HAL
open science

La génération automatique de textes comme support à la compréhension de modèle de tâches en conception : une étude préliminaire

Sybille Caffiau, François Portet

To cite this version:

Sybille Caffiau, François Portet. La génération automatique de textes comme support à la compréhension de modèle de tâches en conception : une étude préliminaire. 29ème conférence francophone sur l'Interaction Homme-Machine, AFIHM, Aug 2017, Poitiers, France. 12 p., 10.1145/3132129.3132137 . hal-01578499

HAL Id: hal-01578499

https://hal.science/hal-01578499v1

Submitted on 29 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

La génération automatique de textes comme support à la compréhension de modèle de tâches en conception : une étude préliminaire

Natural Language Generation to support the understanding of task models : a preliminary study

Sybille Caffiau
Univ. Grenoble Alpes, CNRS, Grenoble-INP
IIHM/Laboratoire d'Informatique de Grenoble
791 rue des Résidences - Domaine Universitaire
38401, SAINT MARTIN D'HÈRES, France
Sybille.Caffiau@imag.fr

François Portet
Univ. Grenoble Alpes, CNRS, Grenoble-INP
Getalp/Laboratoire d'Informatique de Grenoble
791 rue des Résidences - Domaine Universitaire
38401, SAINT MARTIN D'HÈRES, France
François.Portet@imag.fr

ABSTRACT

During the design of IS, several stakeholders, each with their own expertise and culture, will use the task model (TM). Then, TM must be understood by several kinds of stakeholders. Adapting the presentation of the TM information to the stakeholder and the use he or she makes of it, is therefore a key point for them to be understood. however, none of the tools for use information of the TM allow to understand the modeled activity without knowing the notation used. In this paper, we propose to use a presentation in the textual form as a tool for verification and consultation of the task model. Two evaluations were conducted on the use of this presentation for (1) the correction of a model by its designers and (2) the understanding of a task model by computer scientists who did not design the model. These evaluations show that the textual presentation of the task model provides a complementary point of view allowing a better understanding and facilitating the improvement of the model.

CCS CONCEPTS

• **Human-centered computing → User centered design**; *Activity centered design*; Scenario-based design;

KEYWORDS

Task model, scenario, natural language generation, checking tools, evaluation

© ACM, 2017. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution.
The definitive version was published in Actes de la 29ème conférence francophone sur l'Interaction Homme- Machine, IHM '17, August 28–September 1, 2017, Poitiers, France
https://doi.org/10.1145/3132129.3132137

RÉSUMÉ

Pendant la conception de SI, plusieurs intervenants, qui ont chacun une expertise et une culture propres, sont amenés à utiliser le modèle de tâches (MdT) et donc à le comprendre. Adapter la présentation des informations du MdT à l'intervenant et à l'utilisation qu'il va en faire est alors un point clé pour qu'elles puissent être comprises. Or, actuellement aucun des outils de présentation des MdT ne permet de comprendre l'activité modélisée sans connaitre la notation utilisée. Dans cet article, nous présentons une approche de génération d'une présentation textuelle du MdT. Deux évaluations ont été menées sur l'utilisation de cette présentation pour (1) la correction d'un modèle par ses concepteurs et pour (2) la compréhension d'un modèle par des informaticiens qui ne l'ont pas conçu. Ces évaluations montrent que la présentation textuelle apporte un point de vue complémentaire sur le MdT permettant une meilleure compréhension et facilitant son amélioration.

MOTS-CLEFS

Modèle de tâches, scénario, génération automatique de texte, outils de vérification, évaluation

1 INTRODUCTION

La conception d'un système interactif nécessite l'intervention d'un grand nombre d'intervenants, qu'ils soient experts du domaine d'application, utilisateurs finaux, ergonomes, designers ou informaticiens. Cette conception repose sur la création de modèles du système interactif (maquettes, scénarios, automates...) qui seront au cœur des discussions et serviront de fil conducteur et de référentiel commun entre les acteurs de la conception. Cependant, les formalismes de modélisation et la culture d'utilisation ne sont pas partagés par l'ensemble des acteurs. Il s'ensuit de régulières incompréhensions et erreurs de conception dues à un transfert d'information défaillant entre les acteurs. Les travaux présentés dans cet article ont pour but de faciliter la compréhension d'un modèle particulier utilisé en conception de système interactif : le modèle de tâches. Celui-ci, bien qu'intuitif pour les initiés, reste source d'incompréhension ou de méprise pour les néophytes. Plusieurs formalismes permettent de représenter l'activité sous forme

de modèle de tâches. Ils sont principalement représentés sous forme graphique. Nous présentons une étude qui utilise le langage naturel comme mode de description du modèle de tâches et montrons son intérêt pour la compréhension de l'activité par des acteurs de la conception.

La conception centrée utilisateur nécessite la prise en compte des utilisateurs, le contexte d'utilisation ainsi que les objectifs d'utilisation [29]. Dans cet article, nous nous intéressons à la phase de conception qui vise à introduire les connaissances initiales (sur l'environnement d'utilisation) dans le processus de conception. Nous considérons, en particulier, les connaissances liées à l'activité représentée sous forme de modèle de tâches. Celui-ci permet de structurer les activités sous forme de tâches organisées hiérarchiquement (de la plus abstraite aux plus concrètes) et temporellement (via des opérateurs d'ordonnancement et des caractéristiques sur les tâches). De plus, à chaque tâche est associée des informations sur les acteurs qui la réalisent (système, interaction, humaine) et parfois sur le contexte (via des objets). Le modèle de tâches permet donc de représenter formellement des connaissances utiles à différents types d'experts tout au long du processus de conception. Cependant, ces informations ne sont pas accessibles dès lors que celui qui veut y accéder ne connait pas le formalisme employé [24, 32].

Nous avons voulu explorer l'utilisation d'un support à la communication entre les membres d'une équipe de conception : le texte écrit en langage naturel. Outil communément utilisé pour la communication humaine, il ne nécessite aucun apprentissage préalable. Le principe est de générer automatiquement un texte à partir d'une représentation formelle afin de le rendre le contenu plus accessible à l'utilisateur. Ce principe est à la base des approches *model-to-text* M2T dont l'instance la plus répandue est le MOF2Text (tel que spécifié par l'Object Management Group [1]) pour la génération automatique de documents à partir de modèles (p.ex., UML). Il faut cependant souligner que le MOF2Text n'a pas pour objectif le langage naturel mais un langage contrôlé (code source, documentation) et nécessite donc un certain niveau d'expertise pour la lecture du texte généré. Le traitement automatique du langage naturel a déjà fait l'objet de travaux notamment concernant l'*extraction* de concepts à partir de textes naturels pour la conception interactive de modèle de tâches à partir de descriptions textuelles ; on peut notamment citer U-TEL [38] utilisant un WoZ pour l'élicitation des éléments du modèle de tâches et plus récemment [22] utilisant des outils travaillant sur le lexique. Sur l'aspect *génération* d'un texte à partir d'un modèle de tâches, les travaux sont plus rares. On peut cependant citer [30] dont l'objectif était la génération automatique d'aide en ligne décrivant les procédures pour réaliser les tâches sur un SI développé. Cependant, dans les phases amont de la conception, l'objectif de l'utilisation du texte en langage naturel est autre : faciliter la communication des intervenants et favoriser la projection des lecteurs dans des situations [6].

L'utilisation du langage naturel pour la transmission d'information à partir de connaissances ou données non-linguistiques est connue comme le domaine de la Génération Automatique de Textes (GAT) [35]. La GAT est utilisée depuis plusieurs décennies dans la génération de descriptions de modèles et peut donc être rapprochée des objectifs de l'approche M2T. Un des exemples les plus représentatifs étant ModelExplainer [20], un système permettant de décrire en texte un diagramme de classes UML pour la programmation orientée objet ou encore NaturalOWL [2] pour la description automatique d'ontologie. La GAT s'intéresse au processus permettant de formuler un énoncé à partir de sources aussi diverses que des signaux physiologiques [34], des modèles informatiques [20], des diagnostics de machines [11] ou des données météorologiques [14]. La GAT est également employée dans le domaine des "data analytics" pour résumer de grandes quantités de données et ainsi réduire la surcharge d'information. Bien que les représentations graphiques soient souvent vues comme plus intuitives, le langage naturel est plus à même de communiquer certaines informations complexes telles que des informations de causalité ou d'argumentation, des liens et des informations multi-échelles. Par ailleurs, il existe une grande variabilité chez les individus dans la capacité à intégrer des informations sous forme graphique ou linguistique [14]. Plusieurs études, notamment dans le domaine de la décision médicale, ont montré qu'exposés à une présentation textuelle d'information, les cliniciens pouvaient prendre de meilleures décisions qu'avec une présentation purement graphique alors même qu'ils préféraient cette dernière [21, 27, 34]. Ceci montre que bien qu'il est stérile d'opposer graphique et texte étant donné leur nature complémentaire, le langage naturel est une représentation à privilégier lorsque des informations complexes doivent être communiquées au grand public ou à des acteurs non-spécialistes (p.ex., des novices).

