

3. Infrastructures logicielles des systèmes urbains

- Les systèmes logiciels distribués de la ville intelligente
- Intégration des objets connectés
- Intégration des capteurs sociaux
- Faire face à l'échelle des systèmes urbains
- **Faire face à l'hétérogénéité des systèmes urbains**

Valérie Issarny

VILLES INTELLIGENTES : DÉFIS TECHNOLOGIQUES ET SOCIÉTAUX

Concluons cette semaine sur les infrastructures logicielles des systèmes urbains en nous intéressant au problème de l'hétérogénéité de ces systèmes.

Hétérogénéité des systèmes urbains



En effet, les systèmes urbains sont typiquement composés de :

- **systèmes propriétaires** comme par exemple les systèmes de gestion et d'énergie,

Hétérogénéité des systèmes urbains



Systemes propriétaires



Internet des objets



3

- mais également des **systemes relatifs à l'Internet des objets** qui permettent de coupler monde physique et monde virtuel.

Hétérogénéité des systèmes urbains



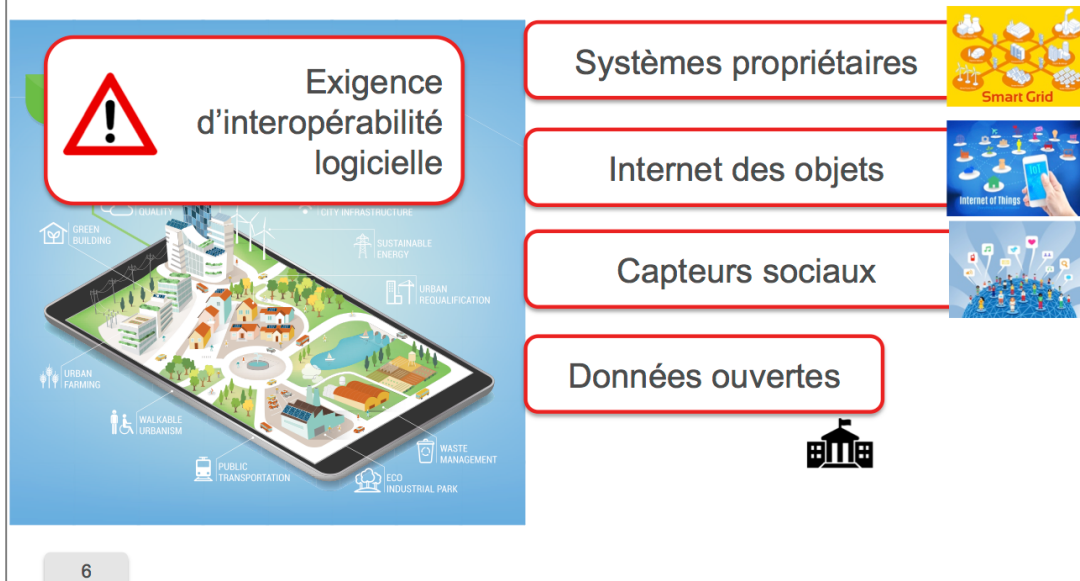
- **capteurs sociaux**, ou encore capteurs humains qui permettent d'accéder ou de bénéficier des données communiquées, sur les réseaux sociaux, ou encore les données fournies par les utilisateurs eux-mêmes, via les systèmes participatifs.

Hétérogénéité des systèmes urbains



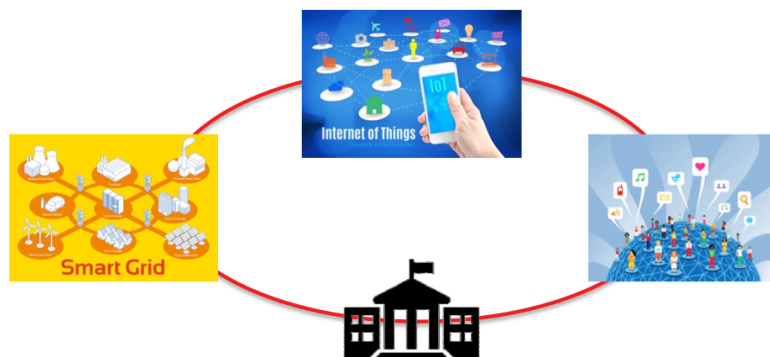
- **données ouvertes** qui sont fournies par les gouvernements, eux-mêmes.

Hétérogénéité des systèmes urbains



Ce que nous voulons, c'est pouvoir construire des systèmes urbains qui intègrent ces différentes sources de données. Nous avons alors à faire face à une **exigence d'interopérabilité logicielle**.

