Construction collaborative d'un dictionnaire multilingue : le projet Papillon

Mathieu Mangeot-Nagata[†], **Slaven Bilac***[*] & David Thevenin[†],

[†], NII, Hitotsubashi 2-1-2-1913 Chiyoda-ku
Tokyo 101-8430 Japon
Tel. : 03-4212-2672 - Fax. : 03-3556-1916
E-mail : {mangeot,thevenin}@nii.ac.jp

* Tokyo Institute of Technology 2-12-1
Ookayama, Meguro-ku,
Tokyo 152-8552 Japon
E-mail : sbilac@cl.cs.titech.ac.jp

Résumé :

Le projet Papillon est un projet de construction collaborative d'un dictionnaire multilingue sur la Toile comprenant entre autres le français et le japonais. Les utilisateurs peuvent contribuer directement sur le serveur en ajoutant de nouvelles données ou en corrigeant des erreurs éventuelles. Les contributions sont ensuite stockées dans l'espace de l'utilisateur puis révisées par un spécialiste avant leur intégration définitive dans la base. Les données ainsi produites sont disponibles publiquement et libres de droits. Le but étant d'appliquer le paradigme de construction coopérative de LINUX à celle d'un dictionnaire d'usage multilingue à large couverture.

Ce projet, lancé par une coopération franco-japonaise a démarré il y a 4 ans. Les partenaires actuels travaillent sur les langues suivantes : anglais, chinois, français, japonais, lao, malais, thai, et vietnamien. Un serveur en ligne disponible à l'adresse http://www.papillon-dictionary.org/ permet de consulter les données existantes ainsi que de les éditer en ligne.

Dans cet article, nous exposons d'abord une méthode d'aide à la recherche spécifique au japonais et ensuite, nous détaillons le fonctionnement de l'éditeur générique de dictionnaires basé sur la description de la structure des articles.

Abstract :

The Papillon project is a collaborative project for building a multilingual dictionary on the Web that includes, among others, entries in French and Japanese. The users can contribute directly on the server by adding new data or by correcting existing errors. The contributions are stored in the user space until checked by a specialist before being fully integrated into the database. The resulting data is then publicly available and freely distributable. The aim is to apply the LINUX cooperative construction paradigm to the building of a broad coverage multilingual dictionary.

This project, launched by a French-Japanese partnership, started 4 years ago. The partners are now working on the following languages: English, Chinese, French, Japanese, Lao, Malay, Thai and Vietnamese. An online server available at http://www.papillon-dictionary.org/ allows users to lookup the existing data before editing it online.

In this paper, we first explain a dictionary access module for Japanese allowing kanji reading errors and finally, we detail the generic editor for dictionary entries based on the description of the entries structure.

Introduction :

Le projet Papillon est un projet de construction collaborative d'un dictionnaire multilingue sur la Toile comprenant entre autres le français et le japonais. La macrostructure du dictionnaire est une structure pivot avec un volume monolingue pour chaque langue et un volume pivot au centre. Les articles monolingues sont reliés entre eux par des liens interlingues regroupés dans le volume pivot. La microstructure des articles est basée sur la lexicographie explicative et combinatoire issue de la théorie sens-texte (travaux d'Igor Mel'cuk et ses collègues). La théorie étant indépendante des langues, la structure des articles sera la même pour toutes les langues.

Les utilisateurs peuvent contribuer directement sur le serveur en ajoutant de nouvelles données ou en corrigeant des erreurs éventuelles. Les contributions sont ensuite stockées dans l'espace de l'utilisateur puis révisées par un spécialiste avant leur intégration définitive dans la base. Les données ainsi produites sont disponibles publiquement et libres de droits. Le but étant d'appliquer le paradigme de construction coopérative de LINUX à celle d'un dictionnaire d'usage multilingue à large couverture.

Ce projet a démarré il y a 4 ans par Emmanuel Planas, François Brown de Colstoun et Mutsuko Tomokiyo. Il a été lancé par une coopération franco-japonaise entre le CLIPS à Grenoble et l'Institut National d'Informatique à Tokyo. Après quatre séminaires (dont 2 à Tokyo, 1 à Grenoble et un à Sapporo), de nombreux partenaires se sont manifestés et ont souhaité rejoindre le projet : Jim Breen, auteur du dictionnaire EDICT (Université Monash, Australie), Francis Bond (NTT, Keihanna), Yves Lepage (ATR, Keihanna), Ulrich Appel, auteur du dictionnaire allemand-japonais WaDoKuJiten, Jean-Marc Desperrier, responsable de l'adaptation au français du dictionnaire EDICT, l'université Kasetsart et le NECTEC (Bangkok, Thaïlande), l'Universiti Sains Malaysia (Penang, Malaisie), les universités de Da Nang et de Hanoi (Vietnam), etc.