La section suivante présente une étude bibliographique des outils d'aide à la réalisation des modèles de tâches. Cette étude fait un bilan des difficultés pouvant se poser lors de la réalisation d'un modèle de tâches et les outils qui proposent d'y répondre. Elle nous a amené à considérer la présentation textuelle comme un outil pouvant compléter les supports déjà proposés. Une démarche de génération de la présentation textuelle en langage naturel est présentée en section 3, c'est cette démarche qui a été implémentée pour les évaluations (section 4 et 5). Enfin, avant de conclure et de présenter les travaux futurs (section 7), nous discutons de l'approche (section 6).

2 ETAT DE L'ART : LES OUTILS D'AIDE À LA RÉALISATION DE MODÈLE DE TÂCHES

Comme tout modèle, le modèle de tâches a pour objectif de présenter des connaissances sur un objet d'intérêt (ici : l'activité). Pour cela, les outils doivent soutenir le processus de modélisation. D'après [7], pour le modèle de tâches, ce processus est une itération d'étapes : analyses systématiques, construction du modèle, validation auprès des opérateurs et correction du modèle. Nous considérons le modèle de tâches dans le cadre de la conception de systèmes interactifs nous avons donc choisi de prendre en compte les corrections du modèle de tâches qui expriment des modifications dues à l'intégration du système interactif dans l'activité en plus de celles vouées à la validation par les opérateurs.

Les analyses systématiques sont les études terrain, interviews...qui permettent d'acquérir des connaissances sur l'activité à modéliser. Les productions à l'issu de ces analyses peuvent prendre des formes variées. Nous avons choisi de ne pas intégrer à cet état de l'art les outils pour les analyses systématiques mais de considérer ceux

1. http://www.omg.org/spec/MOFM2T/1.0/PDF/

dont l'objectif est d'extraire les connaissances pour construire le modèle. En dehors de ces outils dédiés à la construction initiale, les opérations de construction et de correction du modèle nécessitent toutes deux la réalisation d'opération d'édition : navigation dans le modèle de tâches, détection d'erreurs d'édition, réutilisation de "morceaux" du modèle et suivi des modifications réalisées sur le modèle. Les outils permettant de supporter ces opérations sont présentés dans la section suivante alors que la section 2.2 abordera les outils pour la validation du modèle (Table 1).

Table 1: Outils pour la construction et l'utilisation des modèles de tâches

Étape	Opération	Outil proposé
Construction et correction du modèle	Extraction des connaissances des analyses systématiques	Elicitation de scénario
	Navigation dans un arbre de grande taille (lisibilité)	Vue panoramique
		Modularité
	Détection d'erreurs d'édition	Vérification de la grammaire
		Détection de différences de description
	Réutilisation	Patterns
	Suivi des modifications	Méta-opérateurs
Validation/ vérification du modèle auprès des opérateurs	Couverture du modèle vis à vis de certains déroulement	Simulation dynamique (pour un seul déroulement)
	Compréhension de la notation	Changement de présentation du modèle (pour un seul déroulement)

2.1 Construction et correction du modèle

Le modèle de tâches doit être construit à partir de données terrain (observation, scénario, rapport....). Des outils de construction du modèle à partir de données ont été développés. En particulier, [38] et [22] proposent d'extraire les connaissances nécessaires à la modélisation à partir de données textuelles par élicitation de scénarios.

L'édition est l'ajout, la modification ou la suppression d'un concept du modèle ou d'un de ses attributs. De nombreux éditeurs existent pour supporter les divers formalismes qui ont été créés (le lecteur pourra se référer à [15, 23, 26] pour des synthèses des formalismes). Ces éditeurs proposent principalement une représentation graphique du modèle et des fonctionnalités d'aide à l'édition de graphe (copier, supprimer, zoomer, imprimer, déplacer...). A ces fonctionnalités génériques s'ajoutent parfois des fonctionnalités propres à l'objet d'édition (le modèle de tâches). Par exemple, la vérification grammaticale permet de détecter des erreurs faites vis à vis du formalisme employé par le concepteur (par exemple dans l'éditeur K-MADe [3]). On trouve également des navigations facilitées dans l'arborescence, par exemple, des vues à des échelles différentes permettent au concepteur, en fonction de ses centres d'intérêt, de naviguer entre le modèle global et une partie détaillée de celui-ci.

Certains éditeurs facilitent la décomposition en plusieurs sous-arbres (principe de modularité) qui sont alors combinées pour reconstituer l'activité dans son intégralité. CTML [39] par exemple, propose de réaliser cette décomposition par acteur ; les tâches de chaque acteur faisant l'objet d'un arbre alors que HAMSTERS [25] propose des mécanismes de factorisation (par sous-routines et sous-modèles) sur lesquels reposent une division du modèle. Cette approche permet également de détecter des erreurs d'incohérences internes. Dans [12], les auteurs présentent comment l'utilisation de ces mécanismes (visant à simplifier la tâche d'édition par réutilisation d'éléments déjà édités) peut aussi amener à détecter qu'une même tâche a été décrite de deux manières différentes.

La modification du modèle peut être nécessaire pour suivre les évolutions de la conception du système interactif (SI) liées à la précisions du déroulement de l'activité modifiée après l'intégration de l'interaction. Par exemple, il peut être nécessaire d'ajouter une alternative pour réaliser une tâche suite à une évolution technique. Le risque de ces modifications est de ne plus prendre en compte des contraintes qui étaient initialement exprimées (et validées). Pour aider les concepteurs, Wurder et al [40] proposent l'utilisation de meta-opérateurs (à ajouter dans le modèle) qui permettent lors du raffinement du modèle de tâches de vérifier l'absence d'incohérences entre les versions du modèle de tâches.

Certaines tâches sont réalisées en suivant toujours les mêmes procédures (standards). Pour éviter les incohérences de conception (et accélérer l'étape d'édition) de ces tâches, les patterns permettent d'intégrer à un modèle de tâches des procédures standards (comme par exemple la procédure à réaliser pour s'identifier) [10, 37]. Leur utilisation permet alors de produire une procédure conforme aux habitudes des acteurs (procédure indépendantes du domaine applicatif). Ces deux derniers outils sont dédiés à des utilisateurs connaissant à la fois le formalisme utilisé et la conception informatique.

2.2 Validation et vérification auprès des opérateurs

La vérification vis à vis du domaine engendre généralement des corrections sur le modèle (itérations de l'édition). Nous considérons comme ressources liées au domaine les documents (notes, interviews, enregistrements) qui apportent des connaissances sur l'activité que le concepteur modélise et les experts de l'activité. La vérification vis à vis du domaine nécessite, de la part du concepteur, une activité cognitive importante. D'une part, celui-ci doit interpréter les ressources (les comprendre et les analyser) et d'autre part, il doit réévaluer son modèle afin de confronter ce qu'il a exprimé (et non ce qu'il croit avoir exprimé) avec ce qu'il a compris de l'activité. Une aide proposée à la réalisation de cette tâche est un changement de point de vue sur le modèle. La plus largement présente dans les éditeurs est l'outil de simulation [4, 19]. Celui-ci propose une vue dynamique du modèle qui permet d'accomplir un déroulement

particulier des tâches (un scénario) en s'appuyant sur les opérateurs d'ordonnancement.

Lorsque la vérification de l'activité modélisée est réalisée par les experts du domaine s'ajoutent les difficultés de compréhension de la notation utilisée [19, 24, 32]. Tout comme pour la vérification par les concepteurs, s'appuyer sur la dynamicité du modèle (au travers des outils de simulation) est l'approche la plus couramment proposée. Certains simulateurs [13, 18] proposent d'augmenter le changement de point de vue sur la sémantique (obtenue par la simulation) par un changement de représentation de celle-ci en développant une présentation non arborescente du scénario déroulé (ou en cours d'exécution). AMBOSS [13] propose une vue graphique d'un déroulement basée sur une frise chronologique alors que K-MADe (avec l'outil prototask [18]) focalise la présentation sur la (ou les) tâche(s) en cours d'exécution et sur les informations qui s'y rapportent.

