Éviter l'échec des systèmes complexes : en construire collectivement une représentation formelle utile

O. Poitou[1], C. Saurel[1]

[1] ONERA Toulouse, 2 avenue Edouard Belin, 31055 Toulouse, France

prenom.nom@onera.fr

Résumé

Cet article présente nos réflexions sur une nécessaire évolution du processus de construction d'un modèle d'un système complexe. Nous identifions le rôle de transcripteur, ses actions et leur insertion dans un processus de construction incrémentale du modèle. Nous proposons des propriétés permettant d'évaluer la qualité de ce modèle au cours du temps. Nous insistons sur la nécessité de capturer la provenance de ses différents éléments, pour répondre à des exigences de traçabilité, de respect de la confidentialité, et d'explicabilité du modèle résultat.

Mots-clés

Ingénierie des modèles, Modélisation collaborative, Modélisation agile, Fusion d'information, Système Complexe

Abstract

We claim that complex system modeling process needs an evolution. We identify the transcriptor role, associated actions and their orchestration in an incremental collective model contruction process. We propose some properties to evaluate model quality at every moment. We promote a precise capture of model elements provenance, as a way to meet traceability, confidentiality and understandability requirements.

Keywords

System Engineering, collaborative modeling, agile modeling, information fusion, complex systems

1 Introduction

La construction collaborative d'un modèle d'un système complexe [1], ou plus précisément d'une spécification formelle avec représentation graphique, est aujourd'hui mal outillée. L'ingénierie système basée modèles (MBSE), discipline récente, n'est déjà plus directement adaptée ni à la complexité actuelle des systèmes, ni à la gestion de la nécessaire collaboration entre les parties prenantes. Les processus et formalismes qu'elle encourage sont parfois trop rigides : certains détenteurs de données expertes ne savent pas les y exprimer. Ils peuvent ne pas être assez précis : les experts peuvent y décrire des informations ne pouvant être exploitées correctement ni par d'autres intervenants ni par des outils d'analyse [8].

Pour éviter ces écueils et mieux outiller la démarche, nous proposons de la formaliser, à travers des rôles, des actions, et des propriétés attendues ou simplement utiles à l'évaluation de la construction du modèle.

La section 2 introduit l'approche et notamment le rôle central de transcripteur. Ses actions et leur insertion dans un processus de construction du modèle du système sont décrites dans la section 4, après la proposition en section 3 de propriétés permettant d'évaluer la qualité de ce modèle au cours du temps. La section 5 positionne ces travaux dans un rapide état de l'art avant que la section 6 ne conclue.

Les exemples utilisés sont issus d'un cas d'étude du domaine de la maintenance aéronautique introduit dans [11].

2 Problématique, approche et rôles

La MBSE propose un cadre à la capture et la restitution des connaissances sur un système, reposant sur l'utilisation de vues standard. Il se veut suffisamment précis pour que le modèle [2] reflète correctement le système modélisé sur des questions d'intérêt : respect de contraintes de sûreté et de sécurité par exemple, indicateurs de performances… Le processus de construction de modèle décrit par ce cadre, appliqué sur un système complexe, devient collaboratif et incrémental puisque les données expertes ne peuvent être obtenues depuis une source unique [12]. Les grandes étapes classiques d'une démarche MBSE gagnent cependant à être assouplies car la synchronisation des différents intervenants peuvent rendre le processus trop contraignant et éloigné de la réalité du terrain. Nous proposons une démarche plus agile capturant l'information lorsqu'elle devient disponible, et la formalisant en un couple modèle/méta-modèle, que nous nommons ici *représentation formelle* [3], en reposant éventuellement sur des interactions, notamment avec le fournisseur de cette information (cette approche est proche du "free modeling" de [6]).

La construction collaborative d'un modèle fait apparaître

1. Une propriété d'un système dit complexe est qu'une seule personne ne peut le connaître entièrement et parfaitement.

2. Le terme *modèle* utilisé ici correspond au terme *Architecture Description* du standard ISO1471/IEEE42010

3. Habituellement, le méta-modèle est d'abord décrit et validé, puis utilisé comme langage pour exprimer le modèle. Dans une démarche plus agile, le modèle et son méta-modèle sont construits en parallèle au fil des intégration des contributions.

différents rôles. Nous nous intéresserons à trois d'entre eux : transcripteur, contributeur et observateur.

