

Nouvelles de L^AT_EX₃

Troisième édition, janvier 2010

Bonne année

Nous sommes heureux de vous présenter l'édition des vacances d'hiver du « compte-rendu d'activités » de l'équipe L^AT_EX₃.

Développements récents

Deux sorties importantes de code L^AT_EX₃ ont été faites durant les six derniers mois. Dans le répertoire `xpackages`¹ du CTAN, vous trouverez deux éléments intéressants :

- une version revue de `xparse` ;
- le nouveau module `xtemplate`, une ré-implémentation de `template` avec une nouvelle syntaxe.

Nous remercions particulièrement Joseph Wright, qui s'est chargé essentiellement seul de l'implémentation de ces deux modules (avec beaucoup de remarques et de retours d'autres membres de l'équipe et de participants de la liste L^AT_EX-L).

Ces deux modules s'adressent à l'auteur de modules L^AT_EX qui souhaite définir des commandes pour l'auteur du document et des interfaces pour le metteur en pages, avec des outils de haut niveau.

xparse. Ce module permet de définir des commandes de niveau document complexes, avec toutes sortes d'arguments optionnels. Pensez à la façon dont `\newcommand` permet de définir une commande avec un argument optionnel : `xparse` en est une généralisation.

xtemplate. Ce module demande plus d'explications. Le but de `xtemplate` est de séparer la structure logique d'un document de sa représentation visuelle. Pour cela, des modèles (*templates*) sont construits, qui fournissent une façon de composer une partie du document en fonction d'un certain nombre d'arguments. Pour changer l'apparence d'une partie d'un document, il suffit de changer l'instance utilisée pour ce modèle, sans avoir besoin ni de modifier le balisage du document lui-même, ni de toucher aux macros internes de L^AT_EX.

Des modules L^AT_EX_{2 ϵ} comme `geometry` ou `titlesec` fournissent déjà une interface permettant de paramétrer certains éléments d'un document. Par exemple, on peut utiliser `titlesec` pour modifier la mise en pages de `\section`, en modifiant les paramètres qui la régissent avec `\titleformat` et `\titlespacing`. Dans un sens, ces modules fournissent un modèle pour des éléments spécifiques du document, et quelques commandes permettant d'en créer des instances. Cependant, dès qu'il n'est plus possible d'obtenir la mise en page désirée avec un des ces modules, on se retrouve désarmé : il faut soit recourir à de la programmation de bas niveau, soit trouver un autre module de haut niveau, qui ne manquera pas d'avoir ses

propres conventions et commandes de manipulation du modèle.

Le module `xtemplate` peut être vu comme une généralisation de ces idées. Il fournit une interface unifiée pour définir et manipuler des modèles pour toutes sortes d'éléments du document, et surtout, fournit une interface unifiée pour définir une instance d'un modèle.

Ainsi, l'activité de définir ou modifier une classe de document pour une mise en pages donnée se réduit à choisir quels éléments logiques (`\chapter`, `\section`, `\footnote`, etc.) la classe doit offrir, puis sélectionner un modèle (nommé) approprié pour chacun d'entre eux, et enfin de définir des instances en choisissant les valeurs de leurs paramètres. S'il n'est pas possible d'obtenir la mise en pages voulue avec un modèle donné, on peut choisir un modèle différent pour le même élément du document.

Il n'y a pas besoin de programmer, sauf si aucun des modèles disponibles ne convient pour la mise en pages prévue. C'est à ce moment qu'un programmeur L^AT_EX doit construire un nouveau modèle qui rend possible cette mise en pages. Une fois que c'est fait, le nouveau modèle vient s'ajouter à la collection que les utilisateurs et éditeurs peuvent utiliser pour définir ou modifier sans peine des mises en pages.

Ceci n'est qu'un bref aperçu de la complexité du module lui-même ; vous pouvez en apprendre plus dans sa documentation. Nous avons essayé de documenter clairement `xtemplate`, et nous sommes très curieux de savoir en retour si ces idées vous parlent.

En complément de l'arrivée de `xtemplate`, précisons que l'ancien module `template` va bientôt disparaître. Il n'y a, à notre connaissance, qu'un module sur le CTAN qui utilise `template`, à savoir `xfrac`, et les membres de l'équipe L^AT_EX₃ sont en train de le modifier pour utiliser `xtemplate`. Si vous avez du code personnel qui utilise `template`, merci de nous le faire savoir !

Plans pour la suite

Après cette annonce de la mise à jour de `xparse` et de l'arrivée de `xtemplate`, la prochaine étape de développement sera d'utiliser des deux systèmes d'autres composant de `xpackages`, de façon à confronter nos idées de départ à la pratique, et à voir si ces modules sont à la hauteur de nos attentes.

Modules sur la liste

xhead. La première tâche sera de créer un nouveau membre des `xpackages` (probablement appelé `xhead`) pour la composition des titres de sections et autres subdivisions du document. Les titres de section sont un des sujets les

1. <http://mirror.ctan.org/tex-archive/macros/latex/contrib/xpackages/>

plus difficiles à traiter ; ce travail devrait donc être suffisamment exigeant pour `xtemplate` pour qu'on puisse juger si sa forme actuelle est suffisante pour la plupart des usages. Pour l'instant, rien n'a été publié, mais nous annoncerons les développements en temps voulu sur la liste LATEX-L².

galley. Nous voulons aussi faire subir à `galley` le même traitement que `xparse` et `xtemplate`, c'est-à-dire que nous avons déjà une vieille implémentation (en fait, deux) qui a besoin d'un peu de travail avant d'être prête à publier sur le CTAN.

Le module `galley` est utilisé pour insérer du matériel dans la liste verticale pendant la composition, mais avant qu'un saut de page ne survienne. Vu qu'il travaille à bas niveau, il est important que ce module soit robuste avant d'écrire des modèles de mise en pages de plus haut niveau.

Nous aurons à faire face au fait que, pour arriver aux meilleurs résultats, `galley` ne peut pas être utilisé avec du code $\text{\LaTeX} 2_{\epsilon}$. Ceci risque de restreindre son utilité, et nous déciderons peut-être de réduire nos ambitions concernant les fonctionnalités pour permettre une meilleur interopérabilité avec les documents et modules existants. Il reste encore du travail avant que nous soyons prêts à choisir entre ces deux possibilités.

2. Voir <http://www.latex-project.org/code.html> pour les détails.