L'outil de simulation permet d'évaluer si le modèle édité prend en compte un déroulement particulier de l'activité (ou inversement interdit un déroulement). Il est donc plus facile de détecter un déroulement qui ne devrait pas être permis (ie d'invalider le modèle) que d'avoir une vue d'ensemble du modèle qui permette d'en juger la complétude et la validité.

2.3 Bilan de l'état de l'art

Bien que le modèle de tâches est souvent le point de départ de la conception, notamment dans les approches basées modèles [5], sa conception reste une étape difficile à réaliser [12] (coûteuse en temps et en ressources, difficile à valider et difficile à construire). De nombreux travaux cherchent à faciliter cette conception en proposant aux concepteurs du modèle de tâches des formalismes, des éditeurs, des outils de vérification. Les opérations accomplies pour modifier un modèle de tâches sont inégalement supportées par les outils. Le plus grand nombre d'outils proposés s'adressent aux difficultés liées à l'édition du modèle. Les éditeurs développés, pour tirer partie des propositions de solution, intègrent des simulateurs lorsque le langage de modélisation est formel.

Les simulateurs dynamiques sont le principal support proposé à l'évaluation du modèle. Ils peuvent être utilisés à la fois par le concepteur du modèle pour "relire" sa production et par l'expert du domaine qui peut vérifier que des déroulements particuliers sont inclus (ou exclus). De plus, les simulateurs peuvent être utilisés comme support aux échanges entre les concepteurs du modèle de tâches et les experts du domaine [17] ou entre les membres de l'équipe de conception [31].

Lors de la simulation, les concepts du formalismes sont présentés différemment à l'utilisateur (concepteur du modèle de tâches et/ou expert du domaine) proposant une lecture de l'activité par une succession d'ensembles de tâches possibles. Bien que présentée différemment, la nature des concepts (tâche, optionalité...) nécessite d'être comprise. Or, lorsque l'objet d'intérêt est l'information contenue dans le modèle, toute autre connaissance à acquérir est une difficulté supplémentaire. Le problème de l'accès aux connaissances contenue dans le modèle de tâches est d'ailleurs abordé sous l'angle des modèles dans différents travaux [5, 28]. Cependant, la conception d'un système interactif peut nécessiter d'échanger autour de l'activité sans nécessairement avoir comme objectif la réalisation (ou la validation) d'un autre modèle [9]. Par exemple, le texte bien que non formel, est déjà utilisé dans les équipe de conception (sous la forme de scénario [6] ou de user stories [33]) en raison de sa facilité d'accès par tous les membres de l'équipe.

L'approche par simulation propose une vue du modèle qui illustre, en l'animant, la sémantique des opérateurs d'ordonnancement et donc, donne accès aux informations exprimées dans le modèle à un public plus large que celui composé uniquement des concepteurs du modèle de tâches. Cependant, elle ne présente qu'une vue partielle du modèle (car obtenue à travers un nombre fini de cas).

Un support à la compréhension de l'ensemble du modèle sans connaissance du formalisme est donc un outil qui manque pour la réalisation de la vérification du modèle de tâches mais aussi pour faciliter les échanges entre les membres de l'équipe de conception autour de l'activité modélisée. Nos travaux proposent d'utiliser une représentation textuelle de l'ensemble de l'activité modélisée pour présenter une vue globale du modèle qui soit accessible pour l'ensemble de l'équipe de conception.

3 GÉNÉRATION DE LA PRÉSENTATION TEXTUELLE

3.1 Processus de GAT

La génération d'un texte à partir d'une représentation sémantique se fait généralement en trois grandes étapes [35] : 1) La sélection des éléments du discours et leur organisation ; 2) la génération des phrases — syntaxe, lexicalisation (choix du vocabulaire) et l'agrégation de phrases – et 3) la réalisation de surface et le formatage.

Par contraste avec la plupart des systèmes de GAT, dans notre cas, l'entrée du système de GAT est complètement définie (modèle de tâches), complète (toute l'information est présente) et sémantiquement valide (pas d'incertitude ni d'erreur). De ce fait, la réalisation de l'étape 1 consiste à choisir dans le modèle de tâches, les concepts qui sont pris en compte. nous avons considéré les concepts les plus communs aux différents formalismes des modèles de tâches : la décomposition hiérarchique (but/sous-but), le nom des tâches, les opérateurs d'ordonnancement (séquentiel, alternatif, parallèle et sans ordre), le type de tâches (utilisateur, interactif, système et abstrait), l'optionalité et la répétition des tâches. Dans notre démarche, nous considérons que le modèle conçu est conforme au formalisme et aux conventions d'usage (par exemple, une tâche mère ne peut être décomposée en une seule sous-tâche).

L'étape 2 de génération des phrases s'appuie sur la décomposition hiérarchique exprimée par l'arbre des tâches. Chaque tâche décomposée donne lieu à une ou deux phrases. Pour produire ces phrases, une approche par *template* a été employée. L'ordre de génération de ces "blocs" de phrases suit l'ordre de numérotation des tâches décomposées ("profondeur d'abord"). La première phrase du bloc (la phrase obligatoirement présente pour une tâche décomposée) repose sur la logique de but/sous-buts et exprime : la décomposition hiérarchique, le nom des tâches, les opérateurs d'ordonnancement, le type des tâches et l'optionalité. Elle résultat de l'application du template de décomposition : `Pour NOM_TACHE, {sujet(TYPE_T_FILLE) [OPT_T_FILLE] verbe(NOM_T_FILLE) Coord(OPERATEUR_TACHE)}+`

Lorsqu'au moins une des sous-tâches est itérative, une phrase est ajoutée après la phrase de décomposition qui suit le template de répétition. Celle-ci a pour sujet le nom de la (ou des) sous tâche(s) itérative(s) et le groupe verbale est *pouvoir être répétée*.

La lexicalisation consiste à choisir les mots pour chaque élément du discours. Dans cette étude, les choix dépendent des caractéristiques des tâches et sont effectués au niveau des constituants. La table 2 donne les règles de lexicalisation adoptées pour la lexicalisation des constituants (Sujet(), Verbe() et Coord()) en fonction des caractéristiques de la tâche pour le template de décomposition. Dans le cas de tâches filles optionnelles, l'adverbe *optionnellement* est ajouté dans la phrase entre le verbe traduisant la tâche fille et ses compléments.

Table 2: Règles de lexicalisation des constituants du template de décomposition

Règles	Caractéristiques	Traduction
Verbe()	Nom de la sous-tâche	verbe et compléments d'un groupe verbal
Sujet()	Type de la sous-tâche	utilisateur = "l'utilisateur sans logiciel", système = "le système", abstraite = "on", interactive= "l'utilisateur"
Coord()	Opérateur d'ordonnancement	séquentiel = "et", alternative= "ou", parallèle="et en même temps", sans ordre= "et sans ordre"

La sortie de la lexicalisation est un arbre syntaxique par phrase. Par exemple, pour une tâche mère *Acquérir des connaissances en IHM* ayant deux tâches filles séquentielles, l'une abstraite *Suivre le cours d'IHM* et l'autre intéractive *Valider ses connaissances*, l'arbre syntaxique obtenu après l'étape 2 est celui de la **Fig. 1**.

L'étape 3 fléchit (i.e., fait les accords et génére le texte final) cet arbre syntaxique pour donner la phrase : "Pour acquérir des connaissances en ihm on suit le cours d'IHM et l'utilisateur valide ses connaissances."

3.2 Implémentation

L'ensemble de l'implémentation du système a été réalisé en JAVA.

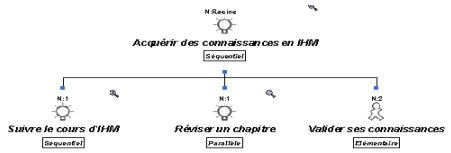
L'étape 1 de sélection et d'organisation étant assurée par le modèle de tâches, l'implémentation de cette étape consiste en la lecture du fichier de sauvegarde du modèle. Bien que les principes peuvent être appliqués aux formalismes ayant les concepts détaillés précédemment, nous avons considéré des fichiers de sauvegarde des modèles de tâches K-MADe [3]. Nous avons choisi ce formalisme car les fichiers de sauvegarde sont au format XML et donc sont facilement interprétables pour l'étape 1 par l'utilisation de la librairie JDOM [2].