Actuellement, les langues couvertes sont l'allemand, l'anglais, le chinois, le français, le japonais, le lao, le malais, le thaï et le vietnamien. Des discussions sont en cours concernant les langues indiennes et le coréen. Un serveur en ligne disponible à l'adresse http://www.papillon-dictionary.org/ permet de consulter les données existantes ainsi que de les éditer en ligne.

Nous exposons d'abord une méthode d'aide à la recherche spécifique au japonais et ensuite, nous détaillons le fonctionnement de l'éditeur générique de dictionnaires basé sur la description de la structure des articles.

1. FOKS : un module d'aide à la consultation de mots japonais avec kanjis

1.1. Problématique

La consultation de mots inconnus comprenant des kanjis est particulièrement difficile en japonais à cause de la complexité du système d'écriture. Le nombre élevé des kanjis (1 945 prescrits par le gouvernement por usage quotidien et plus de 3 000 apparaissant dans les journaux et publications officielles) présente en lui-même un défi, mais le problème est sérieusement compliqué par le fait que chaque caractère peut avoir (et c'est souvent le cas) plusieurs lectures qui de plus sont souvent sans rapport entre elles. Par exemple, le kanji 発 peut se lire *hatsu* ou *ta(tsu)*, de la même manière, le kanji 表 peut se lire *omote*, *hyou* et *arawa(reru)*. Basé sur une combinatoire simple, le mot 発表 (exposé) peut se lire de 6 manières différentes, et lorsque l'on considère les variations phonologiques, ce nombre peut devenir bien plus important.

1.2. Le système FOKS

Le système FOKS (Forgiving Online Kanji Search) de Slaven Bilac [1,2] permet aux apprenants du japonais (et même aux japonais) de rechercher un mot en fonction d'une possible lecture possible, mais pas nécessairement correcte, des kanjis. C'est une avancée importante par rapport aux systèmes précédents qui ne gèrent pas les erreurs dues à une lecture erronée des kanjis.

Tout d'abord, les lectures possibles de chaque kanji sont calculées et les variations phonologiques sont ajoutées au résultat, puis une probabilité est associée à chaque lecture. Ensuite, en combinant ces probabilités avec la fréquence d'apparition de ces lectures dans un corpus, une mesure de plausibilité est calculée pour chaque entrée de dictionnaire.

En réponse à une lecture donnée par un utilisateur, la plausibilité de chaque entrée parrapport à cette lecture est calculée puis une liste de candidats possibles triée selon la plus forte probabilité est affichée.

1.3. Conclusion

Le système FOKS a été intégré dans le serveur Papillon. Il peut être utilisé pour la consultation de mots japonais.

Une première extension envisagée du système est la gestion d'autres types d'erreurs non encore traitées à l'heure actuelle comme la similarité graphique entre les kanjis.

Une autre extension du système serait d'étendre la méthode à d'autres langues et d'autres utilisateurs. Les japonais apprenants du français ont par exemple beaucoup de mal à distinguer les lettres "r" et "l". Il serait intéressant de pouvoir gérer ce type de problèmes lors de la consultation de mots français par des japonais.

2. Éditeur générique en ligne d'articles de dictionnaires

2.1. Spécifications de l'éditeur

L'état de l'art de la thèse de Mathieu Mangeot [5] montre que les méthodes existantes d'édition d'articles de dictionnaire comportent toutes de sérieuses limitations. Pour que le projet Papillon soit viable, il était indispensable de trouver une solution satisfaisante qui permette d'éditer en ligne des articles de dictionnaire ayant une structure complexe d'une part et susceptibles d'évoluer d'autre part.

La solution de développer un éditeur ad-hoc avec la technique du formulaire HTML est intéressante au premier abord car rapide à implémenter. Mais cette solution à long terme devient très laborieuse car il est nécessaire de modifier le code source à chaque changement de structure d'une part et de développer autant d'éditeurs qu'il y a de structures différentes d'autre part.

