

---

# Spécification de structures de liens hypertextuels adaptatives

**Jean-Hugues Réty**

LINC  
IUT de Montreuil  
140 rue de la Nouvelle rance  
93100 Montreuil  
jh.rety@iut.univ-paris8.fr

---

*RÉSUMÉ. Les hypertextes et hypermédias adaptatifs possèdent une structure de navigation (structure de liens) qui est dynamique : les possibilités de navigation offertes à l'utilisateur sont calculées dynamiquement en fonction de certaines des caractéristiques de celui-ci ou de son parcours précédemment suivi. Après avoir mis en évidence le besoin d'outils informatiques et d'outils théoriques qui permettraient la conception, l'analyse et le contrôle de cette structure, nous proposons la définition de langages de spécification de structures. Nous illustrons nos propos par la littérature hypertextuelle et le langage de spécification de liens conditionnels mis en œuvre dans le cadre du projet " Connection Muse ".*

*ABSTRACT. adaptive hypertexts and hypermedia are equipped with a dynamic navigation structure(link structure): navigation choices given to the user are dynamically computed, depending on some user characteristics or her previous navigation path. We first advocate the need for computer science tools and theoretical tools for conception, analysis and structure control. We then propose the definition of a structure specification language. We discuss hypertext literature and conditional links specification language defined in the Connection Muse framework.*

*MOTS-CLÉS : Hypertexte adaptatif, Structure de liens hypertextuels, Sémantique des langages de programmation. Littérature hypertextuelle.*

*KEY WORDS: Adaptive hypertext, Hypertext links structure, programming languages semantics, Hypertext literature.*

---

## **1. Introduction**

Les hypertextes adaptatifs (et plus généralement les hypermédias adaptatifs) forment un domaine de recherche aux applications multiples : enseignement en ligne, documentation en ligne, applications artistiques, etc. L'idée d'adaptativité s'est d'ailleurs vue appuyée par les évolutions de ces dernières années du World Wide Web où l'on cherche de plus en plus à fournir un contenu personnalisé à l'utilisateur. Cette notion d'adaptativité recouvre de multiples réalités et champs de recherche. Elle peut concerner des contenus, des menus de navigation, des plans de navigation dynamiques, etc (voir [Brusilovski] pour une vue d'ensemble).

Nous pouvons ranger le domaine des hypertextes adaptatifs en deux catégories. Dans la première, il s'agit d'adapter un hypertexte existant, non adaptatif, à un utilisateur particulier – par exemple la lecture d'un journal en ligne en fonction des centres d'intérêt du lecteur. Dans le second cas, il s'agit de concevoir l'adaptativité à l'intérieur même de l'hypertexte, ce dernier étant programmé pour prendre en compte certaines caractéristiques de l'utilisateur – c'est le cas par exemple de la plupart des hypertextes d'enseignement et de certains romans ou poèmes hypertextuels. Nous nous intéressons dans cet article à cette seconde catégorie, et plus particulièrement à la question de la structure de liens hypertextuels adaptative (que nous abrègerons souvent par structure adaptative). Nous nous interrogerons sur les moyens de la définir, la spécifier, l'analyser et la contrôler. Il s'agit là de questions centrales à la recherche dans le domaine des hypertextes adaptatifs. Nous argumenterons que cette structure ne doit pas être le résultat d'un travail de programmation, mais plutôt d'un travail de spécification.

Une structure adaptative a pour caractéristique essentielle que les liens qui la composent sont calculés dynamiquement, au fur et à mesure de la progression de l'utilisateur, en fonction de certaines informations connues au sujet de celui-ci. Dans le cadre d'un cours en ligne, il peut s'agir par exemple de ne pas donner accès à un étudiant de niveau débutant à certains développements qui requièrent des connaissances plus complètes, puis de lui permettre de changer de niveau après validation d'acquis. Dans le cadre d'un hypertexte narratif, il pourra s'agir d'entretenir un suspens et ne donner accès à tel passage clé qu'après avoir pris connaissance de suffisamment de développements. Il apparaît clairement qu'une telle structure est plus difficile à créer et à contrôler qu'une structure uniquement composée de liens statiques – non dynamiques.

