ressources

- ◆ Ordonnancement de processeurs, gestion de mémoire, ...
- ◆ Évaluation de performances

■ Gestion de l'information

- ◆ Nommage, liaison, représentation

■ Algorithmique répartie

■ Architecture de systèmes et applications réparties

- ◆ Intergiciel, communication, composition

■ Tolérance aux fautes

■ Sécurité

Enseigner les "systèmes" : pourquoi?

■ Point de vue culturel

- ◆ Pour les infomaticiens non-spécialistes
 - ❖ Les concepts fondamentaux (parallélisme, gestion de l'information, architecture et composition) font partie du bagage de base
 - ❖ La maîtrise de l'intergiciel fait maintenant partie de la pratique de la programmation

■ Point de vue méthodologique

- ◆ La "démarche systèmes" est un outil très utile dans d'autres domaines techniques (architecture matérielle, réseaux, IHM, BD, ...)

■ Point de vue utilitaire

- ◆ Tout le monde utilise des systèmes, donc il faut
 - ❖ apprendre à s'en servir bien
 - ❖ comprendre (dans les grandes lignes) comment ça marche

■ Marché de l'emploi

- ◆ La demande reste forte pour les spécialistes (intergiciel, etc.)

Enseigner les "systèmes" : quoi, à qui ?

Principes

Savoir-faire pratique

L	Fonctionnement général (systèmes, appli. réparties)	Utilisation systèmes, construction élémentaire appli. réparties
M1 pour tous	Principes et tech. de base des systèmes. Programmation concurrente Principes programmation répartie	Pratique de la programmation concurrente, utilisation élémentaire d'intergiciels (RMI, CORBA)
M1 spécialisé	[en plus] Approfondissement systèmes (archi. interne). Intro. algo.répartie Modélisation, éval. de perf., intro. sécurité	Programmation de systèmes avancée (communication, etc.) Administration de systèmes
M2 professionnel	Architecture interne des intergiciels à objets et composants. Tolérance aux fautes, qualité de service. Sécurité	Pratique avancée des outils intergiciels (J2EE/EJB, .NET, MOM, Web Services, ...) Construction, configuration, adaptation d'outils et de systèmes pour domaines spécialisés (mobile, embarqué, etc.)
M2 recherche	Algorithmique parallèle et répartie, aspects fondamentaux de la tolérance aux fautes et de la sécurité. Archi. logicielle. Serveurs hautes perf..	[selon projet de recherche]

Enseigner les “systèmes” : comment ?

- **Apprendre en faisant**
 - ◆ des projets aussi ambitieux que possible, mais juste assez pour être faisables
 - ◆ expliciter les choix de conception et de réalisation
 - ◆ produire de la documentation, pas seulement du code
 - ◆ l'idéal : un système partiellement réalisé, à compléter
 - ◆ un bon support : le logiciel libre
 - **Ne pas oublier l'histoire**
 - ◆ pour identifier les principes et les paradigmes
 - ◆ pour analyser les facteurs de succès et d'échec
 - ◆ pour s'inspirer des “beaux ouvrages”
- P. Brinch Hansen.
Classic Operating Systems, Springer, 2000
- **Apprendre à tirer les leçons de l'expérience**

Enseigner les “systèmes” : quels moyens ?

- **Expérience des enseignants**
 - ◆ Importance de l'expérience concrète (professionnelle ou recherche)
 - ◆ Importance du travail en équipe
- **Matériel et supports**
 - ◆ Nécessité des systèmes ouverts (ex : Linux vs Windows)
 - ◆ Rôle essentiel du logiciel libre
- **Projets**
 - ◆ Facteur déterminant de la qualité de l'enseignement
 - ◆ Évaluation équitable du temps consacré
 - ❖ par les étudiants
 - ❖ par les enseignants

Systemes : quelques défis de la recherche

- **Passage à grande échelle**
 - ◆ Maîtriser les problèmes de performances
 - ◆ Garantir la qualité de service
 - ◆ Sur les réseaux (fixes ou mobiles), les grappes, etc.
- **Administration de systèmes**
 - ◆ Gestion, partage, protection des ressources
- **Systèmes et intergiciels adaptables**
 - ◆ Composants de base pour la construction de systèmes
 - ◆ Méthodes réflexives, systèmes auto-adaptables, ...
- **Aspects méthodologiques**
 - ◆ Mieux comprendre les aspects architecturaux : composition et décomposition, description, évolution
 - ◆ Assurer une meilleure liaison entre théorie et pratique
 - ❖ Ex : calculs de processus, tolérance aux fautes, sécurité, ...

La communauté des systèmes

- **Les grandes revues**
 - ◆ ACM Trans. on Computer Systems
 - ◆ IEEE Trans. on Computers (hard/soft), IEEE Trans. on Parallel and Distributed Systems, IEEE Distributed Systems (on line), Distributed Computing, ...
- **Les grandes conférences**
 - ◆ ACM SOSP, OSDI, Sigmetrics, ...
 - ◆ IEEE ICDCS, Middleware, DOA, ...
 - ◆ Usenix COOTS, USITS, ...
 - ◆ Clusters, ...
- **En France...**
 - ◆ ASF (Association SIGOPS France) 150 membres
 - ❖ Conférence CFSE, journées thématiques, Écoles
 - ◆ Communauté “grappes”
 - ◆ Communauté “temps réel”
 - ◆ ...

Conclusion

- Il existe une discipline “systèmes” en informatique
- Elle a profondément évolué dans les 10 dernières années (et continue d'évoluer)
- Elle met en jeu des connaissances et compétences spécifiques
 - ◆ fondements scientifiques
 - ◆ techniques et savoir-faire
 - ◆ démarche méthodologique
- Elle fait l'objet d'une recherche active
- Elle a des interactions fortes avec plusieurs autres disciplines
- Elle est un composant fondamental de la formation à tous les niveaux

Pour en savoir plus...

**École sur l'Intergiciel et la Construction
d'Applications Réparties (ICAR'2003)**

Autrans, 25-29 août 2003

<http://sardes.imag.fr/ecole/>