

**Un parcours d'enseignant-chercheur:
contributions aux langages de
programmation
et aux protocoles de communication**

Gregor v. Bochmann

École de science informatique et de génie électrique

Université d'Ottawa (Canada)

présentation donnée à l'IRISA, Rennes

le 14 mars, 2012

Résumé

- **Après un doctorat en physique théorique, j'ai commencé à travailler en informatique en 1971. Le choix d'un domaine de recherche n'était pas facile. Dans le domaine des langages de programmation, on parlait beaucoup des systèmes d'écriture de compilateurs. Les aspects syntaxiques étaient déjà bien étudiés – pendant quelques années, j'ai adapté les attributs sémantiques à la compilation en plusieurs phases pour formaliser les aspects sémantiques des définitions de langages. Puis, je cherchais un domaine de recherche moins exploré, et le trouvais en 1974 dans les protocoles de communication. C'était le temps des premières expériences avec des réseaux d'ordinateurs et le concept de protocole était encore très vague. Les travaux sur la spécification de protocoles, leur vérification, l'implantation et le test que j'ai fait avec des collègues de différents pays ont fourni les fondements pour le développement de protocoles de communication et leur normalisation. Depuis la fin des années 1990, j'ai encore cherché d'autres domaines d'activités avec des problèmes à résoudre, et j'ai travaillé sur la négociation de la qualité de service pour les applications multimédia, les systèmes pair-à-pair, et les réseaux optiques.**

Sommaire

- Perspective historique – les facteurs de développement
- Mon parcours personnel
- Défis de recherche
 - en langages de programmation
 - en protocoles de communication
 - en ingénierie de systèmes dirigée par modèles
- Thèmes de recherche récurrents
- Commentaires - enseignement, collaborations
- En guise de conclusion

Perspective historique de l'informatique

- **Les années 1950**
 - Premiers langages de haut niveau: Fortran, Cobol, Algol, Lisp
- **Les années 1960**
 - Théories des langages, compilateurs
 - Systèmes d'exploitation, partage de ressources, temps partagé
- **Les années 1970**
 - Réseaux d'ordinateurs - accès aux ordinateurs à distance
 - Langages pour la programmation de processus concurrents
- **Les années 1980**
 - Conception orientée objet
 - Systèmes multimedia - hypermedia (Videotex , annuaire téléphonique, etc.)
 - Intelligence artificielle (ordinateurs de 5ième génération au Japon, systèmes d'expert, etc.)
 - Normalisation OSI (Open System Interconnection)

Perspective historique de l'informatique (2)

- **Les années 1990**

- Utilisation de l'Internet à grande échelle
 - Service de courriel
 - Service de Web (nouvelle version de "Videotex")
- Normalisation de l'approche orientée objet (e.g. UML)
- Nouveaux protocoles de communication pour vidéo sur demande, téléphonie par Internet, commerce électronique (XML)

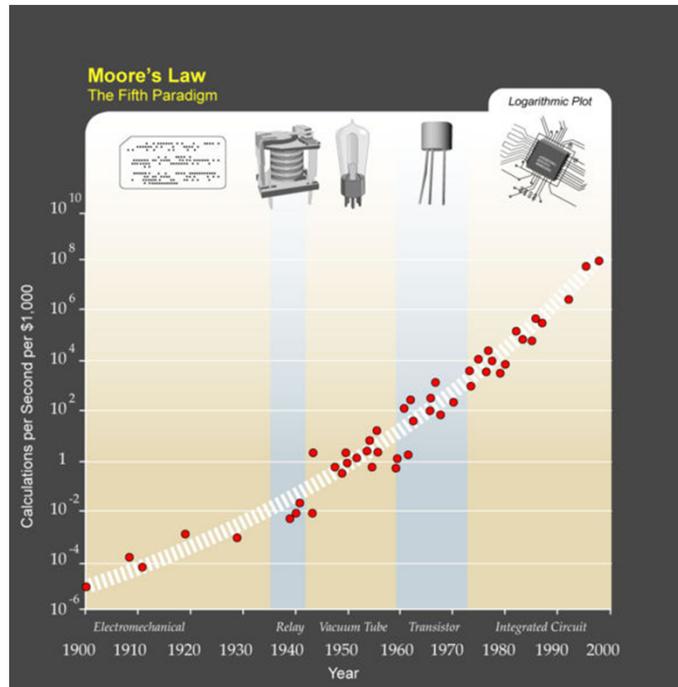
- **Les années 2000**

- Mobilité – réseaux sans fil
- Systèmes pair-à-pair – informatique virtuelle (cloud computing)
- Développement de logiciels "open source"