## **2. Structure de liens : les fondements informatiques**

Les hypertextes non adaptatifs – par exemple les hypertextes écrits en HTML uniquement – utilisent des liens statiques ; posés en dur, une fois pour toute. De tels liens forment une relation binaire sur l'ensemble des pages HTML. Une structure de liens définie par des liens statiques forme donc un graphe orienté, les pages en étant les sommets, et les liens hypertextuels les arêtes. Il s'agit là d'un objet

mathématique bien connu, étudié en détail depuis plus d'un siècle, et pour lequel de nombreux algorithmes et méthodes d'analyse sont disponibles.

Dans le cadre des hypertextes adaptatifs, les liens peuvent être calculés dynamiquement en fonction de différents paramètres. De tels liens sont donc programmés. Leur existence ou leur destination effective dépendent de l'exécution d'un programme qui les déterminent. La structure générée par ces liens forme elle-même un programme, c'est à dire quelque chose qui a fait l'objet d'un immense effort de recherche et d'étude depuis plusieurs décennies ; c'est-à-dire un objet informatique pour lequel existent de nombreux modèles, méthodes et techniques de conception et d'analyse. Il serait dommage d'ignorer les résultats de ces recherches et de ne pas profiter de leurs enseignements.

Une autre manière de voir les choses est de constater qu'une structure adaptative ne peut être directement représentée par un graphe orienté. S'agissant en fait d'un programme informatique, elle peut en revanche être représentée à l'aide d'outils utilisés pour définir la sémantique opérationnelle des langages de programmation [Plotkin], comme par exemple un système de transition.

S'il est besoin de motivations supplémentaires, considérons l'importance que peuvent avoir les représentations graphiques de la structure de liens dans le processus de création d'hypertextes. Le fait de dessiner le graphe de la structure permet non seulement l'obtention d'une vue d'ensemble, mais permet aussi de s'assurer que certaines propriétés globale ou locale sont respectées, comme par exemple l'existence d'un chemin allant d'une page à une autre, ou l'existence d'un chemin de longueur inférieure à une longueur donnée. La représentation graphique d'une structure adaptative est cependant très difficile voir même impossible, du fait même que les liens hypertextuels qui la composent ne sont pas de simples relations binaires entre deux pages, mais sont le résultat de l'exécution d'un programme. Le besoin d'outils plus élaborés se fait ici sentir.

### 3. Le cas des hypertextes littéraires

Considérons le cas des créations artistiques sur support informatique. Celles-ci présentent le plus souvent pour caractéristique d'être très liées à la machine. La programmation représente une grande part du travail de création. Il s'ensuit que l'artiste doit non seulement maîtriser son outil, ce qui est normal, mais maîtriser aussi les mécanismes et rouages internes de cet outil ; et cela à un niveau très fin de détail. Hormis le fait qu'il semblerait abérant de devoir effectuer des études poussées de mécanique auto et comprendre le fonctionnement interne du moteur à explosion, pourquoi telle soupape s'ouvre et à quel moment, avant de pouvoir conduire une automobile, on ne peut que constater que la très grande complexité des systèmes informatiques actuels rend la tâche de création pratiquement surhumaine.

L'hypertexte, né de l'article visionnaire de Vanavar Bush [Bush], du travail de Ted Nelson [Xanadu], puis de l'explosion du World Wide Web, a offert à la création artistique un cadre formel qui a été exploité dès la fin des années 80 aux

Etats-Unis sous l'appellation de Littérature Hypertextuelle. L'idée est d'écrire des poèmes, des romans, des textes littéraires en général sous forme hypertextuelle : des fragments de texte reliés par des liens. Il est important de comprendre qu'avec l'hypertexte littéraire un pas qualitatif est franchi. Les créations artistiques sur ordinateur précédentes reposaient, nous l'avons dit, pour une large part sur des activités de programmation. Une œuvre hypertextuelle – au sens strict – ne nécessite aucune programmation. Plus encore, elle est même indépendante de toute machine. Un hypertexte littéraire ne se caractérise pas par quelques milliers de lignes de code produisant un certain résultat lors de l'exécution, mais par un ensemble de pages (et leur contenu) et un ensemble de liens ; d'un point de vue plus littéraire : un ensemble de fragments muni d'une structure. Notons encore que l'on pourrait même se passer ici des mots " ordinateur " et " informatique ". L'ordinateur permet la réalisation effective, la représentation de l'œuvre – mais l'œuvre elle-même se définit sans lui.

