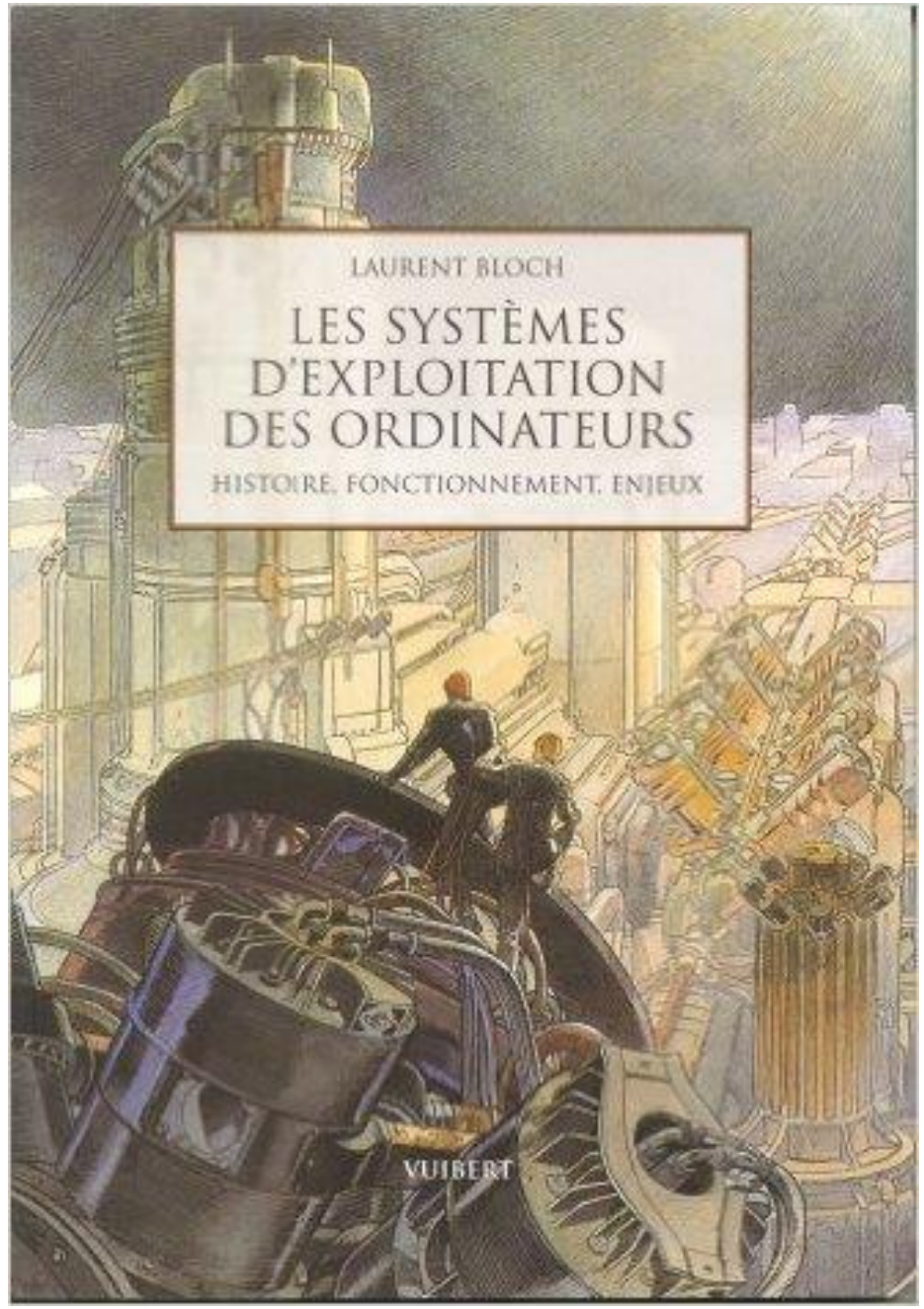
OUVERTURE

Du compliqué au complexe


LAURENT BLOCH
LES SYSTÈMES D'EXPLOITATION DES ORDINATEURS
HISTOIRE, FONCTIONNEMENT, ENJEUX
VUIBERT

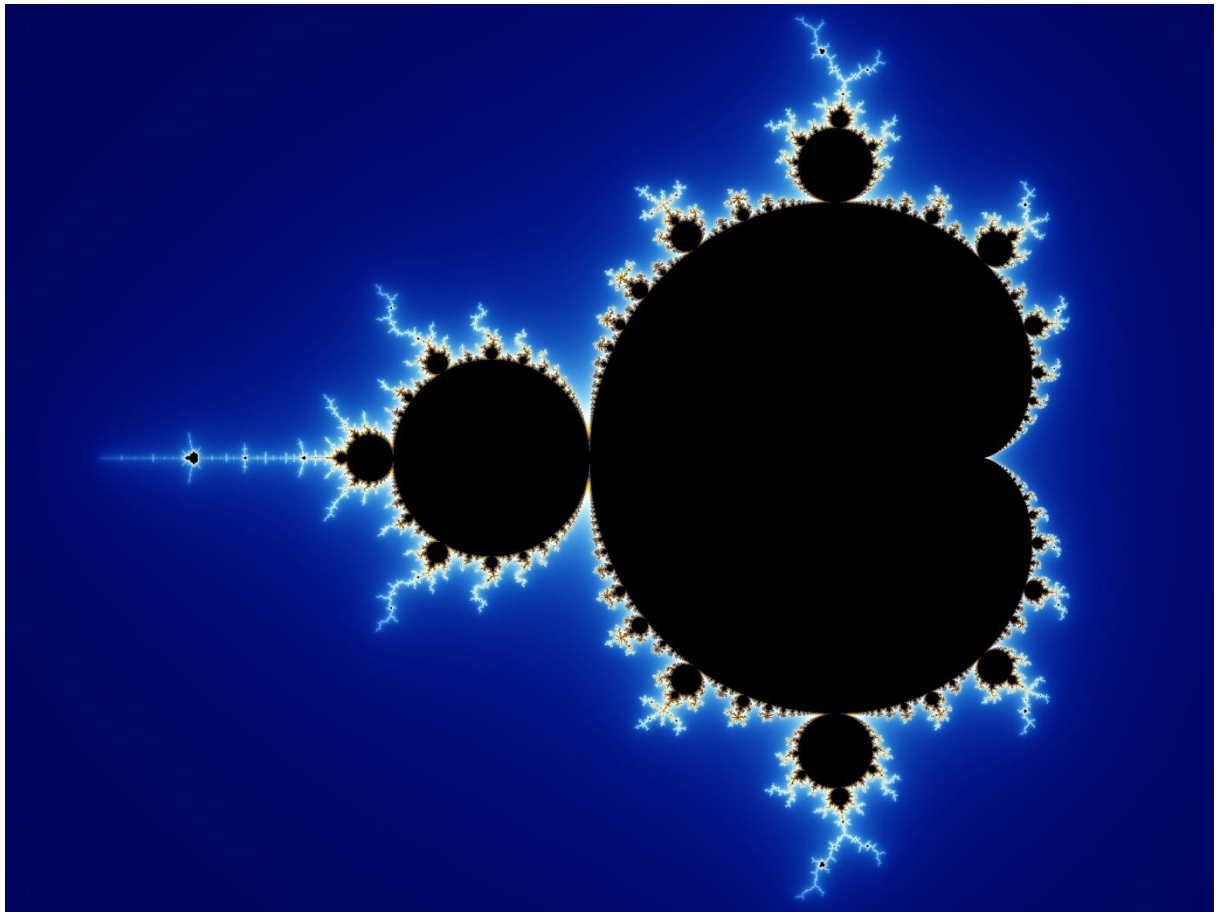
Ce livre constitue une remarquable introduction à l'informatique, science de l'abstrait, par le biais d'un de ses produits les plus immatériels mais les plus répandus : le système d'exploitation. On connaît mal l'extraordinaire complexité du système d'exploitation, sans doute l'une des réalisations techniques les plus ambitieuses du XXe siècle.

L'auteur nous entraîne à travers les grands domaines que doit gérer un système : les processus, la mémoire, le temps, la persistance des données, les échanges avec l'extérieur, la sécurité, l'interface personne-ordinateur et, enfin, la gestion des réseaux, notamment dans leur aboutissement actuel, Internet.