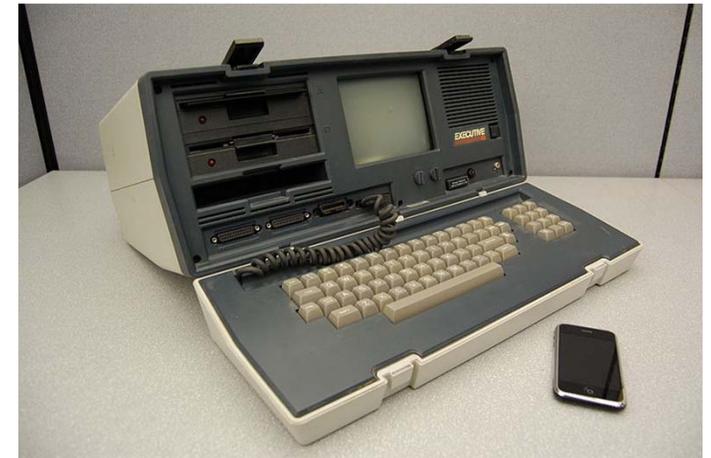
Les facteurs de développement

Avancées technologiques

- Le loi de Moore concernant la vitesse des ordinateurs



Calculs
par seconde
pour 1000\$



Osborne:
premier ordinateur portable
1982

- Développement similaire concernant le taux de transmission dans les réseaux

Les facteurs de développement (2)

Abstraction – interfaces

- Interface d'appels de procédures
 - e.g. orienté objet
 - aussi appels de procédures à distance
- Conception par composantes, réutilisation
- Architectures de systèmes en couches
 - systèmes d'exploitation
 - protocoles de communication

Virtualisation – normalisation

Interfaces fonctionnels normalisés, performance variable, flexibilité d'utilisation dans des contextes différents.

Exemples:

- Machine fictive - virtuelle , e.g. P-code pour Pascal [Wirth 1972], Byte-code pour Java [1995] – interprétée par des ordinateurs différents
- Mémoire virtuelle – pas réelle (performance dégradée)
- Connexion virtuelle – service de communication (e.g. appel virtuel avec X.25 [1976])
- Terminal virtuel (e.g. X.25 PAD, terminal Teletex, navigateur web)
- Émulation d'ordinateur – virtual machine (pas réelle, performance dégradée)
- Informatique virtuelle (cloud computing)

Virtualisation – normalisation (2)

Les forces de la normalisation:

- Organisations internationales (e.g. ISO, ITU-T, IET??) et nationales
- Organisations US avec participation internationale: e.g. IEEE, Internet Society
- Groupes industriels, e.g. OMG
- Normalisation ad hoc par domination du marché:
e.g. - le PC d'IBM - les protocoles d'Internet -
Windows de Microsoft - et le futur ??

Impact de la normalisation:

- facilite la réutilisation
- amène la compatibilité
- augmente la compétition entre manufacturiers
 - donc réduction des coûts et multiplication des applications

Mon parcours

Différents lieux de résidence

- Pendant et après la guerre, ma mère cherche un logement permanent au Schleswig-Holstein: Kiel – 1943 – Selent – 1946 – Möltenort – 1949 – Krummsee – 1955 – Eutin-Fissau – 1970

Différents lieux d'études universitaires: études en physique

- **Kiel** (1961-62) ; **Tübingen** (1962-63) ; **Lübeck** (une année d'études de piano et violoncelle, 1963-64); **Grenoble** (1964-65); une hépatite me fait perdre un an (1965-66); **Munich** (1966-67) ; projet de maîtrise au CERN (**Genève**, 1967-68); études au doctorat à l'Université McGill à **Montréal** (1969-71)

Différents domaines de recherche

- Projet de maîtrise: Physique expérimentale des particules élémentaires (programmation d'une PDP-8 pour l'analyse de données en temps réel)
- Projet de doctorat: simulations de modèles de réactions de particules élémentaires sur des noyaux (physique théorique avec beaucoup de calculs en langages Fortran, PL-1 et Formac)
- Projet de recherche post-doctorale en informatique à l'Université de Montréal (1971-72)
- Professeur en informatique à l'Université de Montréal (1972- 1997), à l'Université d'Ottawa (depuis 1998)

Mon parcours (2)

Domaines de recherche en informatique

- Choix de l'informatique: je cherchais un domaine avec moins de compétition professionnelle que la physique des particules élémentaires, et avec plus d'applications utiles pour l'humain.

Réseaux neuronaux (1971-72) avec Bill Armstrong

- Je trouvais des heuristiques, mais pas de lois fondamentales – et les résultats pratiques étaient mitigés

Systèmes d'écriture de compilateurs (1972 à environ 1976) avec Olivier Lecarme

- J'aimais les règles syntaxiques clairement spécifiées et les algorithmes d'analyse correspondants
- J'ai développé des algorithmes pour l'analyse de la sémantique d'un langage pour une évaluation en plusieurs phases
- J'ai regretté que les principes de base dans ce domaine avaient déjà été explorés

Communication entre ordinateurs et réseaux (à partir d'environ 1974)

- C'est le domaine dans lequel j'ai trouvé la plus grande reconnaissance (voir plus loin)
- Dans les années 1980, il y avait beaucoup d'intérêt industriel pour la normalisation OSI
- Dès le début, j'ai trouvé du financement pour mes travaux au Département des communications du Gouvernement du Canada, plus tard par les partenaires industriels du Centre de Recherche Informatique de Montréal (CRIM) et d'une chaire industrielle avec Hewlett-Packard

Mon parcours (3)

Autres domaines de recherche (à partir d'environ 1995)