Si l'on veut éditer des articles en ligne, il faut utiliser un formulaire HTML. Nous remédions à ses limitations en créant des interacteurs supplémentaires plus compliqués à partir des interateurs existants et en effectuant tous les calculs sur le serveur.

2.2 Utiliser ArtStudio, outil de génération d'interfaces

La thèse de David Thevenin [9] porte sur l'adaptation des interfaces homme-machine pour différentes cibles (plateforme, utilisateur, environnement). L'outil ArtStudio développé pendant sa thèse permet de spécifier une interface et de produire l'application correspondant en fonction de différentes plateformes cibles. Nous avons donc eu l'idée de coopérer pour pouvoir utiliser l'outil ArtStudio de manière à générer des interfaces pour notre éditeur basées sur les données à traiter : les articles de dictionnaire.

Le serveur Papillon est programmé avec Enhydra, serveur Web d'objets dynamiques java. Les données sont stockées sous forme d'objets XML dans une base de données SQL : PostgresQL.

L'outil ArtStudio est entièrement programmé en Java. Pour son intégration dans l'environnement Papillon/Enhydra, nous avons créé une archive java de ArtStudio pour que les codes restent indépendants.

Avant le lancement de notre projet d'éditeur, ArtStudio permettait de générer du code pour applications Java et Waba pour téléphones portables. Il a donc fallu rajouter une nouvelle cible dans l'outil de génération de code pour pouvoir générer du code CGI/HTML qui puisse s'interfacer avec Papillon.

L'autre modification importante a été de baser la génération de l'interface non plus à partir des tâches à réaliser mais à partir des données manipulées : les articles de dictionnaire.

2.3. Modèles de description de l'interface utilisés par ArtStudio

Pour générer une interface, ArtStudio s'appuie sur cinq modèles de description : le modèle des tâches, des concepts, des instances, des interacteurs et de la plateforme cible.

2.3.1. Modèle des tâches

Le modèle des tâches décrit sous la forme d'un graphe, les tâches qui seront réalisables par l'interface à générer. Dans notre cas, nous avons deux modèles, celui pour l'édition d'un article et celui pour visualisation simple d'un article. Dans les deux cas, ils s'avèrent très simples.

2.3.2. Modèle des concepts

Le modèle des concepts décrit la structure des données qui seront manipulées par l'interface graphique. Ce modèle est construit à partir des descriptions en Schémas XML de la microstructure du dictionnaire. À partir de ce modèle, ArtStudio "trouve" les interacteurs et construit l'interface. Les figures 3 et 5 montrent un exemple.

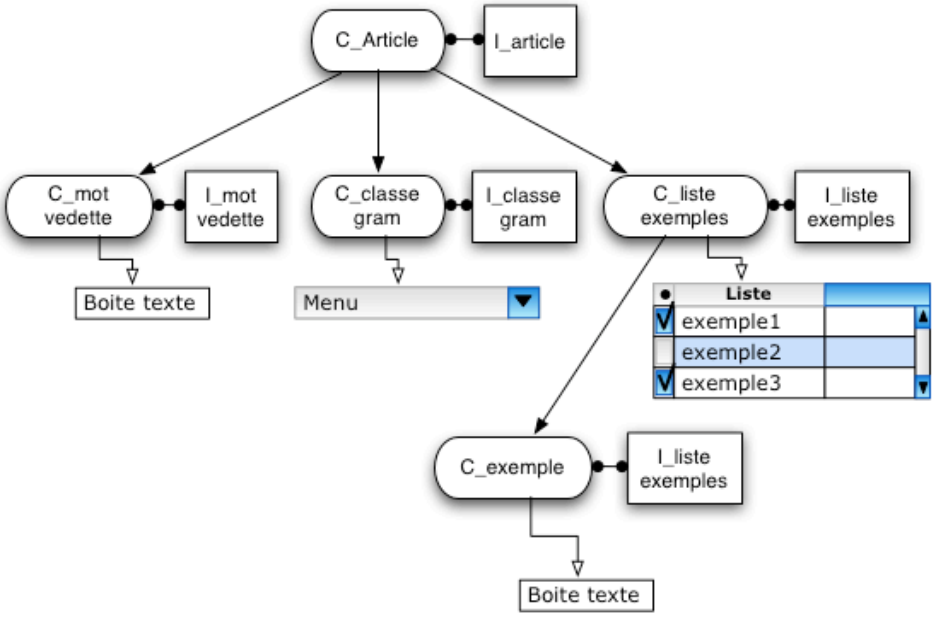

Figure 3 : Représentation du modèle des concepts avec les interacteurs associés

2.3.3. Modèle des instances

Le modèle des instances décrit les instances de concept manipulés par l'interface et le graphe de dépendance entre ces instances. Par exemple nous avons le concept "Article" et une instance de concept "scientifique" (C.f. figure 4).