Les *contributeurs* sont les dépositaires des connaissances sur le système réel (ou souhaité). Ils vont tenter de le décrire le plus fidèlement possible pour la partie et les aspects qu'ils en connaissent, à travers une ou plusieurs *contributions* (textes, tableaux, schémas... pas nécessairement formels). Chaque contribution apporte des informations nouvelles, des corrections ou des détails supplémentaires.

Les *observateurs* sont les commanditaires ou participent à la maîtrise d'ouvrage. Ils ont des attentes sur le système et souhaitent être informés sur celui-ci dans le respect d'éventuelles règles de confidentialité. Ils réclament des indicateurs pour éclairer leurs prises de décisions, et une information ciblée et présentée selon leur souhait.

Le *transcripteur* est le rôle central de notre approche. Il construit, au fil du temps, la représentation formelle du système en intégrant des contributions émises par les contributeurs, de manière à produire les indicateurs et documentations satisfaisant les observateurs.

3 Propriétés et indicateurs

Pour exploiter de manière fondée le résultat d'une transcription, les observateurs sont en droit d'exiger que le modèle, son méta-modèle, ou même le processus de transcription vérifient certaines propriétés, comme en Génie logiciel ([13]). Nous définissons ci-dessous un tel jeu de propriétés, non exhaustif. Nous suggérons aussi des propriétés souhaitables pour la restitution d'une transcription.

3.1 Propriétés du processus de transcription

Le processus offre la propriété de *traçabilité sur la provenance* quand il permet de fournir la provenance de chaque élément de tout modèle produit. La provenance d'un élément est décrite par les acteurs ayant contribué à son élaboration, et les opérations de fusion d'informations avec leur date et leurs entrées (contributions, ou autres).

La propriété de *transcription sélective* est offerte par le processus s'il permet d'accéder à des versions partielles de la transcription (ie : en ignorant certaines contributions, selon leur horodatage mais aussi leur provenance ou toute autre caractéristique...). Cette propriété peut devenir nécessaire en cas d'exigences de confidentialité entre parties prenantes du système objet de la transcription. La démarche est alors de générer une représentation formelle ne prenant en compte que les contributions auxquelles un destinataire a accès. Ceci garantit qu'aucun élément confidentiel ne peut être retrouvé à partir d'éléments dérivés, que ce soit à travers le méta-modèle, les contraintes/règles retenues etc. [4]

3.2 Propriétés du modèle transcrit

Nous convenons que pour être exploitable par des humains et des outils, un modèle obtenu par transcription doit vérifier les propriétés suivantes.

Il doit être *formel* : la transcription doit être décrite dans un langage (ou méta-modèle) interprétable par des outils de calcul ou de raisonnement automatique.

Il doit être *non ambigu* : il ne doit pas permettre deux interprétations différentes de la part des observateurs. La présence d'homonymie est un exemple d'ambiguïté.

Il doit être *complet par rapport aux besoins* formulés par les observateurs. Il doit permettre de satisfaire leurs demandes de documentation par génération de vues pertinentes, et de vérifier l'ensemble des propriétés qui les intéressent, par évaluation de leur expression formelle sur la transcription.

Il doit être *conforme au méta-modèle* actuel obtenu à l'issue de la transcription des contributions disponibles (respect des cardinalités, des types, des contraintes utilisateur).

Il doit être *consensuel* au sein des contributeurs : il doit recueillir au moins une proportion minimale d'approbation de contributeurs, tout en ne dépassant pas une proportion maximale de mise en doute.

Un modèle est *modulaire* s'il peut être structuré en unités indépendantes, réutilisables dans diverses applications.

Ces propriétés concernent le modèle, mais implicitement et conjointement, son méta-modèle. Les propriétés suivantes visent plus directement le méta-modèle.

Le méta-modèle souffre de *superfluité* s'il contient des concepts ou des relations qui ne sont pas instanciés dans le modèle : ces concepts ou relations sont superflus. Une version moins binaire de cette propriété peut être envisagée, limitant les concepts dont le nombre d'instances est sous un certain seuil, ce qui peut montrer un niveau de détail excessif du méta-modèle par rapport au modèle à élaborer.