- Négociation de la qualité de service (1995 – 2006)
 - pour l'accès aux banques de données multimédia et équilibrage de la charge des serveurs
 - pour applications interactives et mobiles (e.g. téléconférences)
- Contrôle de réseaux optiques et performance des protocoles
 - J'étais coordinateur de l'axe "architectures des réseaux" dans un Réseau de recherche "Agile All-Photonic Networks" impliquant 15 professeurs de 5 universités et 5 compagnies (2003 – 2008)
- Systèmes pair-à-pair (depuis 2005)
 - équilibrage de charge
 - distribution de video en temps réel
 - base de données avec consistance faible
- Exploration et modélisation d'applications Web-2 (en cours)
 - Projet en collaboration avec IBM

Défis de recherche en langages de programmation

Concepts des langages de programmation

- Appels de procédures et récursion (Algol 60, LISP)
- L'orienté objet et "garbage collection" (Simula 67, Smalltalk 80, "hype" les 90, maintenant e.g. Java)
- Structured programming (Dijkstra's paper "goto considered harmful")
 - Mon article "Multiple exits from a loop without the goto" (1973)
- Langages très abstraits (APL, SetI)
 - Mon intérêt pour les langages de modélisation, spécification

Techniques de description et de traitement

- Grammaires
- Analyse lexicale et syntaxique
- Systèmes d'écriture de compilateurs (automatisation)
 - mon premier projet de maîtrise supervisé : un SEC pour grammaires LL(1) – et encore aujourd'hui, j'enseigne ce sujet dans un cours
- Définition de la sémantique des langages

Techniques de description et de traitement (2)

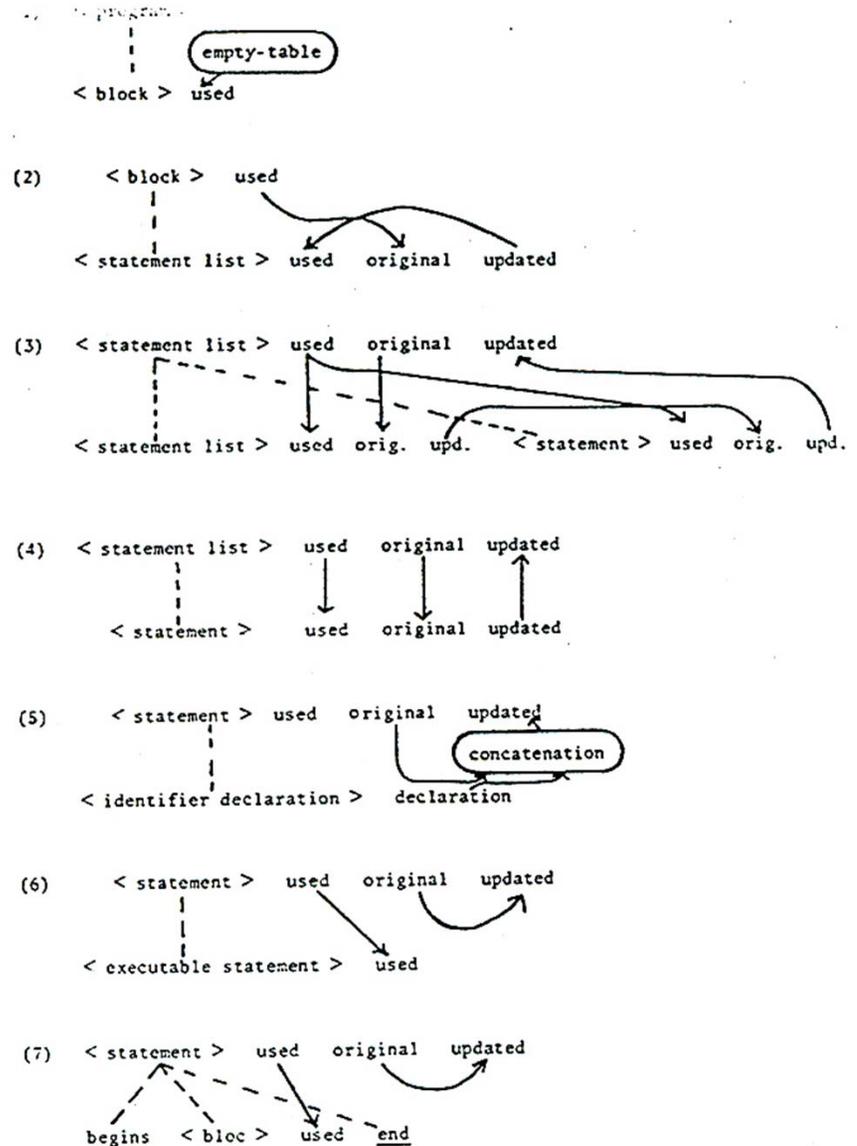
Définition de la sémantique par Attributs sémantiques

(proposés par Knuth en 1968)

