

Défis dans l'ingénierie logiciel des Systèmes de Systèmes

Laboratoire LIUPPA, Université de Pau et Pays de l'Adour
Archware, Laboratoire IRISA, Université de Bretagne-Sud

2 novembre 2014

La complexité croissante de notre environnement socio-économique se traduit en génie logiciel par une augmentation de la taille des systèmes et par conséquent de leur complexité. Les systèmes actuels sont le plus souvent concurrents, distribués à grande échelle et composés d'autres systèmes. Ils sont alors appelés Systèmes de Systèmes (SdS). Un système de systèmes est une intégration d'un certain nombre de systèmes constituants indépendants et inter-opérables, interconnectés pour un période de temps donnée dans le but d'assurer une certaine finalité. Dans la pratique, les systèmes de systèmes sont des systèmes complexes à logiciel prépondérant. Les systèmes constituants, non seulement pré-existent au besoin à couvrir, plus encore, ils ont été conçus à des fins indépendantes de la finalité recherchée. Ils continuent par ailleurs à être opérés, maintenus, gérés par les différentes organisations dont ils sont issus, voire évoluent de manière indépendante. Mais, ils doivent, pour un temps donné, de manière non anticipée, partager des informations, inter-opérer, pour faire émerger la finalité recherchée.

Après des systèmes dont l'unité de composition de base était des fonctions, puis des composants, maintenant il s'agit de développer des systèmes dans lesquels les unités de composition de base sont des systèmes eux-mêmes. La complexité des SdS réside dans leurs cinq caractéristiques intrinsèques qui sont : l'indépendance *opérationnelle* des systèmes constituants, l'indépendance *managériale* de ces mêmes systèmes, la distribution géographique, l'existence de comportements émergents, et enfin un processus de développement *évolutionnaire*. Clairement il ne suffira pas de simplement adapter les méthodes et outils existants de développement, le défi soulevé est la proposition de concepts, de langages, de méthodes et d'outils permettant la construction de SdS capables de s'adapter dynamiquement pour continuer d'assurer leur mission malgré les évolutions internes ou externes qu'ils subissent.

La caractéristique de *développement évolutif* de SdS signifie que la conception originale d'un SdS doit tenir compte des systèmes constituants existants, qui sont parfois traités en génie logiciel en tant que des systèmes hérités (legacy). En outre, à tout moment, de nouveaux systèmes peuvent être ajoutés ou supprimés, et pas seulement au moment de la conception, mais également à l'exécution. De plus, chaque système constituant subira des évolutions, des mises à jour, qui pourront avoir une incidence sur l'ensemble du SdS. Les cycles de vie seront longs, des dizaines d'années pour des SdS tels que ceux de la défense, par exemple. Par conséquent, les cycles de vie itératifs flexibles seraient probablement les plus adéquats.

Du point de vue de l'analyse des besoins, l'évolution continue des besoins est encore plus accentuée par les *comportements émergents* des SdS. Certains de ces comportements sont conçus, mais certains ne sont ni conçus, ni voulus. Comment peut-on analyser les besoins des SdS ? Comment les mesurer ? Comment trouver un équilibre entre les différentes exigences de SdS, notamment les propriétés non-fonctionnelles qui sont souvent contradictoires, telles que la performance, la sécurité, la tolérance aux pannes ? Comment équilibrer les exigences des différents systèmes constituants ?

Face à l'*indépendance opérationnelle* des systèmes constituants il faut être capable d'architecturer et construire un SdS qui pourra remplir sa propre mission, sans violer l'indépendance de ses systèmes constituants qui sont autonomes et ont leur propre mission à remplir. De plus, il devra d'adapter pour répondre aux changements de caractéristiques de son environnement et de ses systèmes constituants.

Les équipes du GDR GPL impliquées dans ce défi proposent de s'attaquer, d'une part, à la modélisation et l'analyse de propriétés non-fonctionnelles des SdS en particulier aux défis liés à la sécurité et, d'autre part, à l'analyse, au développement et à l'évolution dynamique des architectures de SdS.