

LE CONTEXTEUR: UN MODEL COMPUTATIONNEL POUR LE CONTEXTE

Gaëtan Rey, Joëlle Coutaz

CLIPS-IMAG
385 rue de la Bibliothèque
38041 Grenoble cedex 9, France
{Gaetan.Rey, Joelle.Coutaz}@imag.fr

Résumé — Cet article propose le contexteur comme abstraction logicielle pour modéliser et calculer le contexte. Ici, le contexte doit être compris comme un vecteur d'état des variables observables. Une variable observable est une variable dont la valeur peut être acquise (à l'aide de capteurs) et ou être calculée par le système. Les contexteurs partagent une structure commune d'entrée sortie comprenant les canaux de contrôle ainsi que des méta-données permettant d'assurer et d'exprimer les QoS (par exemple), aussi bien que les propriétés communes comme la réflexivité, la précision, la stabilité et la rémanence. Ils peuvent être combinés suivant des graphes orientés ou être en capsulés dans des contexteurs de plus haut niveau. Nous montrons comment ils s'intègrent dans le modèle Arch.

1. Introduction

Le contexte est une « vieille connaissance ». La littérature indique un grand nombre de recherche basé sur la notion de contexte, typiquement dans la linguistique, l'IA, la vision par ordinateur, et l'IHM. Avec l'apparition du concept de l'informatique omniprésente (ubiquitous computing), les chercheurs redécouvrent cette notion. Des travaux développés sur le contexte [4], on peut tirer les leçons suivantes:

Leçon 1: Le contexte peut seulement être défini par rapport à un but. Dans cet article, le contexte est défini pour l'interaction homme machine.

Leçon 2: Le contexte est un espace d'information qui sert l'interprétation [7]. Dans notre travail, l'interprétation est effectuée par le système pour mieux aider les utilisateurs.

Leçon 3: Le contexte est un espace partagé d'information. Dans notre travail, le contexte définit un espace commun entre un système et un utilisateur. C'est donc une chose observable par l'homme.

Leçon 4: Le contexte est un espace ouvert d'information: il évolue. En conséquence, nous faisons une distinction entre une situation et la composition des situations, qui, alternativement, définissent un contexte.

Dans cet article, nous proposons le contexteur comme généralisation des processus perceptuels présentés dans [3], ainsi que des context-widget mis en application dans la context-toolkit [6]. Nous présentons le contexteur comme module pour le contexte se déployant dans système, et montrons comment des contexteurs peuvent être combinés en tant qu'unités informatiques plus riches. En conclusion, nous présentons comment les fédérations de contexteurs s'intègrent dans un modèle architectural faisant référence en IHM (le modèle Arch).

2. La notion de contexteur

Un contexteur est une abstraction logicielle qui fournit la valeur d'une variable du contexte. Après une rapide

description, nous en précisons les propriétés, puis

nous indiquons comment composer les contexteurs en unités fonctionnelles plus riches.

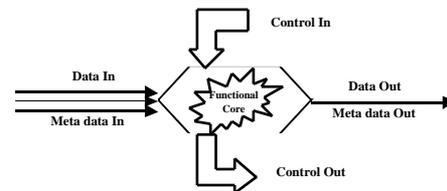


Figure 1 — Représentation graphique d'un contexteur

Le noyau fonctionnel d'un contexteur implémente une relation entre les variables du contexte. Les canaux d'entrée d'un contexteur sont de deux types:

Les données d'entrée correspondent aux variables du contexte qui sont employées comme entrée par le noyau fonctionnel du contexteur. Chaque donnée d'entrée est décorée de Méta-données qui expriment la qualité de la valeur d'entrée.

Le contrôle d'entrée correspond aux commandes reçues d'autres contexteurs pour changer les paramètres internes du contexteur. Ces paramètres concernent aussi bien le comportement fonctionnel du contexteur que le comportement non fonctionnel tel que les QoS.

Symétriquement, les canaux de sortie d'un contexteur sont de deux types:

Les données de sortie correspondent aux valeurs de des variables du contexte retourné par le contexteur. Comme pour les données d'entrée, les données de sortie sont décorées de Méta-données de sortie.

Le contrôle de sortie est employé par le contexteur pour envoyer des requêtes à d'autres contexteurs.

En outre, le canal des données d'entée reçoit des données d'un canal de données de sortie dont le type de données est compatible. Il en est de même pour le canal contrôle.

Les connexions entre les entrées et les sorties peuvent être statiques (c.-à-d., câblées par le

concepteur) ou dynamiques (c.-à-d., calculées au cours de l'exécution).

3. Composition des contexteurs

Les contexteurs peuvent se composer de deux manières : soit de manière hiérarchique, soit par encapsulation comme détaillé dans [2].

3.1. La composition hiérarchique

Les canaux de données d'entrée peuvent être reliés aux canaux conformes de données de sortie pour former une fédération. Deux canaux sont conformes s'ils ont des données du même type.

