Cahiers GUTenberg

ÉDITION STRUCTURÉE ET NON STRUCTURÉE D'EXPRESSIONS MATHÉMATIQUES DANS THOT

Christian Lenne

Cahiers GUTenberg, nº 25 (1996), p. 25-32.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1996___25_25_0>

Édition structurée et non structurée d'expressions mathématiques dans Thot

Christian Lenne

Unité de Recherche INRIA Rhône-Alpes,
ZIRST – 655 Avenue de l'Europe
38330 Montbonnot Saint-Martin, France

Résumé. L'écriture de documents dans le monde scientifique ou dans les industries de pointe nécessite la possibilité d'exprimer des formules mathématiques. Les outils de traitement de textes du marché offrent peu cette fonctionnalité ou, au contraire, sont spécialisés pour cela. Nous exposons ici notre approche dans le cadre de l'éditeur de documents structurés Thot.

1. Introduction

Deux approches se sont dessinées dans l'évolution des systèmes de production de documents : l'approche compilée avec le développement de langages de description de documents comme Troff ou LaTeX et l'approche interactive comme dans Framemaker ou Word. Ces derniers tiennent peu compte de la structure des documents, ce qui limite les fonctionnalités, même si cela donne des résultats de bonne qualité à l'impression. En particulier, le traitement des expressions mathématiques reste une faiblesse de ces produits alors que les langages comme LaTeX sont puissants et généraux.

L'introduction de structure dans les documents offre de nombreux avantages (présentation uniforme, fonctions de recherche, conversions de formats, etc.), mais peut également poser des problèmes lorsque l'entité à décrire est complexe et que le dialogue outil-utilisateur est guidé par la structure. Il faut, dans ce cas, offrir des mécanismes de guidage plus lâche afin d'éviter des changements incessants de dispositif de saisie. C'est ce que nous présentons dans la section 3 de ce document.

Avant toute chose, nous faisons une présentation rapide du système de production de documents structurés Thot.

2. L'éditeur Thot

Thot est la dernière évolution du logiciel Grif développé à l'INRIA et à l'IMAG. C'est un système de production de documents structurés qui permet de créer, de modifier et de consulter de façon interactive des documents qui respectent des modèles [2–5]. Grâce à ces modèles, on obtient des documents homogènes et l'utilisateur peut se concentrer sur l'organisation et le contenu des documents qu'il traite, sans s'occuper de formatage ou de typographie, ces fonctions étant prises en charge par le système. Thot effectue également d'autres traitements pour l'utilisateur, comme les numérotations, le maintien des références croisées, la gestion des index, etc.

Thot est un système intégré et extensible. Il permet de traiter avec le même outil et dans le même document non seulement du texte structuré, mais aussi des graphiques, des tableaux complexes, des expressions mathématiques, etc. Cette liste n'est pas exhaustive : les utilisateurs peuvent ajouter d'autres types d'informations, en définissant les modèles adéquats.

Thot est un système ouvert. Il peut échanger des documents avec d'autres systèmes, par l'intermédiaire d'un outil d'exportation paramétrable. Il peut aussi s'intégrer dans d'autres applications, à travers son interface de programmation et son mécanisme d'appels externes.

La structuration des documents permet d'élargir la gamme des traitements automatiques possibles : extraction d'information, transformation, etc. L'éditeur Thot a justement été conçu comme un système ouvert et extensible pour permettre le développement d'applications fondées sur les documents.

Dans Thot, un document est représenté par sa *structure logique* c'est-à-dire son organisation en éléments comme des titres, chapitres, sections, paragraphes, notes, figures, etc. Le texte ainsi que d'autres éléments de base (symboles, graphiques, images) constituent les éléments terminaux de ces structures hiérarchiques.

La structure logique est contrainte par un *schéma de structure*, qui spécifie principalement les types des éléments utilisables et les relations qui peuvent les relier. Chaque type de document est défini par un schéma de structure et il est possible de définir de nouveaux types de documents grâce au langage S.

Le langage S est très similaire au langage SGML dans ses principes. Il diffère de celui-ci de par sa syntaxe et sur quelques points liés au fait que le langage S vise la description de la structure de documents manipulés interactivement alors que le langage SGML vise essentiellement l'échange de documents structurés entre applications.

Au niveau le plus bas de la structure d'un document, on trouve les types de base que sont le texte, les éléments graphiques (cercle, rectangle, trait, etc.), les symboles mathématiques et les images (Bitmaps, GIF, EPSF). Le langage S offre plusieurs constructeurs

pour définir des types d'éléments plus complexes à partir de ces éléments de base. Ces constructeurs sont : la liste, l'agrégat, le choix, ou la référence.

