Programmer des composants actifs dans le web sémantique

William Portillon
LACIDE-CNS
BP 311, 34014 Trieste

1 Novembre 2015

Résumé

Cet article, montre la possibilité d'utiliser XML comme langage de programmation des composants actifs dans le web sémantique, pour unifier la description de leur partie structurelle et celle de leur fonctionnement. Un modèle d activité de composants actifs est présenté, il permet d'avoir accès en cours d'exécution à une description sémantisée de leur fonctionnement pour pouvoir répondre à des requêtes sur leurs actions.

1 Les composants actifs dans le web sémantique

Les modèles actuels du web sémantique permettent de munir les pages d'une représentation structurelle sémantisée [1] pour répondre à des requêtes sur la signification de l'information qu'ils manipulent [2, 3]. Au contraire, les composants actifs dans le web (services, agent, etc.) avec lesquels les utilisateurs ordinaires vont interagir sont caractérisés par « un fonctionnement » [4]. Contrairement au script CGI qui ne fait que traiter, manipuler ou produire l'information contenue sur le web, ce fonctionnement fait partie intégrante de l'information. Le web sémantique doit donc permettre de représenter et de manipuler des connaissances non seulement structurelles, mais aussi de nature fonctionnelle, c est-à-dire de représenter le fonctionnement des composants actifs qui y agissent. Plusieurs modèles de description de services web ont été proposés, comme WSDL [5], WSMF [6] ou DAML-S [7] comme extension des protocoles de base pour les services (SOAP [8], UDDI1 et WSDL [5]) afin de permettre aux agents de trouver, invoquer, surveiller ou composer en services plus complexes des ressources Web offrant tels services et

vérifiant telles propriétés. Mais dans tous ces modèles, la description du service est séparée de l'information manipule. Au contraire, dans un composant actif, la structure et le fonctionnement doivent être intégrés au sein d'une même représentation, pour lui permettre d'interagir avec l'utilisateur en vue de pouvoir répondre à des requêtes sémantisées aussi bien sur la structure [9, 3] de l'information que sur le fonctionnement du composant. Il faut donc pouvoir programmer directement en XML les composants actifs. Dans cet article, sera décrit le principe général de fonctionnement des composants dans le modèle retenu.

2 Principe général

Le modèle d'exécution retenu pour les composants actifs décrits en XML s'appuie sur la notion « d'observateur ». Un observateur est un composant logiciel externe qui recherche dans le document XML des éléments spécifiques, conformément un schéma de description ou une DTD (qui définit la syntaxe respecter), et les interprète pour effectuer des actions particulières. Nous pouvons alors opposer les observateurs statiques qui ne modifient pas le document de manière proactive [10, 4] (par exemple un observateur produisant une interface utilisateur ou un observateur de réponse aux requêtes structurelles sémantisées) à l'observateur dynamique qui considère le document passé en paramètre comme la description du composant à l'instant t et qui construit sa description à l'instant suivant $t + 1$ en fonction des éléments spécifiques décrivant le fonctionnement de ce composant actif. L'excution d'un composant consiste donc en la réécriture du document XML à chaque cycle en fonction de concepts spécifiques, comme dans les algèbres évoluantes [11] ou « Maude » [12]. La description du composant en XML est sérialisée [13] à chaque cycle d'exécution pour que tous les observateurs puissent interpréter le nouveau document. Ces composants sont proactifs : ils peuvent s'exécuter en dehors de toute interaction avec l'utilisateur. Le modèle d'exécution basé sur les observateurs permet de décrire des documents à la fois structurés et dynamiques. Toutefois, il s'appuie sur l'hypothèse que les observateurs sont robustes : si un élément recherché n'est pas trouvé dans un document ou si le document n'est pas conforme à la syntaxe de l'observateur, alors aucune action n'est effectue par cet observateur.

3 Conclusion

Le web sémantique aujourd'hui s'est beaucoup orienté vers la description de l'information structurelle sémantisée et prend peu en compte l'étude du fonctionnement des services et, plus généralement, des composants actifs interagissant avec les utilisateurs. Ces composants sont aujourd'hui généralement « enchassés » dans les documents sous la forme d'applets, et non intégrés avec le document. Cet article, propose le principe général d'un modèle d'activité de composants actifs qui permet d'attacher, sur les aspects structurels riches, des aspects liés au fonctionnement. Il est alors possible, en utilisant ce modèle, de décrire et d'exécuter les composants actifs sous la forme de pages web en XML. Le modèle proposé reste relativement simple et il est suffisamment générique pour permettre de représenter une large classe de composants actifs.

Références

[1] C. Reynaud, B. Safar, and H. Gagliardi. Une expérience de représentation d'une ontologie dans le médiateur picsel. In *IC 2001*, pages 329 – 348, 2001.

[2] W3C. Simple object access protocol (soap). http ://www.w3.org/TR/SOAP, 2000.

[3] A. Ankolenkar, F. Hutch, and K. Sycara. Concurrent semantics for the web services specification language daml-s. In *The Fifth International Conference on Coordination Models and Languages*, 2002.

[4] J. Meseguer. Rewriting logic and maude : Concepts and applications. In *RTA 2000*, pages 1 – 26, 2000.

[5] C.A.R Hoare. *Communicating Sequential Processes*. Prentice Hall, 1986.

[6] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (wsdl). http ://www.w3.org/TR/wsdl, 2001.

[7] D. Fensel and C. Bussler. The web services modeling framework wsmf. In *1st meeting of the Semantic Web enabled Web Services*, 2002.

[8] Z. Gusessoum and J.P. Briot. From active objects to autonomous agents. *IEEE Concurrency*, 3(7) :68 – 76, 1999.

[9] Y. Gurevich and E. Börger, editors. *Specification and Validation Methods*, chapter Evolving Algebra Lipari Guide, pages 9–36. Oxford University Press, 1995.

[10] J. Charlet, M. Zacklad, G. Kassel, and D. Bourigault, editors. *Ingénierie des connaissances : évolutions et récents défis*. Eyrolles, 2000.

[11] K.D. Forbus. Qualitative process theory. *Artificial Intelligence*, (24) :85–167, 1984.

[12] T. Andreasen, J.F. Nilson, and H.E. Thomsen. Ontology-based querying. In *Conf. on Flexible Query Answering Systems (FQAS 2000)*, pages 15 – 26, 2000.

[13] M. Sintex and S. Decker. Triple an rdf query, inference, and transformation language. In *DDLP 2001*, 2001.