Aucune bibliothèque n'a été utilisée pour l'étape 2 (la génération de phrases). L'implémentation que nous en avons faite suit le processus décrit ci-dessus, nous avons réalisé la génération de surface (par SimpleNLG-EnFr 1.1 [3]) en même temps que cette étape. L'arbre syntaxique n'est donc pas explicitement réalisé, notre implémentation réalise directement la phrase avec les accords et les conjugaisons appliquées. Dans notre implémentation, les agrégations sont faites afin d'éviter les répétitions des conjonctions de coordination. La table 3 montre le résultat de la génération de surface pour le cas des trois tâches séquentielles.

L'implémentation a été faite pour permettre l'évaluation de l'approche, aucune évaluation sur la complexité du code (ni linguistique, ni calculatoire) n'a été réalisée. De plus, il est a noté que pour permettre une plus grande flexibilité de l'approche (utilisation de modèles de tâches en XML mais d'un autre formalisme, choix de niveaux de description différents en fonction de la tâches...), l'implémentation doit être adaptée pour séparer la réalisation des étapes 2 et 3.

La table 3 présente les entrées (modèle de tâches K-MADe) et sortie (phrase) du système. Cet exemple illustre bien le focus de la génération sur la fidélité au modèle et l'absence d'ambiguïté.

Table 3: Exemple de génération de phrase

Représentation graphique K-MADe	
Phrase produite	Pour acquérir des connaissances en ihm on suit le cours d'IHM, on révise un chapitre et l'utilisateur sans le logiciel valide ses connaissances.

4 PROTOCOLE D'ÉVALUATION DE L'UTILITÉ DE LA PRÉSENTATION TEXTUELLE

Nous avons voulu évaluer l'utilisation de la présentation textuelle de l'activité pour (1) le concepteur du modèle de tâches qui doit évaluer son modèle et (2) des concepteurs du SI qui n'ont pas fait le modèle de tâches. Cette étude est composée de deux évaluations (une pour chacune des situations).

L'évaluation A (évaluation du modèle par le concepteur de celui-ci) a pour objectif de répondre à l'hypothèse que la présentation textuelle, en apportant un autre point de vue, permet d'identifier des erreurs de conception supplémentaires (**Hypothèse A**).

L'évaluation B (compréhension du modèle par des concepteurs de SI n'ayant pas fait le modèle) est une évaluation de comparaison de la compréhension d'une même activité présentée sous formes graphique et textuelle. L'hypothèse évaluée est que la présentation textuelle permet de mieux comprendre l'activité que la représentation graphique (**Hypothèse B**).

Pour compléter l'évaluation de ces hypothèses, nous avons voulu comparer le point de vue des participants sur l'utilisation des deux formes (**Préférences utilisateur**). Les participants aux deux évaluations ont participé à cette phase.

Les participants à cette étude sont tous des étudiants. L'ensemble de l'étude a eu lieu pendant le second semestre de l'année universitaire 2016-2017 dans le cadre de l'enseignement de la conception

2. http ://www.jdom.org
3. https ://github.com/rali-udem/SimpleNLG-EnFr

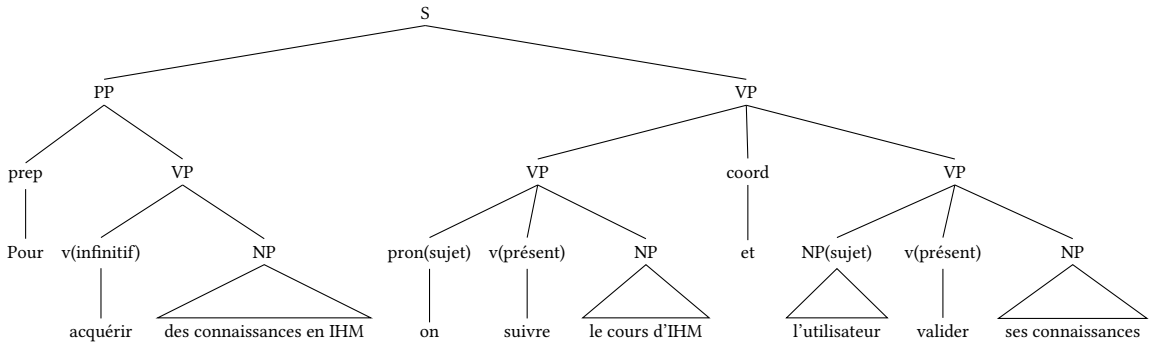


Figure 1: Arbre syntaxique

d'IHM dans lequel est inclus les modèles de tâches. Le logiciel utilisé lors de cette formation est K-MADe [3]. L'enseignement est dispensée en français.

4.1 Évaluation de l'hypothèse A

4.1.1 Participants. Pour la première étude, les participants (24 femmes et 16 hommes) sont les étudiants en 3ème année de licence de technologie pour la santé. Ils sont répartis en 6 groupes de 6 à 7 étudiants ayant chacun un projet de logiciel interactif à produire (sujet libre).

4.1.2 Déroulement. L'évaluation est intégrée aux projets que les étudiants réalisent pour mettre en pratique les connaissances apportées par le cours (pédagogie "par projet"). Par groupe, ils doivent concevoir une IHM (en appliquant plusieurs itérations). Il leur est imposé de réaliser au moins un modèle de tâches lors de la première itération (à partir d'interviews et d'observations qu'ils réalisent).

Avant cette réalisation, ils reçoivent un enseignement théorique sur la modélisation de l'activité, suivi (3 jours plus tard) d'une démonstration interactive de l'utilisation de K-MADe. Cette démonstration est l'occasion pour l'enseignant de revenir sur les concepts de tâche, de décomposition hiérarchique et temporelle et sur les décorations (type de tâches, optionalité, itération). Une fois un premier modèle commun terminé (d'après les étudiants), les outils de vérification de grammaire et de simulation (vue arborescente et prototask) sont utilisés par l'enseignant pour mettre en avant la détection d'erreurs de modélisation (comme l'utilisation de mauvais opérateurs ou l'oubli d'itération).

Chaque groupe de projet travaille ensuite à l'élaboration de leurs modèles de tâches édités avec K-MADe. Le nombre de modèle de tâches à produire n'est pas fixé par les enseignants. Cinq des six groupes en ont produit un seul alors que le groupe 4 (G4) a fait le choix d'en produire deux de petite taille. L'élaboration des modèles peut être faite sur plusieurs semaines. Une fois leurs modèles réalisés (considérés comme terminés - vérification de la grammaire et simulation incluses - par les membres du groupe), celui-ci est récupéré par leur enseignant (via messagerie électronique). Celui-ci génère alors automatiquement la description textuelle du modèle qui est remise au groupe à la séance suivante. Après en avoir pris connaissance, le groupe identifie les éventuelles incohérences entre leur modèle de tâches et le texte généré. L'enseignant n'intervient pas lors de la détection d'incohérences. Les participants sont libres de réaliser la recherche d'erreurs et leur correction en suivant le processus qu'ils souhaitent. Ainsi, certains groupes ont choisi de constituer un sous-groupe pour ce travail alors que d'autres ont laissé l'auteur principal du modèle réaliser cette tâche seul. Lorsqu'ils relevent des incohérences, l'enseignant les aide à identifier la cause et une autre version des modèles est alors produite.

4.1.3 Données récoltées. Pour cette hypothèse, trois types de données sont récoltées : les modèles de tâches produits par les groupes d'étudiants avant correction, l'ensemble des incohérences détectées par les étudiants en autonomie et les notes d'observation prises par l'enseignant lors des échanges avec les concepteurs des modèles de tâches pendant la correction de celui-ci. Ces notes constituent des données qualitatives et n'ont été prises par l'enseignant que pour des situations originales et ce, de manière opportuniste. Pour chacune des observations notées, l'enseignant indique le groupe d'étudiant, l'étape (détection d'incohérences ou correction) et une description.

4.2 Évaluation de l'hypothèse B

4.2.1 Participants. Pour l'évaluation B, les 19 participants (2 femmes et 17 hommes) sont étudiants en 4ième année d'informatique (M1). Ils participent individuellement à l'étude.