La structure logique (figure 1) d'un document est construite par l'éditeur Thot, sous le contrôle de l'utilisateur. L'éditeur assure que chaque document qu'il traite respecte le modèle de son schéma de structure et, pour cela n'autorise que les opérations qui conduisent à une structure logique conforme au schéma de structure. Il utilise également le schéma de structure pour guider l'utilisateur ou pour engendrer automatiquement certaines parties de la structure du document.

Par ailleurs, un langage, T, permet de traduire un document dans un autre format, par exemple en SGML ou en LaTeX, ce qui fait de Thot un excellent système intéractif de saisie pour LaTeX ou HTML [1] !

3. Description d'une formule

L'édition structurée trouve ses limites sur le plan ergonomique lorsque les entités de la structure sont nombreuses et contiennent peu de caractères, comme c'est le cas dans les formules mathématiques. Les saisies claviers sont alors nombreuses mais en faible quantité, ce qui entraîne des va-et-vients incessant entre le clavier et la souris. L'idée est donc de fournir la possibilité de définir une formule mathématique à l'aide d'un seul dispositif de saisie. Nous avons choisi pour cela de privilégier le clavier par rapport à la souris. Ainsi donc, les formules mathématiques seront décrites à l'aide d'un langage. Nous avons adopté celui qui semble être un standard dans notre communauté scientifique, à savoir LaTeX et plus précisemment les environnements *math*, *displaymath* et *array* [6].

Cette approche nous a amené à construire un compilateur de ce langage. L'invocation du compilateur de formules est transparente à l'utilisateur, car celui-ci dispose d'une commande d'insertion de formule LaTeX qui déroute le dialogue dans un formulaire simple lui permettant de saisir sa formule. Un exemple de cette fenêtre de saisie est donné en figure 2.

Lorsque l'utilisateur choisit d'insérer la formule, celle-ci est soit insérée sur la même ligne si au moment de la saisie il a sélectionné le mode *math* (texte compris entre $ et $ ou \(et \)), soit comme ici dans un paragraphe centrée dans le mode *displaymath*, ce qui donne :

$$I_i^{(m)} = -\int_{\frac{\pi}{2}}^{\pi} \int_0^{2\pi} |V| \cos\theta \, \sin\theta \, (H_i^{(m)}) \, d\varepsilon \, d\theta,$$

En analysant les macros associées aux formules mathématiques, nous avons constaté que certaines macros étaient redondantes avec les éléments des schémas déjà définis,