Il encourage la *concision* si le modèle qu'il permet d'exprimer nécessite d'expliciter peu d'informations déductibles à partir d'autres éléments de ce modèle. Par exemple, indiquer la symétrie d'une relation dans le méta-modèle permet de ne pas devoir systématiquement mentionner les liens dans les deux sens dans le modèle. Notons toutefois qu'indiquer la même symétrie en tant que propriété souhaitée, au lieu de règle, permettra par la suite de vérifier qu'un modèle satisfait cette propriété (au lieu de l'imposer). Cet exemple illustre le compromis à trouver entre automatisation, pour éviter la redondance, et exploitation d'une certaine redondance pour vérifier la cohérence.

3.3 Propriétés sur la restitution

Le processus de transcription vise à satisfaire les besoins des observateurs en produisant régulièrement des observations (cf 4.2) via le processus de restitution.

Une restitution est *complète* si l'ensemble des observations produites répond à l'ensemble des besoins exprimés.

Une restitution est *satisfaisante* si la réponse apportée à chaque besoin est acceptable, par exemple en terme de précision et/ou fiabilité.

Une restitution est *concise* si (une grande proportion parmi) les éléments la composant sont pertinents, i.e. nécessaires et suffisants pour répondre au besoin exprimé.

Une restitution est *juste* si l'information qu'elle véhicule est conforme à celle contenue dans le modèle (le processus de

4. Ce risque existe si l'on se contente de filtrer les éléments confidentiels contenus dans une représentation formelle construite à partir de l'intégralité des contributions

création d'une représentation n'a pas introduit de contradiction entre l'information présentée et celle du modèle).

Une restitution est *lisible* si, pour chaque observateur, le résultat lui est exprimé dans un langage, et représenté sous une forme, qui lui sont familiers.

4 Actions des différents rôles

Pendant un processus collaboratif et incrémental, les différents intervenants réalisent des actions. Nous définissons ici les actions sur le modèle ou le méta-modèle pour chaque rôle identifié dans la section 2. Des interactions entre ces rôles surviennent en parallèle. Certaines actions peuvent motiver et guider des interactions. A l'inverse, certaines interactions peuvent déclencher les actions ci-dessous.

4.1 Actions du contributeur

Les contributeurs fournissent successivement au transcripteur des informations qui contribuent à la description du système, selon leur expertise et leur point de vue. Ils expriment ces informations dans leur langage favori (par exemple, des schémas de type BPMN) et selon leurs habitudes [5] au travers de trois actions possibles :

- *ajouter* une nouvelle information,
- *corriger* une information existante,
- *raffiner* une information, en la détaillant.

La démarche du transcripteur vise à obtenir un consensus des contributeurs sur la description finale du système : ils devront donc se prononcer sur l'aptitude du modèle transcrit à traduire la vision de chacun sur le système décrit, avec d'autres actions :

- *approuver* le modèle, intégralement ou en partie,
- *mettre en doute* le modèle, ou une partie de celui-ci.

4.2 Actions de l'observateur

Nous distinguons deux types d'*observations* : les *vues* personnalisées et les *indicateurs*.

Les *vues* personnalisées représentent l'ensemble des informations du modèle concernant une attente particulière (un thème, un niveau de détail, une syntaxe concrète). Les actions à réaliser côté transcription sont la sélection d'informations pertinentes et de leur représentation adéquate.

Les *indicateurs* sont les valeurs associées à des expressions. L'action à réaliser côté transcripteur est principalement le calcul. Ces indicateurs peuvent chercher à évaluer le système à travers son modèle, ou bien le modèle en lui-même.

Nous parlerons de *spécification d'observation* pour la demande formulée par un observateur, tandis que nous utiliserons le terme d'*observation* pour la réponse construite par le transcripteur.

Par interaction avec le transcripteur, l'observateur peut :

- *émettre* une spécification d'observation,
- *corriger* une spécification d'observation ou
- *retirer* une spécification d'observation.