- Dans quel ordre les attributs peuvent-ils être évalués sur l'arbre syntaxique d'un programme donné?
- Souvent, un compilateurs faisait plusieurs passes à travers le programme.
- Dans un article de 1976, j'expliquais sous quelles conditions les attributs peuvent être évalués en plusieurs passes de gauche à droite à travers le programme
 - Ces conditions ont été utilisées dans beaucoup de systèmes d'écriture de compilateurs qui ont été développés dans la suite
 - J'ai retrouvé ces conditions dans un livre récent (de Sebesta) que j'utilisais dans un cours sur les concepts des langages de programmation

Evaluation d'attributs sémantiques: un exemple

- Les règles d'évaluation
une règle d'évaluation pour
chaque règle de production
syntaxique, tirées de mon
article de 1976



Défis de recherche en protocoles de communication

Les réseaux d'ordinateurs – dans les années 1970

Premiers réseaux expérimentaux

ARPANET (USA): réseau longue distance – 1969

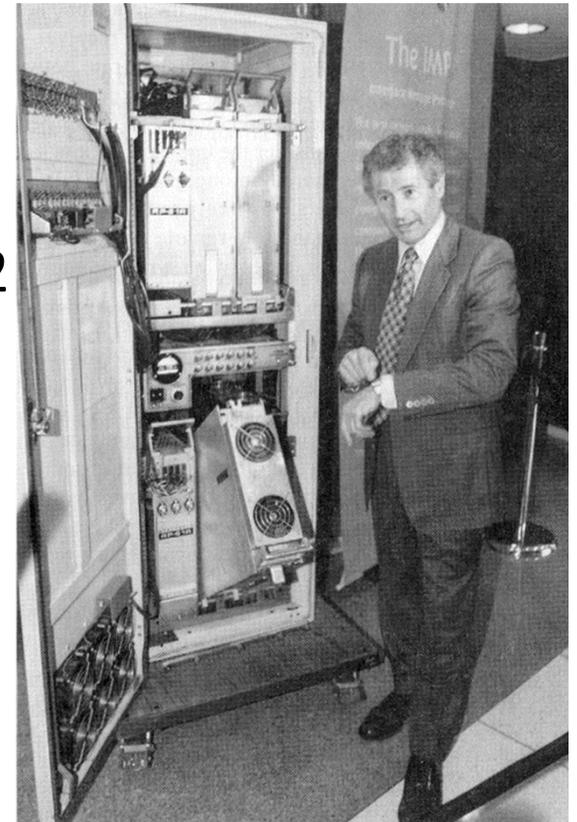
NPL network (UK): premier LAN

Cyclade (France): réseau longue distance – 1972
- introduit un protocole de type IP

Donald
Davies,
NPL



Louis Pouzin
INRIA (France)



Leonard Kleinrock, UCLA
with ARPANet node

Développement de protocoles – années '70 – '80

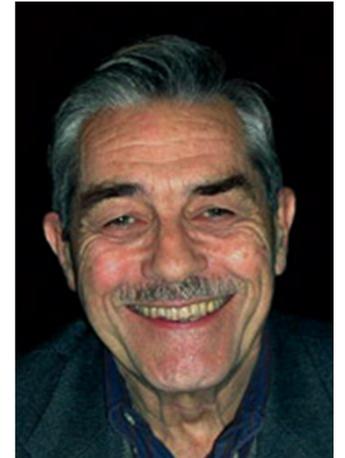
- Protocoles normalisés
 - X.25 : premiers protocoles normalisés pour réseaux d'ordinateurs
- Architectures de protocoles des vendeurs
 - IBM (SNA), DEC, Honeywell, etc.
- Protocoles d'applications
 - Internet: e.g. FTP et SMTP (années 1970)
 - Videotex – première version du toile (en France: terminal d'annuaire de téléphone – environ 1980)
 - ASN.1 et OSI “Remote Operations” (environ 1984) : similaire au “Web Services” d'aujourd'hui

Mon histoire personnelle



J'ai rencontré Louis Pouzin à une conférence en 1973

J'ai analysé le protocole ABP en 1974 et j'ai développé l'analyse d'accessibilité pour vérifier des protocoles modélisés par des automates – article en 1975



- J'ai appliqué la même méthode à la vérification du protocole X.25 (article en 1978)
- J'ai aussi expérimenté avec la preuve de programmes pour vérifier des protocoles (article en 1975)

Mon histoire personnelle (2)

- En 1977, avec Jan Gecsei, j'ai proposé la modélisation avec des automates étendus

En présentant l'article au Congrès IFIP à Toronto, je rencontre Zafiropulo du laboratoire IBM de Zurich qui travaille avec Colin West et Harry Rudin sur la vérification de protocoles.