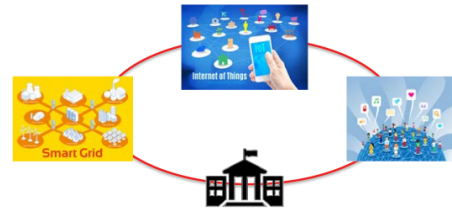
Interopérabilité logicielle



7

L'interopérabilité logicielle, qu'est-ce que c'est ? Cela consiste à permettre à des systèmes hautement hétérogènes et diverses comme ce que nous venons de mentionner, **d'interopérer**, c'est-à-dire de pouvoir **communiquer**, **échanger** des informations et **réaliser** conjointement de nouvelles applications.

Interopérabilité logicielle



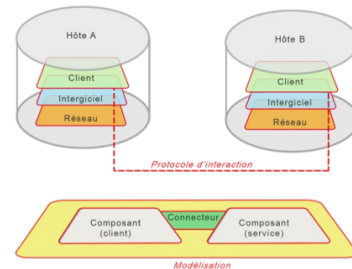
Tanenbaum & Van Steen [TANENBAUM] :
“the extent by which two implementations of systems from different manufacturers can co-exist and work together by merely relying on each other’s services as specified by a common standard”.

8

L’Interopérabilité logicielle a en particulier été définie par Tannenbaum et Van Steen qui la définissent comme l’aptitude à pouvoir travailler ensemble : *“the extent by which two implementations of systems from different manufacturers can co-exist and work together by merely relying on each other’s services as specified by a common standard”.* Et dans leur cas, ils ont posé que ceci devait se faire en s’appuyant sur un standard commun.

Interopérabilité : Solution intergicielle

- Système distribué logiciel pour l'échange d'informations et la coordination d'applications réparties dans le réseau
 - **Abstractions de communication et coordination**
 - ✓ Modélisation des systèmes et des données
 - ✓ Paradigmes de coordination



9

Plus globalement, nous avons vu que l'interopérabilité devait s'appuyer sur une solution intergicielle qui découle de travaux dans le domaine des systèmes distribués depuis les années 90.

Qu'est-ce qu'un intergiciel ?

C'est un système distribué qui facilite l'échange d'informations, et la coordination d'application répartie dans le réseau.

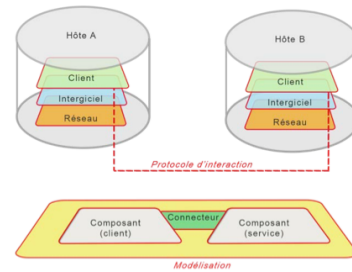
Pour se faire, un intergiciel va définir des abstractions de communication et de coordination.

Nous allons donc avoir :

- des modèles de systèmes et de données,
- des paradigmes de coordination.

Interopérabilité : Solution intergicelle

- Système distribué logiciel pour l'échange d'informations et la coordination d'applications réparties dans le réseau
 - **Abstractions de communication et coordination**
 - ✓ Modélisation des systèmes et des données
 - ✓ Paradigmes de coordination
 - **Technologie**
 - ✓ Langage de définition d'interface
 - ✓ Protocoles de communication et coordination
 - ✓ Protocoles de gestion de la qualité de service

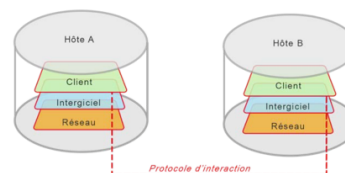


10

Étant donné ces modèles et paradigmes, une intergicelle va implémenter des technologies ou un ensemble de technologies qui sont un **langage de définition d'interfaces**, des **protocoles de communication et coordination**, et des **protocoles de gestion de la qualité de service**.

Interopérabilité : Solution intergicielle

- Système distribué logiciel pour l'échange d'informations et la coordination d'applications réparties dans le réseau
 - **Abstractions de communication et coordination**
 - ✓ Modélisation des systèmes et des données
 - ✓ Paradigmes de coordination
 - **Technologie**
 - ✓ Langage de définition d'interface
 - ✓ Protocoles de communication et coordination
 - ✓ Protocoles de gestion de la qualité de service



Hétérogénéité des intergiciels

Le **problème** auquel on a été confronté, suite à l'émergence des solutions intergicielles, c'est l'hétérogénéité des intergicielles de données. Des solutions à l'interopérabilité des intergiciels ont donc dû être développées.

Les solutions à l'interopérabilité



Traduire d'un protocole à l'autre : les **ponts logiciels**

12

La solution la plus simple consiste à implémenter un **pont logiciel** qui est spécifique à deux protocoles de 2 intergiciels donnés. Elle est très simple, mais elle est aussi très limitatif, puisqu'il faut implémenter un point logiciel à chaque fois que l'on veut coupler deux systèmes hétérogènes.