4.2.2 Déroulement. Cette évaluation (compréhension de l'activité représentée sous forme graphique vs. textuelle) est réalisée pendant une séance qui lui est dédiée. Avant celle-ci tous les participants ont été formés aux modèles de tâches. L'ensemble des participants a été scindé en deux groupes (de 9 et 10 participants), chacun ayant une version du modèle (graphique pour les membres du groupe 1 et textuelle pour les membres du groupe 2). L'activité choisie est "Acquérir des connaissances en IHM". Les caractéristiques du modèle "Acquérir des connaissances en IHM" sont présentées dans le table 4. Ce modèle a été choisi car il exprime une activité dont les étudiants sont connaisseurs (puisqu'eux même acteurs) et qu'il contient les éléments qui sont pris en compte lors de la génération (comme l'itération et l'optionalité). Une fois que la situation dans laquelle ils doivent se projeter leur a été explicitée,

ils doivent répondre à des questions sur l'activité dont ils ont une représentation.

Table 4: Texte généré du modèle de tâches "Acquérir des connaissances en IHM"

Texte généré **236 mots en 12 phrases pour représenter 22 tâches et une profondeur max de 6 niveaux**
Pour acquérir des connaissances en IHM, on suit le cours d'IHM, on révise un chapitre et l'utilisateur sans logiciel valide ses connaissances. Suivre le cours d'ihm et réviser un chapitre peuvent être répétés. Pour suivre le cours d'ihm on vient en cours et on apprend. Pour venir en cours l'utilisateur a le planning (heure et jour) de cours et l'utilisateur sans logiciel va dans la salle. Pour apprendre on comprend les concepts, et en même temps l'utilisateur sans logiciel fait optionnellement évaluer ses connaissances. Comprendre les concepts peut être répété. Pour comprendre les concepts on a la théorie et on met en application sur un projet. optionalité Pour avoir la théorie l'utilisateur sans logiciel écoute, et en même temps l'utilisateur sans logiciel prend en notes. Pour mettre en application sur un projet on conçoit avec les méthodes du cours, l'utilisateur sans logiciel présente optionnellement son travail et l'utilisateur sans logiciel synthétise les connaissances acquises. Pour concevoir avec les méthodes du cours l'utilisateur sans logiciel élabore et l'utilisateur utilise optionnellement un outil. Pour réviser un chapitre l'utilisateur reprend le cours, l'utilisateur sans logiciel relit ses notes, et en même temps l'utilisateur sans logiciel fait des exercices. Relire ses notes et faire des exercices peuvent être répété.

4.2.3 Données récoltées. Pour l'évaluation de l'hypothèse B, les données analysées sont les réponses au questionnaire complété par les participants. Le questionnaire (nommé dans la suite par "questionnaire de compréhension") est composé de 6 questions sur l'activité exprimée (sous forme graphique et de textuelle). Les 6 questions ont pour but de donner des indicateurs de compréhension selon 3 points. Tout d'abord la compréhension de la forme (texte ou graphique) indépendamment de ce qu'elle représente. Deux questions sont posées pour évaluer ce premier niveau de compréhension. La question 1 est une question à choix multiple sur une échelle de 3 valeurs sur laquelle les participants indique leur niveau de compréhension de ce qui leur est présenté (graphique ou texte). La seconde question vise à connaitre l'avis des participants sur les connaissances initiales que doivent avoir un utilisateur de la présentation qu'ils évaluent. Pour cette question, les participants se prononcent sur la compréhension de 5 profils d'utilisateur potentiels dont les niveaux de connaissance sur l'activité modélisée et sur les modèles de tâches varient. Les autres questions de compréhension porte sur la compréhension de l'activité exprimée ; deux questions sur la compréhension globale de l'activité et deux questions sur la compréhension de la décomposition but/sous-but, de l'optionalité, de l'itération et de l'ordonnancement séquentiel et parallèle. Les deux questions pour évaluer la compréhension de concepts du modèle de tâches (décomposition but/sous buts, optionalité, itération, séquence et parallèlisme) se présentaient toutes deux de la même manière. En fonction d'une situation donnée dans l'énoncée (Q3 : "pour faire *ceci*" et Q4 : "après avoir fait *ceci*"), les participants doivent indiquer si les tâches peuvent être réalisées (Toujours, Parfois ou Jamais). Les réponses possibles sont l'ensemble des tâches du modèle. Enfin, la compréhension globale de l'activité a été abordée au travers d'une question ouverte pour identifier l'acteur de l'activité et d'une question dans laquelle les participants doivent se prononcer sur l'inclusion dans le modèle de tâche de 6 scénarios (vraisemblable dans un contexte d'acquisition de connaissances en IHM).

La table 5 synthétise l'objectif et le type de questions pour le questionnaire de compréhension utilisé pour l'évaluation de l'hypothèse B. 9 participants ont répondu aux questions à partir de la version graphique et 10 participants ont répondu aux questions à partir de la version textuelle du modèle.

Table 5: Questionnaire de compréhension

Dimension	Num	Thème	Réponse possible
Forme	Q1	Niveau de compréhension du participant	Echelle de 3 valeurs
Forme	Q2	Niveau de connaissances	Oui/Non pour différents profils
Détaillée	Q3	But/Sous-buts, optionalité	Toujours/Parfois/Jamais pour chaque tâche
Détaillée	Q4	Itération, séquence, parallèlisme	Toujours/Parfois/Jamais pour chaque tâche
Globale	Q5	Inclusion de scénarios	Oui/Non pour 6 scénarios
Globale	Q6	Acteur	Ouverte

4.3 Préférences des utilisateurs

4.3.1 Participants. Les participants à l'évaluation des préférences des utilisateurs sur la forme (graphique ou textuelle) est l'ensemble des participants des évaluations des hypothèses A et B. Les concepteurs des modèles (participants de l'évaluation A) ont exprimés leurs préférences en groupe (une réponse exprimée pour un groupe) alors que les non-concepteurs (participants de l'évaluation B) ont exprimés leurs préférences individuellement (une réponse par participant).

4.3.2 Déroulement. Après avoir suivi le protocole pour les évaluations des hypothèses A et B, un questionnaire supplémentaire est distribué aux participants (nommé questionnaire comparaison dans la suite). Les participants à l'évaluation de l'hypothèse B ayant préalablement reçu les deux versions du modèle.

4.3.3 Données récoltées. Le questionnaire comparaison est composé de 8 questions ouvertes abordant les points positifs, les points négatifs, les améliorations possibles et enfin, les informations les mieux présentées par le texte et par la représentation graphique.

4.4 Limitations

Le protocole de cette étude comporte des choix qui limitent la portée des résultats. Les participants sont tous des étudiants au début de leur apprentissage des méthodes d'IHM. Si, d'après [7] la formation au modèle de tâches est souvent courte, le recul sur le formalisme

nécessaire à l'identification des erreurs dans un modèle va être différent pour un expert réalisant régulièrement des modèles de tâches ou pour un novice en cours d'apprentissage. Certains de nos résultats sont discutés vis à vis de cette limitation. La participation de ce profil de participants se veut comme une simulation de situation réelle dans laquelle les membres de l'équipe ne sont pas des experts.

Une autre limitation est la durée de l'étude, plus particulièrement la durée d'accès à la représentation textuelle. Les textes sont fournis par les enseignants avant de recueillir les données (réponses aux questionnaires et identifications d'erreurs) alors que la représentation graphique est elle, disponible (et utilisée) en libre accès. Les participants à l'évaluation A n'ont pas fait de remarques sur cette différence mais lors de l'évaluation B, certains des participants ont exprimé leur préférence pour la forme graphique car ils y étaient plus accoutumés. Pour anticiper cette difficulté, nous avions pris soin de réaliser l'évaluation B juste après la présentation de l'éditeur K-MADe. Cependant, la présentation théorique des modèles suivant le formalisme K-MAD a eu lieu la semaine précédente. Cette formation de 3 heures a pu être suffisante pour certains d'entre eux. Il faut également prendre en compte que les participants à l'évaluation B sont étudiants en informatique et qu'ils ont, au cours de leur cursus pris l'habitude d'utiliser les représentations graphiques des modèles (comme les diagrammes de classes d'UML). Si la préférence a été exprimée oralement, nous n'avons cependant pas identifié dans nos résultats d'impacts ni sur la compréhension, ni sur la correction des modèles.