L'ouvrage montre ainsi combien, derrière une extrême diversité apparente, les mécanismes choisis pour réaliser le cœur du système sont en fait d'une grande similitude.

Gregory J. Chaitin, *From Philosophy to Program Size: Key Ideas and Methods, Lecture Notes on Algorithmic Information Theory*, Institute of Cybernetics, Tallinn, 2003.

Murray Gell-Mann, *The Quark and the Jaguar: Adventures in the Simple and the Complex*, New York, NY: Freeman, 1994.



Frederick P. Brooks, *The Mythical Man-Month*, Essays on Software Engineering, Addison-Wesley, 1982.

Barry Boehm et al., *Software Cost Estimation with COCOMO II*, Englewood Cliffs, NJ, Prentice Hall, 2000.

Jacques Printz, *Productivité des programmeurs,* Hermès Science, 2001.

Jacques Printz, *Puissance et limite des systèmes informatisés,* Hermès Science, 1998.

Raymond J. Madachy, *Software Process Dynamics,* Wiley-Interscience, 2008.

Définition

Informatique autonome : n.f. ensemble des techniques utilisées pour permettre à un réseau ou à un système informatique de gérer lui-même les éléments qui le composent et d'entreprendre automatiquement les actions nécessaires à son fonctionnement optimal, sans intervention humaine.

synonyme(s) : informatique autogérée, informatique autorégulée.

Grand dictionnaire terminologique
Office québécois de la langue française, 2003
http ://www.granddictionnaire.com/

Objectifs et concepts

• Mettre l'informatique au service de la gestion des applications.

• Élever considérablement le niveau de fiabilité des services informatiques.

• Réduire le coût humain de la fonction de gestion des services et applications.

Caractéristiques d'un système autorégulé I

• Connaissance de lui-même, y compris la capacité à s'introspecter et à agir sur lui-même ;
• Connaissance de son environnement, y compris la capacité à réagir à ce qui s'y passe ;
• Auto-configurabilité, en fonction des conditions de déploiement et de leur évolution en cours d'exécution ;
• Auto-optimisabilité, en fonction des usages et des ressources disponibles ;
• Auto-réparabilité, lors de pannes ou d'événements imprévus ;
• Auto-protection, face aux attaques externes ;

Caractéristiques d'un système autorégulé II

• Ouverture, ne s'enferme pas dans des contextes hermétiques mais on s'ouvre plutôt à l'extérieur en adhérant à des standards reconnus ;

• Capacité d'anticipation des besoins de l'utilisateur dans son auto-gestion.

Infrastructure autorégulatrice : individuellement

- Composants autogérés : du matériel aux composants métiers.

- Éléments gérés (managed elements) :
  composants implantant des interfaces capteurs et actuateurs.

- Éléments autorégulateurs : boucle de contrôle intelligente avec :
  - collecte des informations sur l'exécution,
  - traitement et diagnostic
  - intervention sur l'élément géré

Comment réaliser cette vision ?

• Quelle est la structure d'un élément géré ? Quelles sont ses interfaces capteurs et actuateurs ?

• Comme s'intègrent-ils dans la structure globale de l'application ?

• Comment interagissent-ils les uns avec les autres ?

• Quel est leur état (y compris leurs dépendances envers leur environnement) et comment évoluent-ils dynamiquement entre ces états ?

• Comment modéliser l'historique de leurs changements de comportement étant donné leur état ?

Quelques références

Manish Parashar, Salim Hariri, *Autonomic Computing: An Overview*, LNCS 3566, pp. 247–259, Springer-Verlag, 2005

Markus C. Huebscher, Julie A. McCann, *A survey of Autonomic Computing — degrees, models and applications*, ACM

Jeffrey O. Kephart, *Research Challenges of Autonomic Computing*, ICSE'05, pp. 15-22, ACM, 2005

Mohammad Reza Nami, Koen Bertels, *A Survey of Autonomic Computing Systems*,

…

L'environnement d'exploitation

Un système ne fonctionne correctement que si l'environnement satisfait à certaines conditions.