- J'ai travaillé avec Carl Sunshine sur la formalisation de:
 - **Protocole**: pas défini comme une interface entre deux entités à distance, mais défini par des exigences sur le comportement de chaque entité
 - **Service**: une abstraction d'une couche d'un système distribué qui contient plusieurs entités de protocoles

C'est quoi, un service de communication

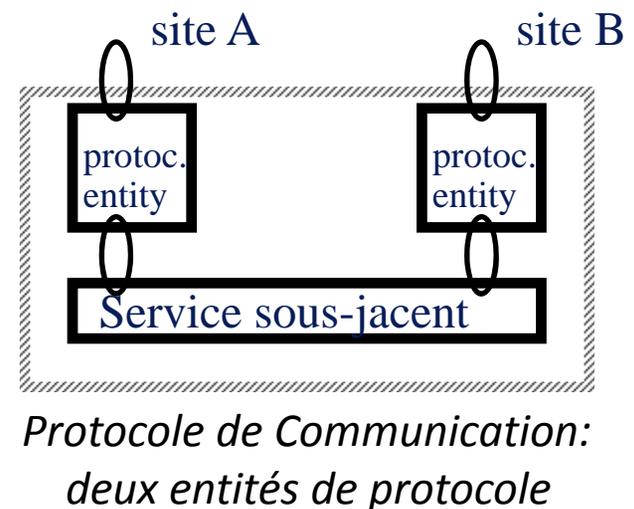
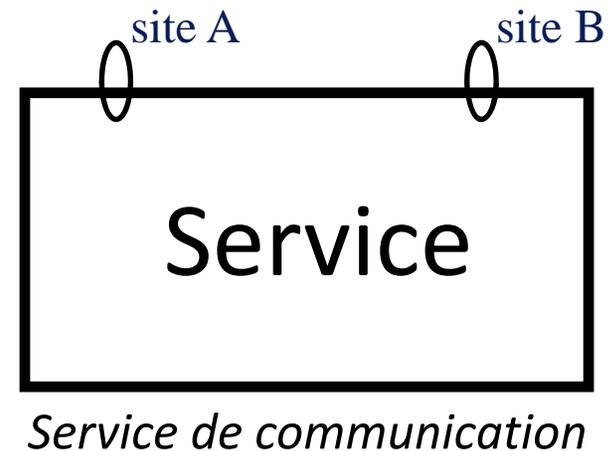
– un protocole de communication ?

Comment spécifier un service ?

- *C'est une composante de système répartie sur différents sites*
- **Propriétés locales** pour chaque interface, en plus des **propriétés globales**
- **Interactions abstraites** sur les interfaces dans les **deux directions** (en contraste avec un interface d'un « objet »)
- NOTE: Généralisation avec Michel Raynal pour composantes quelconque

Comment spécifier un protocole ?

- Un modèle **abstrait** du comportement d'une entité avec **deux interfaces**
- Échange de messages – définition de **l'encodage en détail**



Défis de recherche en ingénierie de systèmes dirigée par modèles

- Ce domaine a été développé par les chercheurs travaillant sur les protocoles de communication déjà dans les années 1980 et avant.
- Dans les années 1990, ce domaine est devenu populaire avec le développement de la notation UML – *plus que 10 ans plus tard*

Une comparaison

Ingénierie de protocoles

Ingénierie dirigée par modèles

<ul style="list-style-type: none">• Spécification du protocole	<ul style="list-style-type: none">• Modèle du système
<ul style="list-style-type: none">• Vérification du protocole• Développer une implantation• Test de conformité d'une implantation (tests basés sur la spécification – "black-box testing")	<ul style="list-style-type: none">• Le "model checking"• Transformation de modèles• Le "model-based testing"

Une comparaison (2)

Langages de spécification

- outils (automatisation) pour

- la vérification,
- l'implantation et
- le test

Ingénierie de protocoles

Ingénierie dirigée par modèles

- Techniques de description formelles pour protocoles et services OSI (années 1980)
 - Estelle (e.g. outil de l'INT)
 - SDL (outils: Geode de Vérilog, Telelogic-IBM)

- UML v1 (1996)
- UML v2 – SDL (2005)

Quelques résultats de nos recherches

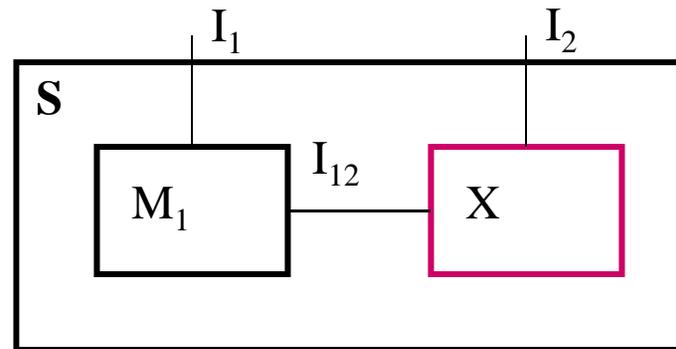
- Analyse d'accessibilité
 - l'étape préalable au "model checking" des systèmes distribués
- Génération de code à partir de spécifications de modèles (langages Estelle, ASN.1)
- Génération de suites de tests à partir de spécifications de modèles
 - De 1989 à 1997 dans le cadre de la chaire industrielle à l'Université de Montréal
- Mondel: langage de spécification orienté objet

Questions de recherche récurrentes:

Dérivation de comportement de sous-modules

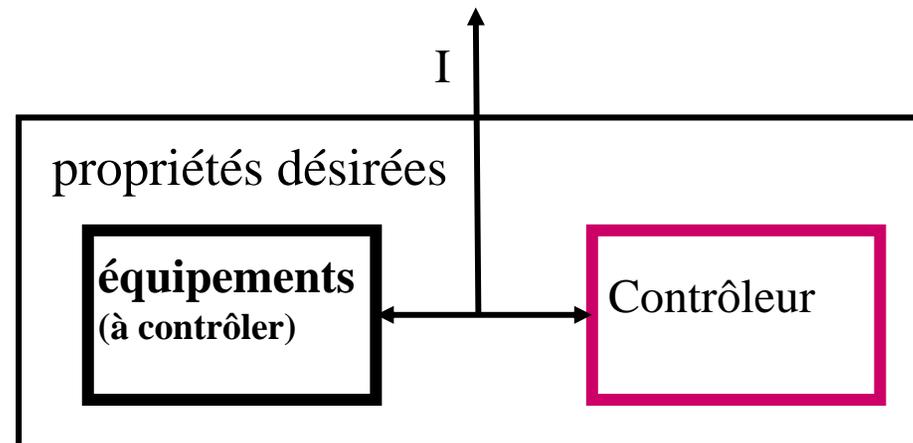
Le problème:

Trouver la spécification du module X , étant donné S et M_1



Les applications:

- Conception d'un contrôleur

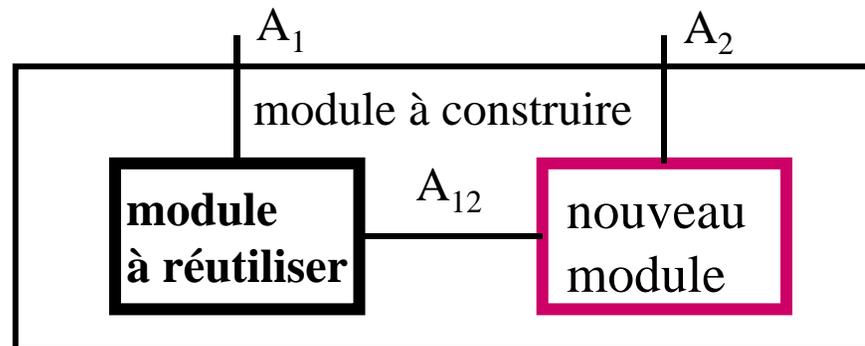
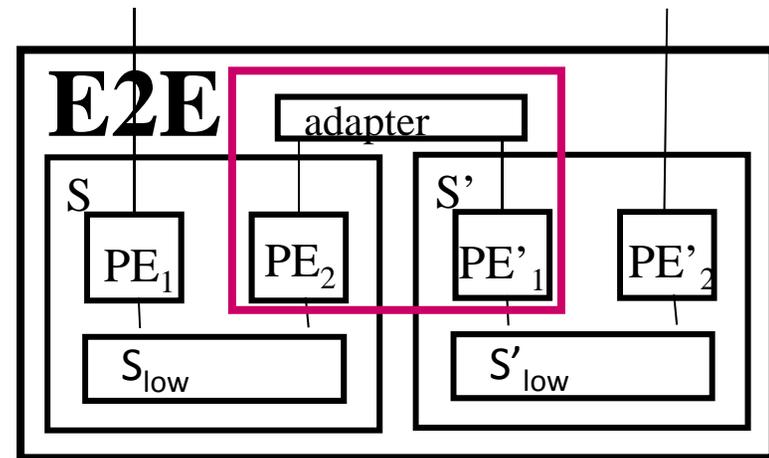
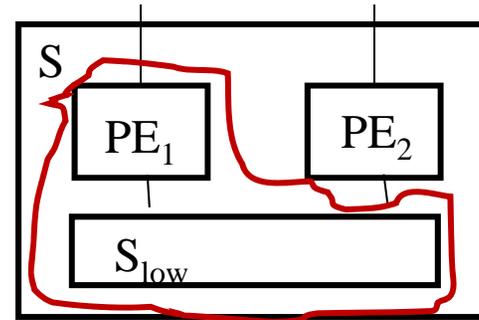


Questions de recherche récurrentes:

Dérivation de comportement de sous-modules (2)

Les applications (suite):

- Conception de protocoles
- Conception d'un convertisseur de protocoles
- La réutilisation



Questions de recherche récurrentes:

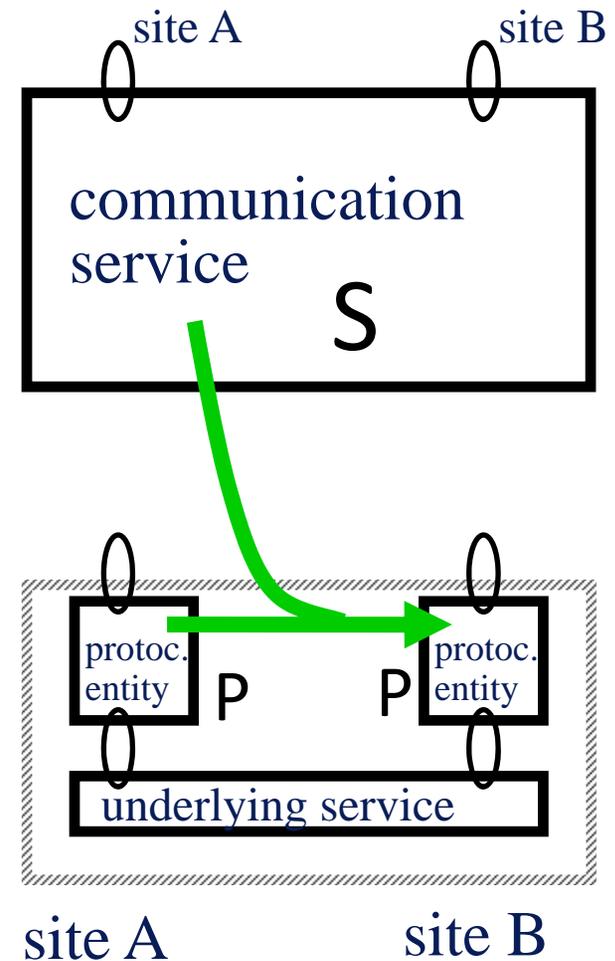
Dérivation de comportement de sous-modules (3)