5 RÉSULTATS

5.1 Synthèse des réponses

L'évaluation A a permis de récolter 6 jeux de données (un par groupe projet) chacun composé de un ou deux (pour G4) modèles de tâches initiaux (modèles de tâches produits avant production de la présentation textuelle), d'autant de questionnaires d'incohérence et des notes d'observation pris par l'enseignant.

Ces données nous permettent d'identifier le nombre et le type d'erreurs que chaque groupe de participants a détecté après lecture de la version textuelle de leur(s) modèle(s). Quatre auteurs de modèles (sur les 7 modèles produits) ont identifié seuls des erreurs après lecture du texte. Trois types d'erreurs ont été corrigées par les participants : la décomposition but/sous-buts (3), les opérateurs utilisés (1) et la présence de deux tâches identiques n'ayant pas le même nom (1).

L'évaluation B a permis de récolter des données sur les connaissances que les utilisateurs doivent avoir pour comprendre l'une ou l'autre des présentations des modèles de tâches (d'après les participants) ainsi que sur les erreurs de compréhension du modèle. 5 participants ayant eu la représentation textuelle déclarent avoir totalement compris le texte et 5 déclarent l'avoir partiellement compris. Parmi les 9 participants ayant eu la version graphique, 6 déclarent l'avoir totalement comprise et 3 partiellement. Pour pouvoir comprendre le modèle de tâches il faut connaitre les modèles de tâches pour 7/9 des participants ayant eu la version graphique et pour 4/10 des participants ayant eu la version textuelle et il faut connaitre l'activité pour 1/9 des participants ayant eu la version graphique et pour 0/10 des participants ayant eu la version textuelle. Le nombre de réponses correctes en fonction de la modalité de présentation (graphique ou textuelle) est indiqué pour chaque point évalué dans la table 6. Certains concepts ont été évalués sur plusieurs questions (car devaient être pris en compte pour donner des réponses à plusieurs questions), dans ce cas, nous avons considéré que le participant avait compris le concept lorsque toutes les réponses de toutes les questions étaient cohérentes vis à vis de celui-ci (et correctes). Les concepts concernés sont la séquence, l'optionalité et l'itération. En suivant cette définition, le concept de but/sous-buts n'est compris par aucun des participants. Il était à utiliser pour donner 9 réponses dans les questions 3 et 4 donc 18 fois par participant. En moyenne, les participants l'ont correctement appliqué 15/18 (même moyenne quelle que soit la représentation).

Table 6: Nombre de réponses correctes en fonction de la présentation

	Graphique (/9)	Textuelle (/10)
Itération	2	6
optionalité	5	8
Séquence	0	5
Parallèlisme	7	8
Acteurs	7	5
Détection de non inclusion de scénarios	2	2
Détection d'inclusion de scénarios	8	5

5.2 Analyse

L'hypothèse A est que la présentation textuelle permet aux concepteurs du modèle de tâches de détecter des erreurs de modélisation supplémentaires. En plus, des problèmes de modélisation (section 5.2.1), l'évaluation A nous a permis d'identifier que ce mode de présentation permettait également de détecter des oublis dans les cas pris en compte (section 5.2.2). Les résultats de l'évaluation de l'hypothèse B (la représentation textuelle permet de mieux comprendre l'activité que la représentation graphique) est présentée en section 5.2.3. Enfin, la section 5.2.4 présente un bilan des préférences exprimées par les participants sur le mode de représentation.

5.2.1 Problèmes de modélisation identifiés par les concepteurs du modèle (Évaluation Hypothèse A). Les trois erreurs de décomposition but/sous-buts qui ont été décelées à la lecture des modèles suivaient le même schéma. La tâche décomposée avait pour sémantique la création d'un objet (compte, projet...) et les sous-tâches utilisaient cet objet (l'éditer, l'ajouter à une liste...). La décomposition qui était éditée traduisait donc une relation de séquence. A la lecture, la formulation "Pour créer un objet O, on utilise O..." ne correspondait pas au modèle mental de l'activité que les participants voulaient transcrire ce qui leur a fait revoir leur modèle. Cette erreur est explicable par le profil des participants à l'évaluation (étudiants). Même s'il est possible que des experts ne la commettent pas, sa détection montre que le support textuel a permis aux concepteurs novices d'appréhender la sémantique des liens de composition de leur modèle. Une fois leur erreur comprise, les trois modèles ont été

corrigés (ajout de la tâche de création en séquence) sans le support de l'enseignant.

La seule erreur sur les opérateurs d'ordonnancement qui a été détectée (parallèle au lieu d'alternatif) a été expliquée comme étant une erreur d'édition (ie de manipulation du logiciel). La lecture du texte généré leur a permis de la voir et de la corriger.

Enfin, dans un modèle, en lisant le texte, le concepteur s'est aperçu qu'il avait décrit la même procédure pour deux tâches de noms différents. Dans la représentation graphique, ces deux sous-arbres étaient éloignés l'un de l'autre mais la lecture de deux phrases pratiquement identiques lui a fait comparer les deux procédures. Il a ensuite corrigé le modèle pour que les deux tâches décomposées aient le même nom.

5.2.2 Ajout de cas pour les autres membres. A partir des mêmes données que pour l'évaluation de l'hypothèse A, nous avons observé des modifications du modèle déclenchées par la lecture du texte mais dont le but n'était pas de corriger des erreurs. Dans le groupe 6, une itération de tâche a été identifiée comme manquante par un membre de l'équipe de conception qui n'avait pas participé à l'édition du modèle de tâches. Avant d'envoyer le modèle à l'enseignant le concepteur du modèle l'avait présenté à tous les membres et fait des modifications en fonction de leurs commentaires. Cependant, après lecture du texte généré, l'un des membres du groupe s'est aperçu que leur modèle imposait l'identification (interactive) avant chaque modification de dossiers ("s'identifier" et "modifier un dossier" étant en séquence), or cela ne correspondait pas à ce qu'ils voulaient exprimer dans le modèle. Après explication du problème à l'enseignant, ils ont corrigé la tâche "modifier un dossier" pour la rendre itérative.

Cet ajout est à rapprocher d'autres modifications faites par trois autres groupes de projet (G2, G4 et G5). Après lecture, le concepteur du modèle ou d'autres membres ont vu la nécessité d'étendre le modèle de tâches pour y inclure soit d'autres moyens de réaliser une tâche, soit un sous arbre entier pour une partie de l'activité qu'ils n'avaient pas du tout pris en compte. Si les erreurs de décomposition et d'opérateurs ont toutes étaient identifiées par les auteurs du modèle (section 4.3.1), la prise en compte de nouveaux cas (y compris par itération) a émergé des autres membres de l'équipe (sauf pour un cas). Cette observation semblerait montrer la facilité d'accès au modèle via le texte pour les non concepteurs de celui-ci.

5.2.3 Compréhension (Évaluation Hypothèse B). La première question à laquelle nous avons cherché à répondre pour évaluer cette hypothèse est que la version textuelle soit comprise par les lecteurs. Aucun des participants de l'évaluation B n'a répondu ne rien comprendre du texte. Cependant les niveaux de compréhension du texte ou de la version graphique varient. Nous n'avons pas de données sur ce qui n'a pas été compris et les données récoltées ne permettent pas d'observer des différences significatives de compréhension entre les formes. Néanmoins, l'activité modélisée était connue de tous et connaitre l'activité ne semble pas être nécessaire pour comprendre le modèle. De plus, d'après les réponses des participants, connaitre les concepts de la modélisation des tâches semble être moins important pour comprendre le modèle lorsque celui-ci est sous forme textuel (6/10) que sous sa forme graphique (2/9). Donc, la cause des incompréhensions semble bien être la forme de présentation des informations.

De plus, lorsque les participants sont interrogés sur les points positifs de le présentation textuelle vis à vis de la présentation graphique, spontanément 9/19 participants (dont 3/5 de ceux ayant déclaré ne le comprendre que partiellement le texte) indiquent qu'il peut être compris par n'importe qui ou qu'il ne nécessite par d'apprentissage.

Dans un second temps, nous avons comparé la correction des réponses aux questions sur l'activité modélisée apportées par les participants à partir de la représentation graphique (9) et à partir de la représentation textuelle (10). Aucune différence significative n'a été mise en évidence en fonction de la présentation du modèle qu'ils ont eue.