1. Une nouvelle version de Thot est en fait un véritable *authoring system+browser* [1]

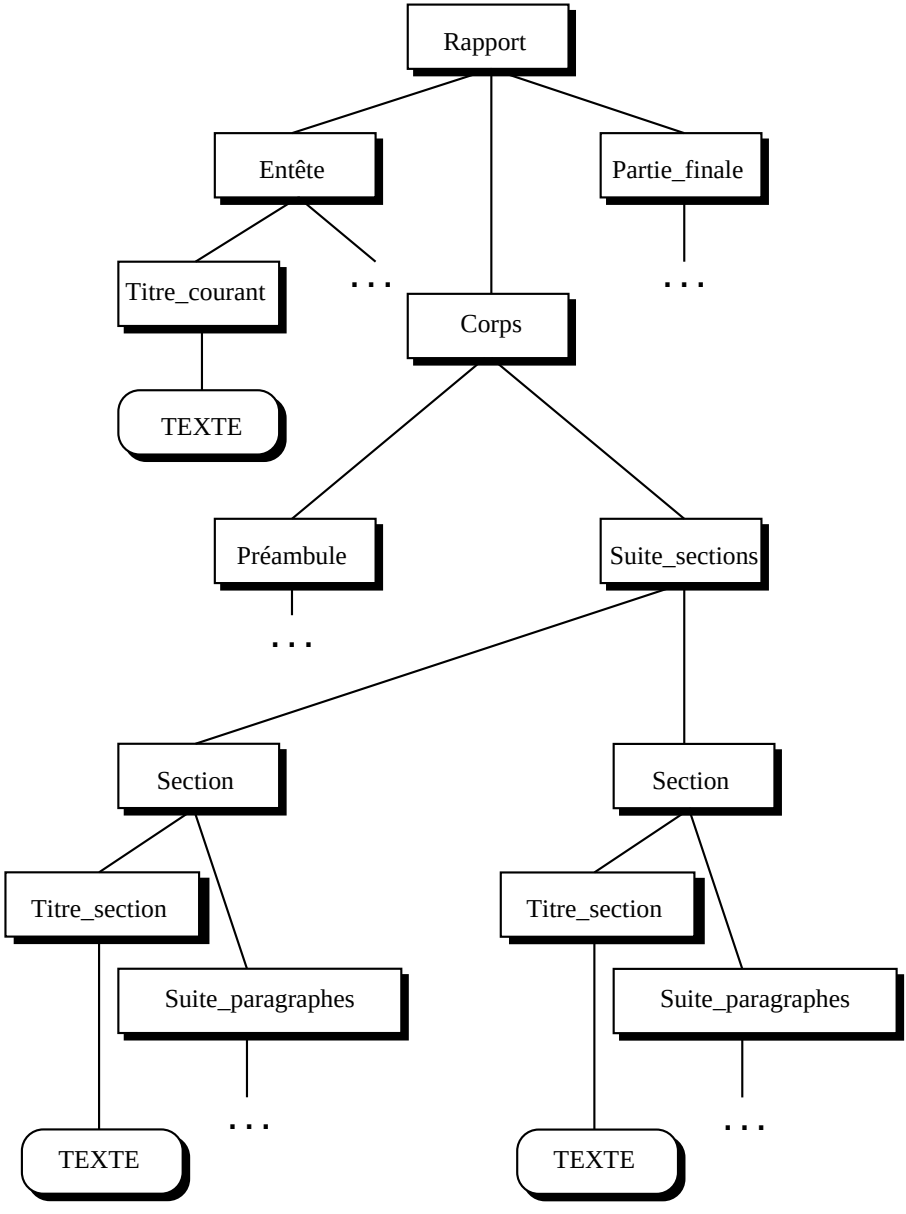

Figure 1 – *la structure logique d'un document*

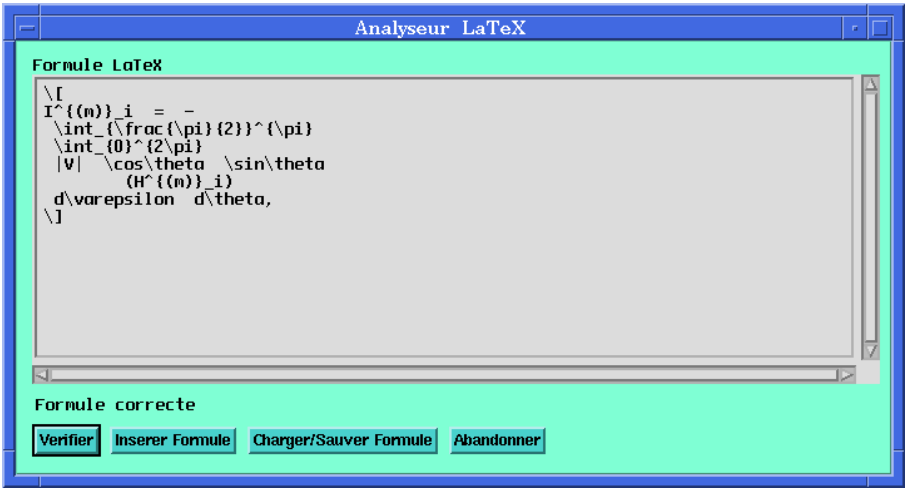

FIGURE 2 – *Fenêtre de saisie d'une formule*

comme par exemple les tableaux fortement structurés. De tels éléments n'ont pas été pris en considération au niveau LaTeX, car cela n'apportait rien (bien au contraire) et qu'il était beaucoup plus simple de les définir à l'aide de la souris.

Des trois environnements LaTeX traités, nous avons retenu une vingtaine d'éléments structurés comme :

- la fraction ($\left(\frac{A+B}{C}\right)$),
- les indices et exposants (X_i^2),
- les racines ($\sqrt[3]{8}$),
- les intégrales
$$\left\{ \int_0^n f, \int \int_0^n f, \oint_0^n f \right\}$$
- les sommes et produits ($\sum_{i=0}^{n} x_i$, $\prod_{i=0}^{n} x_i$, $\bigcup_{i=0}^{n} x_i$, ...),
- les fonctions particulières ($lim_{x \to \infty} x = 0$),
- les matrices
$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{pmatrix}$$