- Idée initiale avec Philip Merlin en 1980
 - spécification en automates asynchrones
 - pour automates d'entrée-sortie et autres relations de conformité (dans les années 1990 avec les étudiants de doctorat Drissi et Tao)
- Une observation par Nina Yevtushenko en 2000:
 - similarité des formules pour automates asynchrones et synchrones
- Généralisation du problème et des solutions
 - Formulation du problème dans le contexte de banques de données (2001)
 - Formulation du problème dans le contexte de la logique de premier ordre et adaptation de la solution aux automates synchrones et asynchrones, communiquant par rendez-vous ou par entrée/sortie (2011)

Questions de recherche récurrentes: Dérivation d'un protocole à partir d'une spécification de service

Notes historiques:

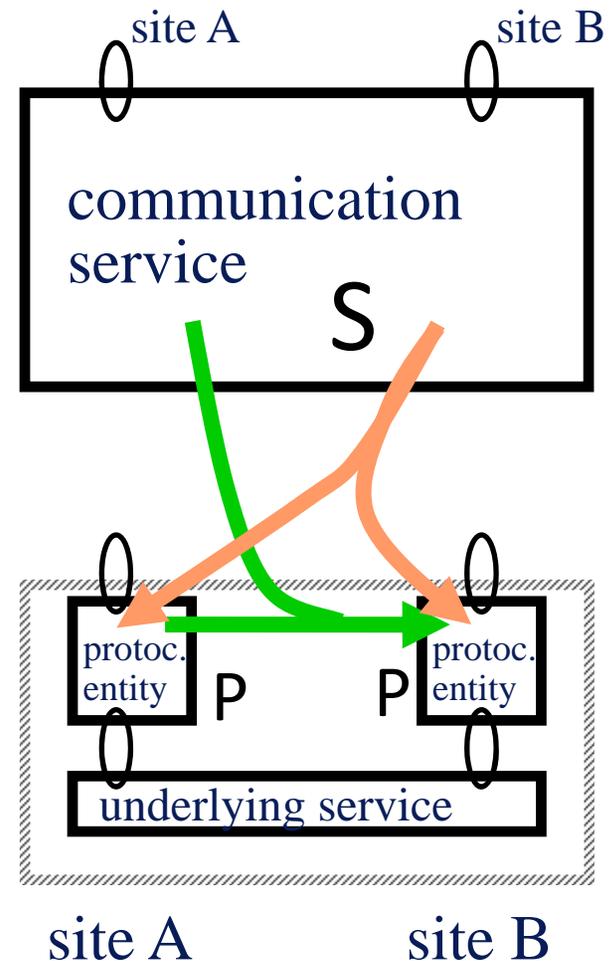
- 1978: le sens de “un protocole **P** offre le service **S**” (Finite State Description of Communication Protocols)
- 1980: construction de sous-module (avec Merlin)



Questions de recherche récurrentes: Dérivation d'un protocole à partir d'une spécification de service

Notes historiques:

- 1978: le sens de “un protocole **P** offre le service **S**” (Finite State Description of Communication Protocols)
- 1980: construction de sous-module (avec Merlin)
- 1986: dérivation de protocoles (with Gotzhein)
- 2002 with Yamaguchi, El Fakih (Osaka, Japon)
- 2006 with Braek, Castejon (Trondheim, Norvège)



Questions de recherche récurrentes:

Dérivation d'un protocole à partir d'une spécification de service (2)

Défis de recherche

- L'importance de la spécification du service de communication
 - mon article dans IEEE Tr Comm de 1980 ; “The Importance of the Service Concept in the Design of Data Communications Protocols” par Vissers et Logrippo en 1986
- Algorithmes de dérivation
 - Approche originale avec Gotzhein en 1986
 - Généralisation pour admettre le séquençement faible (2008)
- Automatisation et outils
 - Outils pour des spécifications de service en forme de réseaux de Petri (équipe à l'Université d'Osaka – dans les années 1990)
 - Outil pour l'optimisation de l'allocation des ressources sur les différentes composantes du système distribué (avec El Fakih et Yamaguchi – 2002)
 - Outil pour services en forme de diagrammes d'activités (Lamaarti, UofO, 2010)
- Utilisation dans le cadre de Services Web
 - Expérimentation et application de démonstration: Service en forme de diagramme d'activités, génération du comportements des composantes (en forme de diagrammes d'activités), traduction automatique en BPEL et implantation en forme d'une application distribuée utilisant le protocole SOAP des Services Web