5.2.4 Préférences des utilisateurs. Enfin, nous avons étudié les réponses aux questionnaires de comparaison, en particulier les deux questions ouvertes sur les informations mieux présentées dans l'une ou l'autre des modes de présentation (**Fig. 2**). D'après tous les groupes de participants, la version graphique du modèle de tâches permet de mieux présenter les liens entre les tâches alors que la représentation textuelle semble présenter les informations sur la tâche (caractéristiques de la tâches indépendantes).

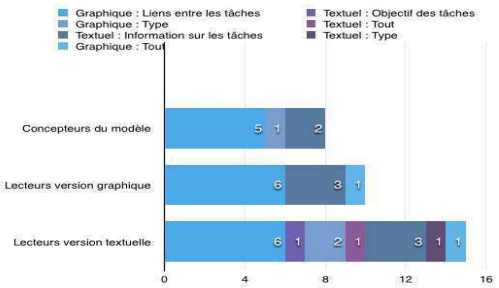


Figure 2: Informations mieux présentées

Au cours de notre étude, les deux évaluations nous ont permis également d'identifier quelques limites à l'approche de présentation textuelle pour la compréhension de l'activité. Tout d'abord, nous avions fait le choix de présenter le texte généré sans aucune mise en forme particulière (chaque phrase débute à la ligne, aucun mot n'est mis en avant d'une manière ou d'autre autre). Cette absence de mise en forme fait partie des points négatifs de la forme textuelle ayant été énoncés par les participants (3/7 de l'évaluation A et 16/19 de l'évaluation B). Cependant, cette absence semble avoir également nuit à la compréhension générale. Par exemple, lors de l'identification d'erreurs de modélisation (évaluation A), nous avons observé un exemple d'erreur non identifiée dans aucune des présentations. L'un des textes présentait une incohérence (séquence modélisée sous forme de lien but/sous-but) que le groupe n'a pas identifié. Plusieurs explications peuvent être imaginées : la forme du texte (aucun travail de mise en forme n'ayant été faite), un manque d'attention ponctuelle des étudiants lors de la lecture ou encore la modalité "écrite" qui n'est pas adaptée à tous. Cependant toutes ce explications vont dans le sens d'un besoin de travail autour de le mise en forme des textes.

En plus de la mise en forme, le style des textes lui-même peut avoir des conséquences sur la compréhension. Nous avions fait le

choix de traiter les répétitions d'opérateurs en sein d'une même phrase par des agrégations. Les propositions verbales étant séparées par des virgules jusqu'aux deux dernières entre lesquelles la conjonction de coordination est spécifiée. Ce choix a été guidée par le souhait d'alléger le style des phrases. Malheureusement il a été la cause, pour deux groupes, d'interrogations sur l'interprétation de ces virgules. Le lien sémantique entre les virgules et les opérateurs (présents en fin de phrase) a eu besoin d'être explicité.

6 DISCUSSION

Les résultats des évaluations montrent que l'impact de la représentation textuelle sur la compréhension et la détection d'erreur est non négligeable. Cette population de novices a déjà été identifiée comme celle bénéficiant le plus d'un support en langue naturelle (cf. le projet BabyTalk [34] où les jeunes docteurs et infirmières réalisaient de meilleures décisions médicales avec une présentation textuelle que graphique). Ces résultats préliminaires sur cette population sont donc encourageants. Cependant, ils doivent être consolidés et surtout la méthode d'évaluation doit être complétée afin de mesurer l'apport du texte sur différents plans. Par exemple, il conviendrait d'identifier le profil des personnes à qui la représentation textuelle bénéficie le plus (développeur, expert, ergonomes, débutant, senior…) à travers des études comparant les performances de groupes. Un autre point d'évaluation important serait de faire une expérimentation contrôlée pour identifier plus précisément le type d'information qui est le mieux transmis graphiquement (p.ex., ascendance) ou textuellement (p.ex., objets de la tâche, ordonnancement). Cette évaluation devrait inclure les autres types de représentation d'un modèle de tâches (p.ex., en formulation logique) afin de vérifier si les apports du langage naturel ne sont pas déjà couverts par d'autres formalismes. Les résultats de cette évaluations permettrait notamment de réaliser un système de génération multimédia où la représentation graphique serait complété d'une description textuelle soulignant les informations plus difficilement accessibles par le graphique. L'utilisateur du modèle de tâches pourrait ainsi passer d'une représentation à l'autre en fonction de ses besoins comme [16] le propose entre les scénarios et l'UI pendant la conception de celle-ci.

Un autre facteur qui influence l'étude est la 'qualité' du texte généré. Si la faible quantité de textes utilisé dans les études permettait d'effectuer une vérification manuelle de la qualité des sorties, il n'en sera pas de même pour une étude plus large. Pour cela, nous comptons intégrer les méthodes d'évaluations des systèmes de résumés automatiques [8] qui mesure la qualité d'un texte par sa conformité à la langue (grammaticale) et par sa cohérence par rapport à l'information transmise (cohérence sémantique). Parmi les mesures à mettre en œuvre le taux d'ambiguïté doit être particulièrement surveillé. En effet, les évaluations ont révélé les différentes interprétations possibles d'une virgule qui peut signifie une succession (séquence) ou un parallélisme (et en même temps). La lexicalisation de ces coordinations doit donc être revue.

Un défi important dans la compréhension d'un modèle de tâches et la taille de celui-ci. La représentation textuelle doit donc également être évaluée par sa capacité à transcrire un modèle de tâches de grande taille. Toutes les représentations (graphiques et logiques) sont affectées par les grandes tailles d'arbres qui rendent le modèle difficile à appréhender. La représentation textuelle peut également être complémentaire par l'utilisation d'un mode de formatage adapté (tabulation, liste à puces, etc.) notamment par la génération d'hypertexte (navigation selon les sous-branches de l'arbre).

Concernant le texte généré, un point marquant est le manque de fluidité du texte et de perspective. Or, il a déjà été souligné que l'aspect narratif des textes générés joue un rôle dans la compréhension et la rétention d'information des destinataires [36]. Plutôt que de décrire le modèle, une possibilité serait donc de 'raconter' un parcours de l'arbre des tâches selon différents points de vue (utilisateur, groupe, système…). Cette technique de narration automatique est encore récente mais accessible à terme à partir de modèles formels et complets comme ceux considérés dans cet article [1]. Le récit (réel ou fictif) créé à partir du modèle de tâches permettrait de s'intégrer à la conception basée sur scénarios. La présentation de différents points de vue serait également plus facile à appréhender par les différents acteurs d'activités collaboratives.

7 CONCLUSION ET TRAVAUX FUTURS

Cet article présente une étude sur l'utilisation du texte écrit comme mode de représentation du modèle de tâche dans le but de faciliter l'accès aux informations du modèle de tâches pour la conception des SI. L'intérêt de ce mode de représentation est de s'abstraire du formalisme de représentation du modèle de tâches pour utiliser le langage naturel comme vecteur de communication d'informations et permettre une compréhension du modèle de tâches quelle que soit l'expertise du lecteur. Nous avons présenté comment un générateur de texte pouvait produire un document textuel décrivant un modèle de tâches. Les évaluations que nous avons menées ont montré que le texte pouvait être une présentation complémentaire à la présentation graphique lors de la conception de SI. Le texte peut être un support aux concepteurs novices pour l'amélioration de leurs modèles (détection d'erreurs ou ajout de cas). Dans cet objectif, nous envisageons d'étendre notre approche pour prendre en compte l'itération des corrections du modèle. Pour cela, nous pensons introduire l'élicitation des modifications faites par les utilisateurs dans les textes générés [22, 38]. La qualité des textes générés et de leurs présentations peut elle aussi être améliorée (mise en forme, gestion de la redondance, références, structure du texte...). Cette qualité de la présentation fera l'objet de nos prochaines études. Enfin, nous considérons que si le texte est une modalité accessible au plus grand nombre, la présentation doit être adaptée aux objectifs du lecteur et à ses caractéristiques (pour prendre en compte, par exemple, son niveau de connaissance sur l'activité).

RÉFÉRENCES

[1] Belén A. Baez Miranda, Sybille Caffiau, Catherine Garbay, and François Portet. 2014. Task based model for récit generation from sensor data : an early experiment. In *5th International Workshop on Computational Models of Narrative*. France, 1–10.