Comme représenté sur le schéma 2, le graphe orienté résultant forme une fédération dont la base est formée par des contexteurs élémentaires, et où les contexteurs du dessus fournissent à des applications des données contextuelles au niveau d'interprétation approprié.

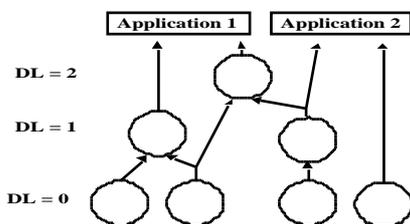


Figure 2. Une fédération de contexteurs.

Dans une fédération, un contexteur peut être caractérisé par un niveau de dépendance (DL) qui peut être exploité pour évaluer le coût d'une reconfiguration dynamique de la fédération.

Après avoir présenté la composition des contexteurs, voyons comment ces fédérations sont intégrées dans l'architecture d'une application.

4. Le modèle Arch augmenté

Le modèle Arch est un modèle conceptuel efficace pour concevoir la structure fonctionnelle d'un système interactif [1]. Avec PAC-Amodeus, nous avons étendu, raffiné, et exploité Arch de nombreuses manières en ce qui concerne le développement d'interfaces multimodales [5]. Semblable, dans l'esprit, à la proposition de Salber et selon l'ontologie présentée dans [3], nous proposons de prolonger Arch avec une branche dédiée au contexte (voir le schéma 3).

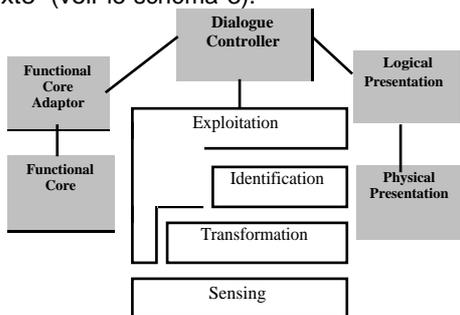


Figure 3. Le modèle Arch augmenté pour les applications sensibles au contexte.

La couche de capture (Sensing) est constituée par les contexteurs élémentaires, c.-à-d., les contexteurs qui encapsulent des capteurs physiques. Elle est au contexte ce que le composant physique de présentation est aux interfaces utilisateur classiques.

La couche de transformation, qui correspond aux fédérations de contexteurs, fournit des informations contextuelles au bon niveau d'abstraction. Elle est au contexte ce que le composant logique de présentation est aux interfaces utilisateur classiques.

La couche d'identification détecte les changements de situation et de contexte, et identifie la situation et le contexte actuel. Cette couche peut être mise en application en utilisant une approche de type « black board » [7].

La couche exploitation agit en tant qu'adaptateur entre le contrôleur de dialogue et la partie de calcul du contexte. Pour des raisons d'exécution, elle peut sauter l'identification et échanger l'information directement avec les couches transformation et capture.

Comme dans Arch, le méta-modèle « slinky » s'applique : quelques couches peuvent ne pas exister, et les fonctions peuvent se décaler entre les couches.

5. Conclusion

Les modèles et les principes présentés dans cet article sont basés sur notre expérience du développement en interaction multimodale comme en vision par ordinateur. Par analogie avec les interacteurs, les contexteurs sont motivés par les avantages de la technologie distribuée orientée objet. Nous mettons en application actuellement les contexteurs dans le projet européen GLOSS en utilisant une approche P2P (égal à égal). Comme discuté dans la section 4, nous ne favorisons pas un paradigme uniforme simple pour le calcul de contexte. Au lieu de cela, nous proposons que les niveaux plus élevés de la nouvelle branche contextuelle de Arch utilise des techniques d'IA telles que le « black board » (tableau noir).

6. References

- [1] Arch, UIMS Tool Developers' Workshop. A meta-model for runtime architecture of an interactive system. SIGCHI Bulletin, 24(1):32{37, 1992.
- [2] J. Coutaz, G. Rey Foundations for a theory of contextors. In Proc CADUI02, ACM Publ., 2002, pp. 283-302.
- [3] J. L. Crowley, J. Coutaz, G. Rey. Perceptual Components for Perceptual Computing. In Proc. Ubicom (Ubiquitous Computing)2002.
- [4] Moran, T. P. and Dourish, J. P. (editors). Special Issue on Context-Aware Computing. Human Computer Interaction, Volume 16, Numbers 2-4. Erlbaum.
- [5] L. Nigay J. Coutaz, A Generic Platform for Addressing the Multimodal Challenge, Conférence internationale avec comité de lecture : Proceedings of CHI'95, ACM New York Publ., Denver, May 1995, pp. 98-105.

- [6] D. Salber, A.K. Dey, G. Abowd. The Context Toolkit: Aiding the development of context-enabled Applications. In Proc. CHI99, ACM Publ., 1999, pp. 434-441.
- [7] T. Winograd. Architecture for Context, Human Computer Interaction, Lawrence Erlbaum Ed., Vol. 16, pp. 401-419