Influence de la recherche sur l'enseignement

Permanence des concepts scientifiques

- **Ces concepts sont importants à enseigner**
- Exemple: l'approche orientée objet : Simula 1967, Smalltalk 1980, Java 1995, Services Web -
- ce qui n'est rien d'autre que l'orientation objet avec codage des données en XML
- Exemples reliés à mes recherches
 - Dérivation de comportement de sous-modules – dans différents langages de spécification et pour différentes applications
 - Dérivation d'un protocole à partir d'une spécification de service – mêmes principes pour différents langages de spécification
- Important: **enseigner les concepts qui restent valables / applicables pendant longtemps (pendant que les technologies changent)**

Influence de mon expérience de recherche sur l'enseignement

- Je connais très bien le domaine dans lequel j'ai fait des recherches
 - Je me sens à l'aise d'enseigner de tels sujets
 - Mentionner d'anciennes publications reliées au sujet enseigné rend l'enseignement plus crédible
- Certains aspects de mes recherches actuelles sont discutés dans mon cours gradué – apprentissage de la recherche
 - esprit de questionnement - comment trouver un bon sujet de recherche - méthodes et outils pour résoudre des problèmes - expérimenter avec des exemples

Collaborations avec Rennes

Visites

- Jean Pierre Verjus et Laurent Trilling (de Rennes) enseignent à l'Université de Montréal (fin des années 1960 – avant mon temps). Ils reviennent plus tard pour des visites plus courtes.
- Je fais une visite à Rennes en 1977 avant mon séjour comme professeur invité à Lausanne.
- Michel Raynal vient pour un séjour de recherche à l'Université de Montréal en 1981
- Claude Jard (du CNET) vient pour un séjour de recherche à l'Université de Montréal en 1982
- Je fais un séjour de trois semaines, en 1983, au CNET à Lannion. Aussi, pendant ces années, j'ai collaboré avec Roland Groz du CNET sur la normalisation d'Estelle et de SDL

Collaborations avec Rennes (2)

- **Publications en collaboration**

- avec Michel Raynal

- Spécification d'une composante avec interfaces multiples – propriétés locales et globales (1983)

- avec Jean Pierre Verjus

- Programmation structurée ou par machines d'états ? (1987)

- avec Claude Jard

- E.g. utiliser une spécification de service comme oracle lors de tests d'un système distribué (1983)

- **Rapporteur externe pour des thèses**

- J'étais rapporteur externe pour cinq thèses de doctorat à l'Université de Rennes entre 1981 et 2007

En guise de conclusion

**Le titre: Un parcours d'enseignant-chercheur:
contributions aux langages de programmation et aux
protocoles de communication**

Question: Quel est l'impact aujourd'hui ?

Réponse: Voici des exemples d'avancées en informatique reliées à mes travaux sur les protocoles de communication:

- L'architecture des protocoles en couches
- Le "model checking" pour systèmes distribués
- La notation UML et les outils associés
- L'ingénierie de systèmes dirigée par modèles

Impact aujourd'hui

L'architecture des protocoles en couches

- Ces concepts sont généralement acceptés et utilisés pour la conception de réseaux et de systèmes distribués

Le “model checking” pour systèmes distribués

- Les outils pour le “model checking” de systèmes distribués d'aujourd'hui sont basés sur l'analyse d'accessibilité et les outils antérieurs pour la vérification de l'absence de blocage et de réceptions non spécifiées; ils offrent en plus la possibilité de vérifier des propriétés spécifiques décrites en logique temporelle. Un exemple est l'outil SPIN.

Impact aujourd'hui (2)

La notation UML et les outils associés

- Parmi les trois TDF (Estelle, LOTOS et SDL), SDL avait le plus de succès. Il a été utilisé pour décrire beaucoup de normes de protocoles de communication et autres systèmes industriels, et les outils commerciaux ont été utilisés pour le développement d'implantations industrielles, par exemple dans le secteur de la téléphonie sans fil. Récemment, SDL a été intégré comme un profil dans la notation UML-2, et les outils ont été adaptés à ce nouveau contexte.

L'ingénierie de systèmes dirigée par modèles

- L'ingénierie de systèmes dirigée par modèles est devenue à la mode. Dans le contexte du génie des protocoles, cette approche a été utilisée dès le début. La spécification d'un protocole est un modèle abstrait de toutes les implantations, et la vérification du protocole est faite à ce niveau d'abstraction. En effet, les TDF SDL et Estelle, comme aussi les State Charts de Harel de 1987, sont basés sur le concept de machine à états étendue des années 1970, et ils peuvent être considérés comme les ancêtres des diagrammes d'états de la notation UML d'aujourd'hui.

En guise de conclusion (2)

Le titre:

Un parcours d'enseignant-chercheur . . .

Un autre impact de mon parcours, et pas le moindre:

Ce sont les amitiés

qui se sont développées au courant des années