[2] Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis. 2013. Generating Natural Language Descriptions from OWL Ontologies : the NaturalOWL System. *Journal of Artificial Intelligence Research* 48 (2013), 671–715.

[3] M. Baron, V. Lucquiaud, D. Autard, and D. L. Scapin. 2006. K-MADe : Un Environnement Pour Le Noyau Du ModèLe De Description De L'Activité. In *Proceedings of the 18th Conference on L'Interaction Homme-Machine (IHM'06)*. ACM, 287–288.

[4] Sybille Caffiau, Laurent Guittet, Dominique L Scapin, and Loé Sanou. 2008. Utiliser les outils de simulation des modèles de tâches pour la validation des besoins utilisateur : une revue des problèmes. In *ERGO'IA*.

[5] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, L Bouillon, and Jean Vanderdonckt. 2003. A unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computer* 15, 3 (2003), 289–308.

[6] John M. Carroll. 2000. *Making Use : scenario-based design of human-computer interactions*. 368 pages.

[7] Stanislas Couix. 2007. *Usages et construction des modèles de tâches dans la pratique de l'ergonomie : une étude exploratoire*. Technical Report. LEI - Laboratoire d'Ergonomie Informatique - Université Paris 5.

[8] Hoa Trang Dang and Donna Harman (Eds.). 2007. *Proceedings of Document Understanding Conference (DUC) 2007*. Rochester, USA.

[9] Dan Diaper and Neville Stanton. 2003. *The handbook of task analysis for human-computer interaction*. CRC Press.

[10] Jürgen Engel, Christian Märtin, and Peter Forbrig. 2015. A Concerted Model-driven and Pattern-based Framework for Developing User Interfaces of Interactive Ubiquitous Applications. In *LMIS@ EICS*.

[11] Rachel Farrell, Gordon Pace, and Michael Rosner. 2015. A Framework for the Generation of Computer System Diagnostics in Natural Language using Finite State Methods. In *ENLG 2015 - Proceedings of the 15th European Workshop on Natural Language Generation*.

[12] Peter Forbrig, Celia Martinie, Philippe Palanque, Marco Antonio Winckler, and Racim Fahssi. 2014. Rapid Task-Models Development Using Sub-models, Subroutines and Generic Components. In *Human-Centered Software Engineering - HCSE 2014*.

[13] Matthias Giese, Tomasz Mistrzyk, Andreas Pfau, Gerd Szwillus, and Michael von Detten. 2008. *AMBOSS : A Task Modeling Approach for Safety-Critical Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 98–109. DOI : http://dx.doi.org/10.1007/978-3-540-85992-5_8

[14] Dimitra Gkatzia, Oliver Lemon, and Verena Rieser. 2016. Natural Language Generation enhances human decision-making with uncertain information. In *54th Annual Meeting of the Association for Computational Linguistics*.

[15] Frédéric Jourde, Yann Laurillau, and Laurence Nigay. 2014. Description of tasks with multi-user multimodal interactive systems : existing notations. *Journal d'Interaction Personne-Système (JIPS)* 3, 3 (2014), 1–33.

[16] Koki Kusano, Momoko Nakatani, and Takehiko Ohno. 2013. Scenario-based interactive UI design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.

[17] Thomas Lachaume, Patrick Girard, Laurent Guittet, and Allan Fousse. 2011. Prototypage basé sur les modèles de tâches : une étude pilote. In *Proceedings of the 23rd Conference on l'Interaction Homme-Machine*. ACM.

[18] Thomas Lachaume, Patrick Girard, Laurent Guittet, and Allan Fousse. 2012. ProtoTask, new task model simulator. In *International Conference on Human-Centred Software Engineering*. Springer Berlin Heidelberg, 323–330.

[19] Thomas Lachaume, Laurent Guittet, Patrick Girard, and Allan Fousse. 2014. Task model simulators : a review. *Journal d'Interaction Personne-Système (JIPS)* (2014).

[20] Benoit Lavoie, Owen Rambow, and Ehud Reiter. 1996. The ModelExplainer. In *International Natural Language Generation Workshop (INLG-96)*. Herstmonceux Castle, UK, 9–12.

[21] A.S. Law, Y. Freer, J. Hunter, R.H. Logie, N. McIntosh, and J. Quinn. 2005. A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. *Journal of clinical monitoring and computing* 19, 3 (2005), 183–194.

[22] Christophe Lemaigre, Josefina Guerrero García, and Jean Vanderdonckt. 2008. Interface model elicitation from textual scenarios. In *Human-Computer Interaction Symposium*. Springer, 53–66.

[23] Quentin Limbourg, Costin Pribeanu, and Jean Vanderdonckt. 2001. Towards uniformed task models in a model-based approach. In *International Workshop on Design, Specification, and Verification of Interactive Systems*. Springer, 164–182.

[24] Célia Martinie, Philippe Palanque, David Navarre, and Eric Barboni. 2012. A development process for usable large scale interactive critical systems : application to satellite ground segments. In *International Conference on Human-Centred Software Engineering*. Springer, 72–93.

[25] Célia Martinie, Philippe Palanque, and Marco Winckler. 2011. Structuring and composition mechanisms to address scalability issues in task models. In *IFIP Conference on Human-Computer Interaction*.

[26] Célia Martinie De Almeida. 2011. *Une approche à base de modèles synergiques pour la prise en compte simultanée de l'utilisabilité, la fiabilité et l'opérabilité des systèmes interactifs critiques*. Ph.D. Dissertation. http://www.theses.fr/2011TOU30254 2011TOU30254.

[27] N. McIntosh, A.J. Lyon, J. Reiss, J.C. Becher, R. Logie, K. Gilhooley, E. Alberdi, and J. Hunter. 2000. The cognitive processes of doctors and nurses in the interpretation of physiological monitoring data in the neonate. *Early Human Development* 58, 1 (2000).

[28] David Navarre, Philippe Palanque, Fabio Paternò, Carmen Santoro, and Rémi Bastide. 2001. A Tool Suite for Integrating Task and System Models through Scenarios. 88–113.

[29] Donald A Norman and Stephen W Draper. 1986. User centered system design. *New Perspectives on Human-Computer Interaction, L. Erlbaum Associates Inc., Hillsdale, NJ* 3 (1986).

[30] Cecile Paris, Nathalie Colineau, Shijian Lu, and Keith Vander Linden. 2005. Automatically generating effective online help. *International Journal on Elearning* 4, 1 (2005), 83.

[31] Fabio Paternò. 2002. Tools for Task Modelling : Where we are, Where we are headed. In *TAMODIA*. 10–17.

[32] Fabio Paternò, Carmen Santoro, and Lucio Davide Spano. 2012. Improving support for visual task modelling. In *International Conference on Human-Centred Software Engineering*. Springer, 299–306.

[33] Jeff Patton, Claude Aubry, and Dominique Maniez. 2015. *Le story mapping : Visualisez vos user stories pour développer le bon produit*. Dunod.

[34] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artif. Intell.* 173 (2009), 789–816.

[35] Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

[36] Ehud Reiter, Albert Gatt, François Portet, and Marian Van Der Meulen. 2008. The importance of narrative and other lessons from an evaluation of an NLG system that summarises clinical data. In *Proceedings of the 5th International Conference on Natural Language Generation (INLG-08)*. Salt Fork, Ohio, United States, 147–153.

[37] Daniel Sinnig, Maik Wurdel, Peter Forbrig, Patrice Chalin, and Ferhat Khendek. 2007. Practical extensions for task models. In *International Workshop on Task Models and Diagrams for User Interface Design*.

[38] R. Chung-Man Tam, David Maulsby, and Angel R. Puerta. 1998. U-TEL : A Tool for Eliciting User Task Models from Domain Experts. In *Proceedings of the 3rd International Conference on Intelligent User Interfaces (IUI '98)*. ACM, New York, NY, USA, 77–80. DOI : http://dx.doi.org/10.1145/268389.268402

[39] Maik Wurdel, Daniel Sinnig, and Peter Forbrig. 2008. CTML : Domain and Task Modeling for Collaborative Environments. *J. UCS* (2008).

[40] Maik Wurdel, Daniel Sinnig, and Peter Forbrig. 2008. *Task Model Refinement with Meta Operators*. Springer Berlin Heidelberg, Berlin, Heidelberg, 300–305. DOI : http://dx.doi.org/10.1007/978-3-540-70569-7_28