Les solutions à l'interopérabilité



Traduire d'un protocole à l'autre : les **ponts logiciels**



Choisir un protocole de référence : les **standards**

13

Une solution que nous avons déjà évoquée consiste à **définir un standard**. On va donc avoir des standards d'intergiciel en supposant que les différents fournisseurs d'équipements vont implémenter ces standards. Nous savons que cette solution est limitative, puisque dès lors que nous avons de nouvelles solutions logicielles, de nouveaux standards émergent, et donc, nous devons faire face à l'interopérabilité entre standards.

Les solutions à l'interopérabilité



Traduire d'un protocole à l'autre : les **ponts logiciels**



Choisir un protocole de référence : les **standards**



Traduire en un protocole intermédiaire commun : les **bus logiciels**

14

C'est pourquoi les **bus logiciels** ont été introduits. Un bus logiciel définit un protocole intermédiaire commun. Et dès lors que l'on veut introduire un nouvel intergiciel ou nouveau système, on va implémenter un pont de logiciels, entre ce nouveau système et le bus logiciel. Toutefois, la limitation des bus logiciel est qu'en imposant un protocole intermédiaire commun, tous les systèmes mis en présence dans les systèmes urbains peuvent ne pas se mapper sur ce protocole intermédiaire. Par exemple, si l'on reprend l'Internet des objets, ou encore le système propriétaire de gestion de l'énergie, on imagine très bien que les abstractions, aussi bien de communication que de données vont être très différentes. Envisager un protocole intermédiaire commun pour sa diversité des systèmes n'est donc pas approprié.

Les solutions à l'interopérabilité



Traduire d'un protocole à l'autre : les **ponts logiciels**



Choisir un protocole de référence : les **standards**



Traduire en un protocole intermédiaire commun : les **bus logiciels**



Parler le protocole de l'autre : **adaptation** à la volée du protocole exécuté

15

C'est pourquoi la recherche s'est intéressée à l'**adaptation automatique de protocoles** qui consiste à **générer à la volée, des traducteurs**.

Vers une interopérabilité universelle...

- **Des systèmes urbains qui sont massivement hétérogènes et dynamiques**
 - Hétérogénéité des protocoles intergiciels et applicatifs
- **Découverte, apprentissage et synthèse de protocoles à la volée**
 - Exploitation des ontologies pour la modélisation des systèmes
 - Découverte et composition dynamiques des systèmes du réseau urbain
 - Apprentissage des protocoles exécutés par les systèmes connectés
 - Synthèse de contrôleur pour traduire et coordonner les exécutions de protocoles des systèmes répartis

16

Étant donné une solution à l'interopérabilité ou non, qu'elle soit universelle, ou non, nous pouvons collecter les données mises à disposition par les différents systèmes hétérogènes déployés dans le réseau urbain. Nous pouvons ensuite traiter et analyser ces données. Pour se faire, nous faisons appel à l'informatique en nuage qui vous sera présentée la semaine suivante.

Références bibliographiques

- [TANENBAUM] Andrew S. Tanenbaum and Maarten van Steen. Distributed Systems: Principles and Paradigms. <http://www.cs.vu.nl/~ast/books/ds1/>
- Marco Bernardo, Valérie Issarny: *Formal Methods for Eternal Networked Software Systems - 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems*, SFM 2011, Bertinoro, Italy, June 13-18, 2011. Advanced Lectures. Lecture Notes in Computer Science 6659, Springer 2011, ISBN 978-3-642-21454-7.
- Gordon S. Blair, Amel Bennaceur, Nikolaos Georgantas, Paul Grace, Valérie Issarny, Vatsala Nundloll, Massimo Paolucci: *The Role of Ontologies in Emergent Middleware: Supporting Interoperability in Complex Distributed Systems*. Middleware 2011: 410-430.
- Nikolaos Georgantas, Georgios Bouloukakis, Sandrine Beauche, Valérie Issarny: *Service-Oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability*. ESOC 2013: 134-148.
- Amel Bennaceur, Valérie Issarny: *Automated Synthesis of Mediators to Support Component Interoperability*. IEEE Trans. Software Eng. 41(3): 221-240 (2015)

Illustrations & photos : crédits

p. 2-6 : © elenabsl, Shutterstock ; © chombosan, Fotolia

p. 3-6 : © weedezn, Fotolia

p. 4-6 : © Cienpies Design, Fotolia

p. 7-8 : © chombosan, Fotolia ; © weedezn, Fotolia ; © Cienpies Design, Fotolia

p.12-15 : © creepjank, Shutterstock

p. 13-15 : © Rawpixel, Shutterstock

p. 14-15 : © TechnoVectors, Shutterstock

p. 15 : © file404, Shutterstock