Les avantages en sont immenses. L'œuvre retrouve ici une autonomie inespérée par rapport à la technique. Les auteurs d'hypertextes littéraires y gagnent de pouvoir mieux se concentrer sur la création, et non pas sur la réalisation. L'archivage aussi est concerné. Archiver une œuvre programmée sur ordinateur signifie archiver la machine elle-même, d'où des problèmes techniques insurmontables. Une œuvre hypertextuelle, qui est définie en terme de fragments et de liens entre ces fragments, peut potentiellement être représentée sans trop de difficultés sur n'importe quelle machine existante ou à venir. Une des raisons essentielles à cela est que sa structure de liens forme un graphe orienté, c'est à dire quelque chose qui a une signification – précise – indépendamment de toute machine sur laquelle pourrait être représentée cette structure. Tout cela provient de l'utilisation de l'hypertexte en tant que modèle conceptuel de la littérature hypertextuelle.

Ce modèle s'est hélas bien vite révélé insuffisant. La structure des réalisations construites sous la forme d'hypertextes (basés sur des liens statiques) souffre le plus souvent de manques dont la manifestation la plus flagrante est le fameux " perdu dans l'hyper-espace ". Qui a parcouru le célèbre premier roman hypertextuel *Afternoon, a story* de Michael Joyce [Joyce], s'est très probablement trouvé confronté à la situation de tourner en rond parmi des fragments déjà lus, à la recherche d'un hypothétique lien vers de nouveaux développements.

Il s'est ainsi avéré nécessaire de disposer d'outils plus puissants de définition et de contrôle de la structure de liens que les seuls liens statiques. Cette question trouve une réponse naturelle dans la notion d'adaptativité, dont l'une des motivations originelles est de permettre le guidage de l'utilisateur – le lecteur – à travers la complexité de la structure de liens. Cette approche, n'est pas nouvelle. Le logiciel StorySpace ([Bernstein]), apparu à la fin des années 80 et entièrement dédié à l'écriture hypertextuelle, possède une notion de garde qui permet d'introduire dans la structure de liens certaines fonctionnalités adaptatives. Celles-ci reposent sur une forme très limitée de lien conditionnel, l'existence d'un lien pouvant dépendre du fait que certaines pages aient été lues ou non, ou bien du fait que certains liens aient

été suivis ou non. Cette fonctionnalité adaptative est cependant bien trop limitée pour réellement répondre aux problèmes soulevés par l'écriture hypertextuelle. Le projet " Connection Muse " va pour sa part beaucoup plus loin, proposant un large éventail de fonctionnalités adaptatives basées sur un langage de spécification de liens conditionnels. Nous reviendrons en détail sur ce point plus loin. Remarquons que dans les deux cas, il s'agit de réaliser des hypertextes littéraires basés sur des structures de liens adaptatives.

Introduire dans les hypertextes littéraires une structure adaptative, c'est à dire programmée, résultat dynamique de l'exécution de programmes, a hélas pour conséquence d'effectuer un pas en arrière en terme d'indépendance par rapport à la machine et à la technique en général. D'autre part, les écrivains n'ont en général pas envie de coder par eux-mêmes les programmes permettant de définir le comportement adaptatif des liens. L'objectif est alors de rendre cette tâche, du point de vue de l'auteur, simple et transparente, et du point de vue de l'informatique : déclarative.

#### **4. De la notion de déclarativité**

L'un des grands courants ayant marqué l'histoire des langages de programmation est caractérisé par la notion de déclarativité. Les langages de programmation impératifs, tel Basic, Pascal ou C, restent très proches du fonctionnement de la machine, et en particulier la notion de variable (impérative) qu'on y trouve reflète directement le fonctionnement de la mémoire des ordinateurs. Les langages déclaratifs, au contraire, sont basés sur un modèle de calcul abstrait de la machine, bénéficiant d'une définition mathématique précise appelée sémantique déclarative. Les deux principaux représentants de la programmation déclarative sont la programmation fonctionnelle et la programmation en logique [Kowalski], cette dernière nous intéressant plus particulièrement ici.