Ce modèle est décrit en phase de spécification. Chaque instance sera effectivement instanciée au court de l'exécution de l'interface, avec les données modifiables contenues dans l'article de dictionnaire que l'on veut éditer (C.f. figure 4.). Si l'éditeur est lancé sans article en paramètre, un article vide est utilisé à la place pour instancier le modèle.

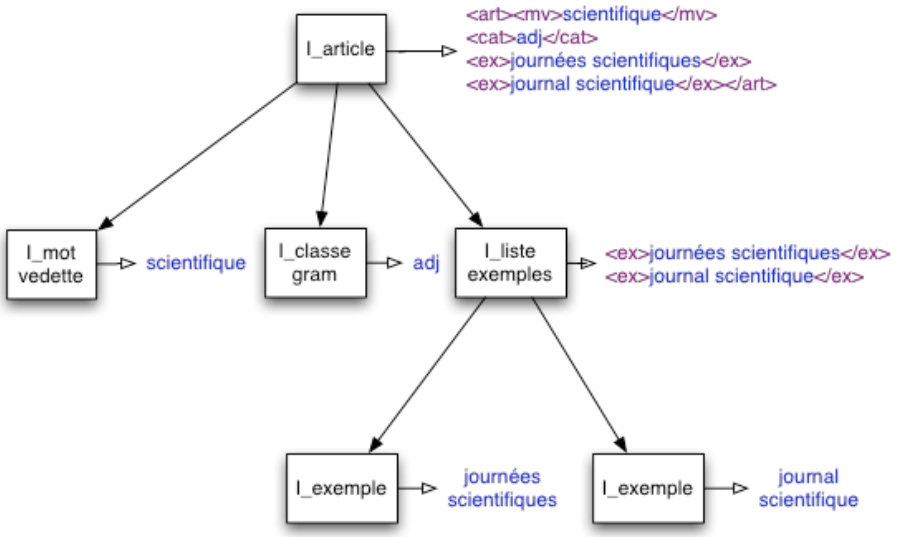

Figure 4 : Représentation du modèle des instances et de leur valeur

2.3.4. Modèle des interacteurs

Ce modèle décrit les interacteurs (Widget ou Interactor en anglais) qui sont présents sur la boîte à outils graphique de la plateforme visée. Dans ce projet, il s'agit des interacteurs présents dans les

formulaires HTML (textBox, comboBox, popup menu, bouton, checkBox, radioButton) et les balises HTML. En faisant des combinaisons, nous construisons des interacteurs plus sophistiqués, par exemples pour éditer une liste d'exemple (C.f. figure 5).

2.3.5. Modèle de la plateforme cible

Ce modèle décrit la plateforme pour laquelle nous allons produire des interfaces pour Papillon. Ici il s'agit de décrire les caractéristiques graphiques de l'ordinateur "type" qui sera utilisé pour naviguer dans le dictionnaire et éditer des articles.

2.4. Processus d'édition d'un article

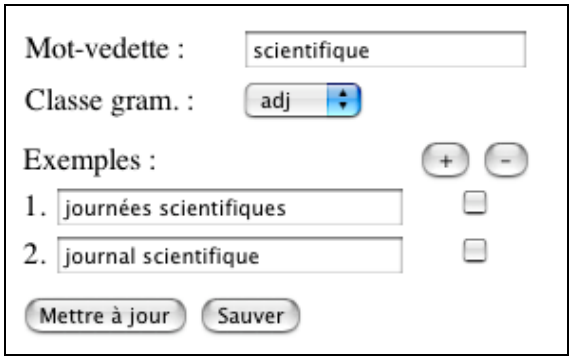

Figure 5 : Capture d'écran du formulaire HTML

Le serveur Papillon/Enhydra permet de stocker des objets Java au cours d'une session utilisateur. Lorsqu'un utilisateur se connecte au serveur, une session est créée et l'utilisateur est identifié grâce à un cookie. Si l'utilisateur lance l'éditeur, les objets java utilisés seront conservés pendant la durée de la session.