Le système doit donc avoir, en son sein, une représentation de cet environnement, ne serait-ce que pour détecter les situations où il est susceptible d'être actif hors du contexte dans lequel il peut garantir son bon fonctionnement.
(La défaillance d'Ariane 5, en juillet 96, est typique de ce cas de figure.)

Ce point est fondamental pour les systèmes à forte criticité qui doivent parfaitement contrôler leur environnement. Si cette distinction n'est pas faite, il sera très difficile, en cas de défaillance, de savoir si l'erreur est dans le logiciel ou si les conditions extérieures sont telles qu'une erreur est inévitable.

Vers la programmation systémique

• Composants gérés : utilisation de la réflexion pour réaliser la « connaissance de lui-même ».

• Intégration de notions de décision et planification :
le méta-niveau joue le rôle d'élément autorégulateur avec modèles de décisions de la RO et de l'IA.

• Découpage de la décision en deux parties :
  ◦ homéostatique : maintien des équilibres à court terme avec une décision purement réactive.
  ◦ délibératif : inférence de nouvelles politiques sur le plus long terme.

Pourquoi systémique ?

Systémique : science de l'étude des systèmes trop complexes pour être abordés de manière réductionniste.

- Même besoin de contrôle réparti comportant :
  - maintien des structures existantes (homéostasie, autorégulation), et
  - évolution par apprentissage à partir des interactions avec l'environnement.

- Références : cybernétique, théorie des systèmes généraux, macroscope de Joël de Rosnay.

La science des systèmes - Les sciences de l'artificiel

Lire l'article Wikipedia sur Jean-Louis Lemoigne et explorer les liens

Prépare-t-on convenablement les ingénieurs de demain en abordant aussi peu ce qui fera l'essentiel de leur quotidien dans la vie active, à savoir la conception de projets complexes en situations complexes pour laquelle l'application des sciences dures (dites "fondamentales") n'aura généralement qu'une utilité secondaire ?

Recherche Opérationnelle

Enseignement des "nouvelles sciences du management" à l'Université

Groupe de Recherche sur l'Adaptation, la Systémique & la Complexité Économique

Programme Européen Modélisation de la Complexité (MCX)

La théorie du système général

Théorie de la modélisation

JEAN-LOUIS LE MOIGNE

COLLECTION
LES CLASSIQUES DU RESEAU
INTELLIGENCE DE LA COMPLEXITE
www.mcxapc.org

Chapitre 6. — **Le projet du Système Général :**
une intervention finalisante dans un environnement.

1. Des « points d'articulation naturels »
Séparabilité et articulations naturelles
Frontières, finalités et interprétations génétiques

2. Une articulation en neuf niveaux
Le premier niveau : l'objet passif et sans nécessité
Le deuxième niveau : l'objet actif
Le troisième niveau : l'objet actif et régulé
Le quatrième niveau : l'objet s'informe
Le cinquième niveau : l'objet décide de son activité, (l'objet a donc quelques projets)
Le sixième niveau : l'objet actif a une mémoire
Le septième niveau : l'objet actif se coordonne
Le huitième niveau : l'objet actif imagine, donc s'auto-organise
Le neuvième niveau : l'objet actif s'auto-finalise

Stephen M. Omohundro, "The Nature of Self-Improving Artificial Intelligence"

Self-improving systems are a promising new approach to developing artificial intelligence. But will their behavior be predictable? Can we be sure that they will behave as intended even after many generations of self-improvement?

Self-improvement causes systems to converge to an architecture that arises from von Neumann's foundational work on microeconomics. Self-improvement causes systems to allocate their physical and computational resources according to a universal principle. It also causes systems to exhibit four natural drives: 1) efficiency, 2) self-preservation, 3) resource acquisition, and 4) creativity. Unbridled, these drives lead to both desirable and undesirable behaviors.

How can we ensure that this technology acts in alignment with our values? We have leverage both in designing the initial systems and in creating the social context within which they operate. But we must have clarity about the future we wish to create. We need not just a logical understanding of the technology but a deep sense of the values we cherish most. With both logic and inspiration we can work toward building a technology that empowers the human spirit rather than diminishing it.