Parmi les avantages de la programmation déclarative se trouve le fait que celle-ci bénéficie d'une sémantique déclarative totalement indépendante de la machine. La sémantique d'un programme logique est ainsi directement définie en terme de logique du premier ordre. Les hypertextes littéraires ont tout d'abord permis de s'abstraire de la machine, avant de réintroduire la programmation par l'intermédiaire de structures de liens adaptatives. Afin de retrouver cette indépendance perdue, nous chercherons donc à proposer un langage déclaratif de définition des liens adaptatifs. Le fait que ce langage bénéficie d'une sémantique déclarative claire assurera, entre autres, l'archivabilité des hypertextes.

#### **5. Outils formels d'analyse de la structure**

Comme nous l'avons déjà mentionné, des outils pour construire et analyser les structures de liens sont nécessaires. Dans le cadre non adaptatif, les outils basés sur

le graphe de la structure, et à commencer par la représentation graphique de celui-ci, peuvent être utilisés. Nous avons proposé dans [Réty] de représenter une structure de liens adaptative basée sur des liens conditionnels sous la forme d'une théorie logique en logique du premier ordre. Cette théorie logique, se présente de plus naturellement sous la forme de clauses de Horn, c'est-à-dire sous la forme d'un programme logique. Cela donne lieu à un cadre formel pour tester des propriétés de la structure, l'exécution du programme logique obtenu permettant de prouver des propriétés exprimées sous la forme de formules logiques. Notons aussi que Furuta et Stotts ont proposé dans [Furuta-Stotts] l'application de méthodes de model checking à un cadre de systèmes d'hypertextes dont la structure adaptative est basée sur des réseaux de Petri.

L'idée est ici de tester une structure afin de vérifier qu'elle satisfait bien à certaines propriétés. Considérons un roman hypertextuel. L'auteur peut par exemple vouloir que telle page où se passe une action mettant en jeu deux personnages ne puisse en aucun cas être lue avant telle autre page dans laquelle est racontée la rencontre de ces deux personnages. Dans le cas où existe une page " Fin ", l'auteur pourrait aussi vouloir s'assurer du fait que le lecteur n'arrivera jamais à cette page avant d'avoir lu 80% au moins - ou bien la totalité – des pages du roman. On pourrait aussi vouloir éviter que le lecteur ne passe plus d'un certain nombre de fois sur une même page. L'écriture d'un poème ou d'un roman hypertextuel peut ainsi nécessiter une structure de liens satisfaisant un nombre important de propriétés de ce genre. D'où l'intérêt d'outils permettant de tester ces propriétés de manière automatiques. Nous pensons que des outils formels issus de la recherche sur les langages de programmation pourraient être utilisés ici avec succès.

Mais tester une propriété est une chose. Dans le cas d'un roman comportant plusieurs centaines de pages et des milliers de liens, que faire si la propriété en question n'est pas satisfaite. Cela signifie alors que la structure de liens n'est pas satisfaisante, qu'il faudrait la modifier. Mais quel lien (ou quels liens) modifier sur les plusieurs milliers que compte l'hypertexte, et comment les modifier. Dès que le nombre de liens et de pages devient important, il s'agit là d'un problème inextricable. Nous pensons que la notion informatique de débogage peut utilement s'appliquer ici. Rappelons une fois de plus qu'une structure de liens adaptative est un programme. La recherche en informatique s'est depuis longtemps posé la question de la correction des programmes éronnés, notamment dans le cadre de la programmation en logique. Des outils de débogage déclaratif pourraient utilement s'appliquer aux structures de liens hypertextuelles, et particulièrement aux structures adaptatives.

## **6. Structures de liens : de la nécessité d'un langage de spécification**

De tels outils formels ne peuvent s'appuyer que sur un langage formel : un langage de spécification de la structure de liens muni d'une sémantique déclarative

claire. Se pose alors la question suivante : quel type de langage, quels types de liens hypertextuels, pour quelles structures ?