Lorsque l'éditeur est lancé avec un article en paramètre, les modèles correspondant au volume de l'article sont chargés en mémoire, puis le patron de l'article est instancié avec les données contenues dans l'article. Les modèles ainsi que le patron instancié, sont stockés dans la session puis ensuite, le résultat est affiché dans un formulaire HTML (figure 5).

Ensuite, à chaque modification de l'utilisateur (principalement l'ajout ou la suppression d'un élément dans une liste), le formulaire renvoie les données HTML au serveur par un mécanisme de CGI. Celui-ci actualise ensuite le patron stocké dans la session puis réaffiche le résultat modifié dans le formulaire HTML.

En fin de session, l'article modifié est extrait des données de la session puis stocké dans la base de données.

Conclusion

Les deux premières années du projet ont servi à déterminer les choix théoriques des structures du dictionnaire Papillon qui comprend une macrostructure pivot avec des liens interlingues et une microstructure très détaillée basée sur la théorie sens-texte.

La troisième année a essentiellement été consacrée à l'implémentation de la plateforme Papillon et à la collecte de ressources existantes (disponibles à la consultation).

La mise au point de l'éditeur générique a pris environ un an entre les premières spécifications et la mise à disposition d'une version complète et fonctionnelle de l'éditeur sur le serveur Papillon. Nous souhaitons compléter ce travail en suivant plusieurs directions.

La prochaine étape est celle de la construction d'un squelette de dictionnaire en récupérant et intégrant les données des dictionnaires préalablement collectés.

Nous pourrons ensuite ouvrir enfin le projet au grand public afin que les utilisateurs puissent contribuer en corrigeant les données existantes et en ajoutant des données manquantes.

Références

[1] Slaven Bilac (2002) *Intelligent Dictionary Interface for Learners of Japanese*. Masters's thesis, Tokyo Institute of Technology, 56 p.

[2] Slaven Bilac, Timothy Baldwin, and Hozumi Tanaka (2002) *Bringing the Dictionary to the User: The FOKS System*. In Proc. of the 19th International Conference on Computational Linguistics (COLING2002), pp 89--95.

[3] Mutsuko Tomokiyo, Mathieu Mangeot & Emmanuel Planas (2000) *Papillon : a Project of Lexical Database for English, French and Japanese, using Interlingual Links* JST'2000 Journées Science et Technologie de l'ambassade de France au Japon, Tokyo, Japon, 13-14 novembre 2000, 3p.

[4] Mathieu Mangeot & Gilles Sérasset (2001) *Projet Papillon : architecture du serveur Web et structure des articles*. JST'2001 Journées Science et Technologie, National Olympic Memorial Youth Center, Tokyo, Japon, lundi 19 & mardi 20 novembre, vol 1/1, pp 149-150.

[5] Mathieu Mangeot (2001) *Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue*. Thèse de nouveau doctorat, Spécialité Informatique, Université Joseph Fourier Grenoble I, jeudi 27 septembre 2001, 280 p.

[6] Mathieu Mangeot (2002) *Projet Papillon : intégration de dictionnaires existants et gestion des contributions*. JST-2002 Journées Science et Technologie, National Olympic Memorial Youth Center, Tokyo, Japon, 17-19 novembre, vol 1/1, pp 64-65.

[7] Mathieu Mangeot-Nagata, Gilles Sérasset & Mathieu Lafourcade (2003) *Construction collaborative de données lexicales multilingues, le projet Papillon*. Revue TAL Traitement Automatique des Langues, édition spéciale, 30 p. (automne 2003).

[8] Gilles Sérasset & Mathieu Mangeot (2001) *Papillon Lexical Database Project: Monolingual Dictionaries & Interlingual Links*. Proc. NLPRS'2001, Hitotsubashi Memorial Hall, National Center of Sciences, Tokyo, Japan, 27-30 November 2001, vol 1/1, pp. 119-125.

[9] David Thevenin (2001) *L'adaptation en Interction Homme-Machine : le cas de la plasticité*. Thèse de nouveau doctorat, Spécialité Informatique, Université Joseph Fourier Grenoble I, décembre 2001, 238 p.