5. Ces habitudes proviennent de leur domaine métier, de la charte de leur organisation ainsi que de pratiques personnelles

4.3 Actions du transcripteur

Le transcripteur est en charge de transformer les contributions en un modèle du système d'intérêt, vérifiant les propriétés souhaitées par les contributeurs et observateurs, et permettant d'évaluer les spécifications d'observations émises par les observateurs. Pour cela, le transcripteur procède à plusieurs actions au cours du temps :

- *déchiffrer* une contribution (cf 4.3.1).
- *interpréter* une contribution déchiffrée (cf 4.3.2)
- *consolider* une contribution interprétée (cf 4.3.3)
- *fusionner* un modèle et des contributions (cf 4.3.4)
- *prendre en compte* les spécifications d'observations (cf 4.3.5)
- *évaluer* la représentation formelle (cf 4.3.6)

4.3.1 Déchiffrer

Parfois, une contribution n'est pas directement utilisable et doit être déchiffrée : décrite dans un format exploitable.

Le déchiffrage comprend l'abstraction de certains éléments, son résultat est modifié par les choix d'abstraction réalisés. Dans l'exemple d'une contribution graphique, le déchiffrage peut décrire des textes dans des rectangles, ou conserver les dimensions, et la position exacte de chacun de ces rectangles contenant du texte.

Pour limiter le risque de perdre des éléments porteurs de sémantique, le déchiffrage devra essayer de conserver le plus d'information possible. Notons que cette action est uniquement une action de perception ; les choix d'interprétation seront faits plus tard. Par exemple, le format de représentation des couleurs sera choisi lors de cette action (représentation symbolique, valeurs RVB…) ; mais la question de la sémantique de ces couleurs sera reportée à l'étape d'interprétation : la couleur pourra être utilisée pour participer au typage des éléments, valuer un attribut, ou encore traduire le niveau de d'abstraction, comme dans le NAF (canevas d'architecture de l'OTAN).

Repartir de l'image de la contribution peut être nécessaire, par exemple si une contribution graphique est fournie dans un format de fichier propriétaire (dont le transcripteur n'a pas l'outil correspondant) et que seules les données image sont lisibles. Néanmoins, une représentation utilisable de la contribution peut parfois être directement lue depuis un fichier dans un format ouvert.

4.3.2 Interpréter

Le transcripteur doit générer une représentation formelle de chaque contribution déchiffrée, en vue d'une intégration dans le modèle en cours d'élaboration : on obtient une contribution interprétée. Toutes ses actions doivent conserver les informations de provenance de chaque élément. Pour cela, le transcripteur va, en parallèle :

- faire évoluer un méta-modèle : langage permettant de représenter le contenu des contributions ;
- exprimer dans ce langage chaque contribution, pour en obtenir une version formelle ;
- lier le méta-modèle et la représentation graphique du contributeur dans une feuille de styles.

Par exemple, un carré contenant "Detect" dans une contribution pourra être interprété au niveau méta-modèle par un

concept Activité, au niveau modèle par une entité "Detect" de type Activité, et au niveau style par un lien entre Activité et forme carrée.

Selon le type de contribution, le transcripteur doit :

- enrichir méta-modèle et modèle pour un ajout ou un raffinement,
- réviser méta-modèle et modèle pour une correction.

La complexité de l'opération de révision dépend des cas de figure. Pour les éléments pour lesquels aucune contribution temporellement intermédiaire n'existe, il suffit de remplacer les éléments concernés des contributions à corriger par ceux de la contribution correctrice : on utilisera les mêmes identifiants internes pour ne pas perdre d'éventuels liens avec des éléments non mis à jour. Pour les éléments pour lesquels une contribution intermédiaire existe, un processus de fusion plus avancé doit être mis en place pour décider de la version finale.

4.3.3 Consolider une contribution interprétée

Ces opérations sont à effectuer sur une contribution interprétée, ou un jeu de contributions interprétées d'un même contributeur. Elles facilitent son intégration dans le modèle. Elles sont à placer dans un contexte où les conventions de notation des langages utilisés par les contributeurs ne sont pas toujours respectées, ou les contributions sont faites avec une notation métier. Le transcripteur doit capturer et restituer l'intention du contributeur.

Le transcripteur aura par exemple à identifier des soupçons :

- de lacunes sur des contributions : par exemple, avec des contributions de type BPMN, manque de liens conditionnels entre activités décrites comme déclenchables sur condition, manque de marque de fin globale dans la description d'un processus...
- d'inadéquation du label d'une entité avec son type selon la convention graphique du contributeur,
- de redondances,
- de non conformités par rapport au méta-modèle.

