

Assemblage dynamique et opportuniste de composants logiciels pour les systèmes ambiants, mobiles et autonomes

Stage de fin d'études d'Ingénieur ou de Master 2 - 2016-2017

Mots-clés : ingénierie logicielle, intelligence artificielle, systèmes multi-agents adaptatifs, IHM, informatique ubiquitaire et mobile

1. Contexte et problématique.

Le **développement logiciel** traditionnel repose sur la conception de programmes ou d'architectures afin de répondre à des besoins préalablement explicités. L'évolution des logiciels et leur adaptation à des conditions d'exécution en perpétuelle évolution constituent cependant un défi majeur. Or, les besoins évoluent et, de plus en plus, l'infrastructure sur laquelle repose le logiciel évolue davantage que les fonctionnalités offertes. C'est en particulier le cas dans les systèmes ambiants ou mobiles, la robotique distribuée et autonome, ou l'Internet de objets (IoT), de par la dynamique de ces environnements voire leur ouverture et leur imprévisibilité. La définition d'une **approche nouvelle** pour la construction d'applications « de manière **opportuniste** » a ainsi été amorcée pour apporter une réponse originale à ces problèmes.

Quelle que soit l'approche, le développement logiciel s'effectue pour partie en mode descendant (ou **top-down**) et pour partie en mode ascendant (ou **bottom-up**). Le développement traditionnel, en mode top-down, est principalement guidé par la satisfaction des besoins. Il inclut néanmoins une partie bottom-up dans laquelle on considère les composants logiciels existants et on essaye de les réutiliser (pour satisfaire des exigences de productivité par exemple). L'approche opportuniste privilégie le mode bottom-up : elle considère prioritairement les composants logiciels existants et vise à produire des logiciels en fonction de ceux-ci, parce que les circonstances en offrent l'opportunité. Par exemple, on peut imaginer que des composants logiciels disponibles sur différents appareils communicants soient mis en interaction et forment ensemble une application « intéressante » pour l'utilisateur. En particulier, ces composants peuvent être des composants d'IHM et le problème est de proposer à l'utilisateur une interface adaptée.

Pour produire ces applications, il est possible de s'appuyer sur des modèles (des plans d'application) et de les instancier dynamiquement en fonction des circonstances c.-à-d. des composants logiciels disponibles. Dans l'approche opportuniste proposée par l'équipe SMAC¹ de l'IRIT, les composants logiciels sont assemblés automatiquement et dynamiquement parce qu'ils sont disponibles et en situation d'interagir sans que leur composition n'ait été planifiée, faisant ainsi **émerger des applications**. Ici, l'« intérêt » de l'utilisateur n'est pas explicite : il doit être observé afin de déterminer l'utilité de la composition, de décider de sa préservation et d'aider à la sélection de

¹ <http://www.irit.fr/-Equipe-SMAC->

futures compositions pertinentes. On retrouve donc ici une partie top-down dans laquelle l'adéquation entre le logiciel et les besoins doit être contrôlée. De manière générale, cette approche pose un certain nombre de problèmes parmi lesquels l'identification des composants et la sensibilité au contexte, la composabilité (connexion entre services requis et fournis) et la sémantique de l'assemblage obtenu, la réalisation automatique de la composition, la maîtrise de la combinatoire et la validation (prise en compte de l'intérêt de l'utilisateur). La solution que nous développons est fortement décentralisée, basée sur un système multi-agent qui est le moteur de la composition opportuniste : des agents, situés au niveau des composants et des services et dotés de capacités d'apprentissage, décident localement des connexions et les supervisent.

Ce projet de recherche entre dans le cadre de l'opération neOCampus² de l'université Paul Sabatier, et fait l'objet d'une collaboration avec le groupe Rainbow du laboratoire I3S de l'Université de Nice – Sophia Antipolis³ et l'équipe IHM du Laboratoire d'Informatique de Grenoble⁴.

2. Objectif du stage.

Le travail réalisé dans le cadre de plusieurs stages en 2015-2016 a permis de produire une 1^{re} version en Java d'un **prototype de moteur de composition opportuniste**, ainsi qu'une démo (à base de smartphone et tablette Android, carte Arduino...). Le prototype permet la composition de composants logiciels disponibles dans l'environnement et les compositions se font et se défont au fur et à mesure des changements dans les conditions d'exécution. Il s'appuie sur la plateforme à composants WComp⁵ (développée par le groupe I3S/Rainbow) et le protocole UPnP⁶ pour la connexion entre appareils et la découverte de service.

Le stage que nous proposons a pour objectif de consolider la solution au niveau de la présentation des composants d'IHM d'une part et au niveau de l'apprentissage multi-agent d'autre part. Le premier point consiste à prendre en compte les caractéristiques propres aux composants graphiques et à réfléchir aux interactions possibles avec l'utilisateur pour la validation des compositions (prise en compte du feedback, méta-IHM...). Concernant le moteur de composition, il s'agit d'enrichir les mécanismes d'apprentissage des agents pour la prise en compte du contexte et des compositions antérieures.

Les propositions seront validées au travers de cas d'utilisation réalistes, en particulier en lien avec l'opération NeoCampus qui vise la conception d'un campus connecté, innovant, intelligent et durable.

3. Conditions du stage.

Le stage se déroulera au sein de l'équipe IRIT-SMAC sur le site de l'Université P. Sabatier à Toulouse. Il sera encadré par S. Trouilhet et J.-P. Arcangeli.

Ce travail pourra éventuellement être poursuivi dans le cadre d'une thèse de doctorat.

Contact : jean-paul.arcangeli@irit.fr sylvie.trouilhet@irit.fr

² <https://www.irit.fr/neocampus/>

³ <https://rainbow.i3s.unice.fr/>

⁴ <http://iihm.imag.fr/>

⁵ <https://www.wcomp.fr/>

⁶ <http://www.upnp.org/>