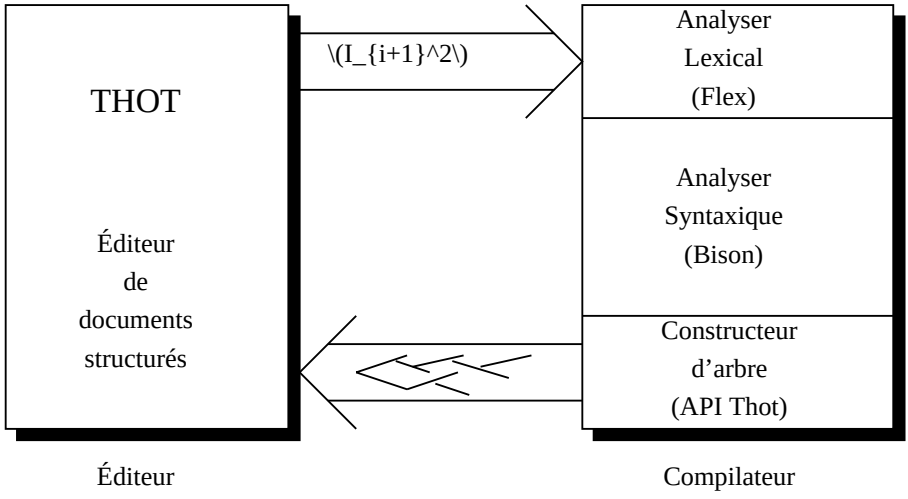

Figure 3 – *Architecture éditeur/compilateur*

En plus de ces constructions, nous traitons les directives sur les espaces, les caractères spéciaux et les effets comme les vecteurs, la négation ou les accolades. Toute macro ou directive non reconnue est conservée dans le texte, dans l'élément structuré **Embedded TEX**. Cette directive n'apparaît pas lors de l'impression effectuée directement par Thot, mais est conservée et restituée lors de l'exportation en LATEX.

La modification d'une formule existante peut se faire soit en mode structuré guidé, c'est-à-dire le mode de fonctionnement normal de Thot, soit en LATEX en passant par le compilateur. Le premier mode convient pour les modifications simples (passage du mode display au mode displaymath, correction de faute d'orthographe, ajout d'éléments simples, etc). Pour les modifications de structure, il vaut mieux utiliser le mode LATEX. En sélectionnant la formule mathématique et le mode d'édition LATEX, l'utilisateur est dérouté sur une fenêtre (figure 2) dans laquelle s'affiche la formule LATEX recomposée associée à la formule du document.

Le compilateur de formules LATEX est un outil indépendant communiquant avec Thot par pipe Unix. Écrit en C, il est construit à l'aide des outils Flex et Bison pour l'analyse lexico-syntaxique. La construction de l'arbre qui sera inséré dans le document initial est réalisée à l'aide de l'API Thot.

L'interface avec l'utilisateur est quant à elle bâtit sur la boîte à outil fournie avec l'éditeur. La figure 3 synthétise l'architecture de notre système.

4. Limites de l'approche

Lors de la mise en œuvre du traitement des expressions mathématiques, nous nous sommes heurtés au problème de la multiplicité des symboles mathématiques. Nous traitons environ 80% des symboles et nous sommes actuellement limités par les fontes utilisées dans Thot (Iso-Latin-1 et Symbol) qui ne permettent pas l'affichage et l'impression de tous ces caractères spéciaux. Toutefois, l'architecture de notre compilateur permet d'étendre le jeux des caractères traités en associant à un symbole, un caractère approchant dans une des fontes supportées. Cette extension se fait dans le fichier de définition des symboles supportés, fichier chargé à l'initialisation du compilateur. Dans tous les cas, l'expression d'un caractère non supporté pourra être affiché lors de l'exportation du document en LaTeX et représenté à l'écran comme de l'*embedded TeX*.

Une autre limite de notre approche est la non prise en compte de macros utilisateur. En effet, notre système traite directement les macros des environnements liés aux expressions mathématiques ce qui fige la grammaire du langage traité.

5. Conclusion

Même si nous ne prenons pas en compte la totalité de LaTeX, le choix de cette syntaxe et du sous-ensemble supporté nous parait bon, dans le cadre de notre projet, car elle est d'une part simple, d'autre part peu verbeuse et enfin répandue chez les utilisateurs potentiels de notre logiciel. Notre système permet de définir des formules mathématiques sophistiquées simplement en intégrant de façon harmonieuse l'approche syntaxique et l'approche textuelle.

Bibliographie

[1] *Amaya Overview*, http://www.w3.org/pub/WWW/Amaya/, juillet 1996.

[2] V. Quint, I. Vatton, "Grif: an Interactive System for Structured Document Manipulation", *Text Processing and document Manipulation, Proceedings of the International Conference,* J. C. van Vliet, ed., pp. 200-213, Cambridge University Press, 1986.

[3] R. Furuta, V. Quint, J. André, "Interactively Editing Structured Documents", *Electronic Publishing - Origination, Dissemination and Design.,* vol. 1, num. 1, pp. 19-44, Avril 1988.

[4] V. Quint, I. Vatton, *Hypertext Aspects of the Grif Structured Editor: Design and Applications,* num. R.R. 1734, INRIA, Rocquencourt, July 1992.

[5] C. Roisin, I. Vatton, "Merging Logical and Physical Structures in Documents", *Electronic Publishing – Origination, Dissemination and Design, special issue Proceedings of the Fifth International Conference on Electronic Publishing, Document Manipulation and Typography, EP94,* vol. 6, num. 4, pp. 327-337, April 1994.

[6] L. Lamport, "LaTeX: A Document Preparation System, User's Guide & Reference Manual", *Addison-Wesley Publishing Company* pp. 41-52.