Les liens conditionnels forment un cas particulier de liens dynamiques. Il s'agit d'une classe particulièrement intéressante de liens hypertextuels, ce pour deux raisons. D'une part, de tels liens offrent un large pouvoir d'expression dans la définition de la structure. Cela a été mis en évidence par le projet " Connection Muse " [Connection]. Le poème Penetration de Robert Kendall [Kendall] qui utilise les outils développés dans le cadre du projet offre un intéressant exemple d'une structure permettant de guider le lecteur vers la lecture de la totalité des pages – lui évitant ainsi de tourner en rond – tout en mettant en valeur la structure narrative de l'œuvre. D'autre part, il a été mis en évidence dans [Réty] que les liens conditionnels peuvent être modélisés sous forme de clauses de Horn en logique du premier ordre. Une structure basée sur des liens conditionnels apparaît donc comme étant un bon candidat.

## **7. Vers un langage de spécification de la structure : l'exemple " Connection Muse "**

Le projet " Connection Muse ", projet sur lequel nous travaillons avec l'écrivain américain Robert Kendall depuis 1999, a pour objectif de proposer un environnement logiciel pour l'écriture littéraire hypertextuelle pour le World Wide Web. Mais bien au delà, il s'agit de cerner quelles sont les particularités et les besoins spécifiques à l'écriture littéraire hypertextuelle. Bien qu'ayant vocation, à plus long terme, à prendre en compte l'ensemble de la problématique que représente ce que l'on a coutume de nommer outre-atlantique la " littérature hypertextuelle ", notre projet s'est dans un premier temps concentré sur cette question centrale que représente la structure de liens hypertextuels. D'un point de vue littéraire, cette question se ramifie en de nombreuses problématiques que nous n'aborderons pas ici. Le point important pour notre propos présent est, pour cette écriture, l'évidente nécessité d'une structure de liens dynamique, qui se recombine en fonction du lecteur et de son parcours, au fur et à mesure de sa lecture ; bref, d'une structure de liens adaptative. D'un point de vue technique, une telle structure doit être programmée : il s'agit pour l'auteur de programmer le comportement futur de l'hypertexte. Mais l'auteur, en général, n'a pas envie de s'occuper de programmation ; le moins possible, en tout cas. Nous nous sommes donc orienté vers un langage de spécification de la structure de liens dynamique qui soit compréhensible et facile à mettre en œuvre – du moins, autant que possible – par des non informaticiens. Ce langage de spécification est composé d'un ensemble de prédicats et d'un ensemble de fonction de sélection. Comme tout système d'hypertexte adaptatif, ce langage est aussi basé sur un modèle d'utilisateur qui représente l'ensemble des informations connues sur le lecteur.

### 7.1. Le modèle d'utilisateur

Le modèle d'utilisateur considéré consiste en la liste ordonnée des pages visitées par le lecteur. Cette information est suffisante pour tracer la totalité des déplacements du lecteur dans la structure de liens. Il est ainsi possible de connaître, par exemple, si un nœud a été visité ou non, si oui, combien de fois, combien de nœuds ont été visités depuis la dernière fois que celui-ci l'a été, etc. Connaissant le nombre total de nœuds que possède l'hypertexte, il est aussi possible de savoir quel pourcentage a été parcouru. Ce modèle d'utilisateur, aussi simple soit-il, s'est avéré déjà très puissant en regard des objectifs fixés concernant l'écriture hypertextuelle. L'écrivain est en premier lieu confronté à des questions d'ordonnement effectif lors de la lecture, et c'est précisément cet ordonnancement que prend en compte le modèle d'utilisateur considéré.

Des modèles d'utilisateurs plus complexes, par exemple intégrant des variables d'état comme c'est le cas dans *A life Set For Two* de Robert Kendall [Kendall2], peuvent bien sûr s'avérer utiles mais débordent le cadre qui nous intéresse ici.

### 7.2. Langage de spécification des conditions

Le langage de spécification des conditions est formé par un ensemble de prédicats portant sur le modèle d'utilisateur, c'est-à-dire sur la liste des pages visitées par le lecteur, dont voici quelques exemples :

visited(page)

Spécification : vrai si la page a été visitée, faux sinon.

allVisited(liste\_de\_pages)

Spécification : vrai si toutes les pages de la liste ont été visitées, faux sinon.

allLinked(liste\_de\_pages)

Spécification : vrai si toutes les pages de la liste sont la destination de l'un des liens de la page courante, faux sinon.