Il devra ensuite :

- proposer des corrections quand un outil permet d'en élaborer, sinon,
- tracer le doute, rattaché à la contribution, dans l'attente de résolution par d'autres contributions.

Le résultat attendu de ces actions sera

- une *contribution consolidée* grâce aux interactions transcripteur/contributeur,
- des informations de provenance concernant les opérations de consolidation,
- une trace regroupant les arguments associés à un soupçon de problème non résolu (pour optimiser ensuite les interactions avec les contributeurs).

4.3.4 Fusionner un modèle et des contributions

Pour intégrer une contribution interprétée supplémentaire au modèle obtenu par intégration d'autres contributions interprétées, tout en assurant les propriétés requises, le transcripteur devra régler les questions qui suivent. Pour cela, il pourra adapter la contribution pour la rendre conforme au méta-modèle courant, ou faire évoluer ce méta-modèle.

Beaucoup de ces questions concernent le label des éléments qui peuplent les contributions. Le *label* d'un élément est son appellation dans une contribution ; il est à distinguer de son *identifiant*. Nous convenons qu'un élément donné n'a qu'un identifiant dans la représentation formelle ; en revanche il peut se voir associer plusieurs labels au fil des contributions, et selon le vocabulaire de chaque contributeur.

Lorsque le transcripteur complète la représentation formelle en cours d'élaboration, en intégrant une contribution consolidée, il doit :

- détecter et aider à résoudre les problèmes soupçonnés d'*homonymie*, sources d'ambiguïté ;
- détecter les occurrences de *presque-homonymie* et aider à les résoudre si elles viennent d'une erreur ;
- détecter les *synonymies*, car elles peuvent générer des incompréhensions entre observateurs ou contributeurs. Les couples (label, utilisateur) doivent être liés à un unique identifiant ; en outre cela permettra d'améliorer la lisibilité des restitutions, en utilisant le vocabulaire adapté à chacun.

Il y a soupçon d'homonymie lorsqu'un label est partagé par des éléments que le transcripteur soupçonne être différents (par exemple, dans une même contribution, plusieurs occurrences d'actions "Defect fix or defer" associées à deux sous-systèmes distincts). On caractérise ce problème selon : le nombre contributions concernées, le nombre de contributeurs, la différence de nature ou de type des éléments (par exemple, label "defect reported" associé à une condition et à un type de message). Selon ces caractéristiques, les arguments permettant au transcripteur de décider qu'il y a ou pas homonymie diffèrent. Le choix du transcripteur (ne rien changer, ou transformer les labels pour enlever l'homonymie) pourra varier entre une décision unilatérale de sa part, et le recours à des interactions avec le(s) contributeur(s) pour lever l'indécision.

Il y a presque-homonymie en cas de labels aux graphies presque identiques (par exemple, "request" et "resquest" ; "request Work order status" et "Work order status request"). Le traitement consiste à déterminer si ces labels désignent des éléments réellement différents (compte-tenu d'une erreur d'un contributeur). Associés à des entités de nature et de types identiques, des liens de labels presque homonymes suggèrent une erreur de graphie. A l'opposé, le transcripteur peut parier qu'il ne s'agit pas d'une erreur de graphie si les entités sont de nature différente (par exemple, relation et concept). Dans de nombreux cas, le transcripteur sera cependant amené à

- élaborer une correction de la transcription pour n'avoir plus qu'un label,
- la faire valider par les contributeurs si le doute reste trop fort pour le transcripteur, ou
- la tracer pour la faire valider plus tard.

Il y a synonymie lorsque plusieurs labels distincts désignent le même élément (par exemple, Aircraft Maintenance Technician et Line Maintenance Operator désignant le même sous-système dans des contributions différentes). Leur étude permet de traduire des éléments du vocabulaire d'un contributeur dans celui d'un autre contributeur.