Notons que le langage définit de plus un certain nombre de fonctions qui peuvent être utilisées en combinaison avec des prédicats pour former des conditions. Par exemple, avec la fonction nbVisits() dont la spécification est de retourner le nombre de fois qu'a été visitée la page courante, on peut construire la condition nbvisits() > 2 qui est vraie si la page courante a été visitée plus de 2 fois, et fausse sinon.

### 7.3. Fonctionnalités de sélection

Le langage dispose aussi de fonctionnalités de sélection permettant par exemple de sélectionner la destination d'un lien parmi un ensemble de destinations possibles. La sélection peut être aléatoire ou bien dépendre de conditions. Les fonctionnalités de sélection présentes actuellement dans le langage permettent de sélectionner aléatoirement une page parmi un ensemble de pages, ou bien de sélectionner une page vérifiant certaines conditions parmi un ensemble de pages. Comme précédemment, le comportement de chaque fonctionnalité de sélection est spécifié de manière très précise.

Ce langage dans son ensemble est basé sur des liens conditionnels qui peuvent être munis d'une sémantique déclarative comme indiqué dans [Réty]. D'autre part, il a déjà permis des réalisations intéressantes, faisant usage d'une structure adaptative élaborée. Pour plus de détails, se reporter à [Connection] et [Kendall-Réty].

## 7. Conclusion

Nous avons énoncé dans [Réty] que la structure de liens d'un hypertexte formée de liens conditionnels peut être transformée en une théorie logique en logique du premier ordre. Dans cet article, nous proposons alors d'utiliser le cadre de la programmation en logique pour mettre en œuvre des méthodes d'analyse et de débogage de la structure. Nous avons souhaité ici apporter des arguments en faveur de l'utilisation de langages déclaratifs de spécification de la structure de lien dans le cadre des hypertextes adaptatifs pour lesquels l'adaptativité est définie dans la structure elle-même.

Nous pensons que l'hypertexte, au moins en ce qui concerne la problématique de la structure de liens adaptative, peut tirer beaucoup d'enseignement des recherches menées depuis des décennies dans le domaine des langages de programmation. Le langage de spécification de la structure mis en œuvre dans le projet " Connection Muse " est orienté dans cette direction.

## 7. Bibliographie

- [BER] BERNSTEIN M., *Storyspace*, <http://www.eastgate.com>
- [BRU 96] BRUSILOVSKI P., « Methods and techniques of adaptive hypermedia », *User modeling and user-adapted interaction*, 6(2-3), pp. 87-129, 1996.
- [BUS 45] BUSH V., « As we may think », *The atlantic monthly*, juillet 1945.
- [FSC 98] FURUTA R, STOTTS P.D., CABARRUS C.R., « Hyperdocuments as automata: verification of trace-based browsing properties by model checking », *ACM transactions on information systems*, 16(1), pp. 1-30, 1998.

[JOY 87] JOYCE M., *Afternoon, a story*, Eastgate Systems, 1987.

[KEN 00] KENDALL R., *Penetration*, Eastgate Systems,  
<http://www.eastgate.com/Penetration>, 2000.

[KEN 96] KENDALL R., *A life set for twoctes*, Eastgate Systems, 1996.

[KR] KENDALL R., RETY J.-H., *Connection Muse*, <http://www.wordcircuits.com/connect>

[KR 00] KENDALL R., RETY J.-H., « Toward organic hypertext », *ACM conference on hypertext and hypermedia*, 2000.

[KOW 79] KOWALSKI R., « Algorithm = logic + control », *Communication of the ACM*, 22(7), pp. 424-436, 1979.

[NEL] NELSON T., *Xanadu project*, <http://www.xanadu.com>

[PLO 81] PLOTKIN G., « A structural approach to operational semantics », TR DAIMO FN-19, Aarhus University, Denmark, 1981.

[RET 99] RETY J.-H., « Structure analysis for hypertext with conditional linkage », *ACM conference on hypertext and hypermedia*, 1999,.

[LAL 97] L A., « », *Actes*, 1997,.