Le soupçon de synonymie peut intervenir si deux éléments de labels différents reliés aux mêmes éléments apparaissent dans des contributions de sources différentes (hypothèse de vocabulaires différents selon l'organisation ou le domaine métier), ou d'un même contributeur mais espacées dans le temps (hypothèse de changement de vocabulaire du ou des contributeurs au cours du temps). Un même contributeur peut aussi alterner entre deux désignations d'un même élément (hypothèse de synonymie standard). Des actions possibles du transcripteur consistent alors à :

— noter cette hypothèse de synonymie et surveiller l'évolution de sa vraisemblance en fonction de la prise en compte d'autres contributions,
— en cas de synonymie confirmée, ne garder dans le modèle qu'un identifiant, et la liste des labels utilisés par chaque contributeur : cela permettra au transcripteur de générer des vues avec le vocabulaire de leur destinataire.

Il existe d'autres relations. Par exemple [10] définit : l'absence de relation, l'équivalence, l'inclusion et le recouvrement partiel. Nous ne les traitons pas ici.

Ces actions de fusion améliorent la qualité de la représentation formelle. Les informations de provenance conservent les actions effectuées au titre de fusion et leur justification.

4.3.5 Exploiter des spécifications d'observations

Pour assurer la complétude du modèle, le transcripteur doit

— formaliser les spécifications d'observations en exploitant voire enrichissant le méta-modèle,
— les évaluer sur le modèle,
— interagir avec les contributeurs pour compléter les contributions si des données manquent pour évaluer des spécifications d'observations formalisées,
— fournir le résultat de l'évaluation, sous une forme pertinente pour l'observateur. En plus de la spécification d'observation, il s'appuiera sur un profil de l'observateur construit à partir de ses préférences personnelles, celles de l'organisation à laquelle il appartient et celles de son domaine métier [14].

4.3.6 Évaluer la représentation formelle

Pour estimer la qualité de la représentation formelle, le transcripteur peut évaluer les propriétés définies section 3.

Ces évaluations lui permettront de faire remonter les points faibles de la représentation formelle obtenue à partir des contributions considérées. Grâce aux informations de provenance, il sera parfois possible de cibler les contributeurs à solliciter pour améliorer certains points (en leur produisant éventuellement une vue centrée sur ces derniers). Nous estimons que ce mécanisme peut avoir un effet positif très important sur le processus de modélisation collective, en améliorant l'efficience des interactions.

5 Etat de l'art

Dans [2], certaines situations de modélisation sont identifiées comme pouvant se poser lors de la construction agile d'un modèle de système, à partir de contributions diverses. Bien que le positionnement de ces situations y soit fait à un niveau un peu plus abstrait, la démarche est très proche.

La prise en compte de contributions hétérogènes, du point de vue de leur représentation, leur contenu et les concepts utilisés, correspond à l'approche "fédération de modèles" dans [7]. Les auteurs proposent de ne pas construire de méta-modèle commun et de reposer plutôt sur des synchronisations modèle à modèle. Le but est de ne pas imposer à chaque contributeur de reformuler son apport dans un langage qui serait commun à tous, mais moins adapté à l'information qu'il cherche à apporter ou à son domaine métier. Si notre approche tente au contraire de construire un méta-modèle commun, les échanges vers contributeurs et observateurs personnalisent la représentation, afin qu'elle reste pertinente pour le destinataire de l'information.

[5] identifient l'insuffisance du retour sur investissement humain des approches basées modèles comme le principal frein à leur adoption dans le domaine de l'ingénierie logicielle. Nous pensons qu'une observation assez proche peut être faite sur l'ingénierie système, [9] considère qu'il manque principalement des preuves que ce retour est à la hauteur de l'effort consenti. Notre approche va dans le sens de ce que [5] nomment "cognification" de l'ingénierie dirigée par les modèles. Plutôt que de ne reposer que sur le jugement et l'action humaine, nous proposons d'outiller la construction de la représentation formelle pour alléger et guider l'effort humain. Pour cela nous spécifions les actions à mener, et nous identifions certains éléments de décision pour effectuer ou supporter les corrections et ajouts.

Les caractéristiques permettant d'évaluer la qualité de l'information, certaines propriétés du modèle, et certaines idées générales (telle la nécessité d'adapter l'information présentée à un interlocuteur donné) proposées dans cet article, s'inspirent d'éléments utilisés dans le cadre plus général d'évaluation de l'information lors de collaborations ([14]).

L'idée de pouvoir justifier la présence d'un élément d'information par l'historique des opérations ayant contribué à sa genèse n'est pas nouvelle : mais elle donne de la valeur au modèle obtenu par transcription, et aide à convaincre les observateurs du bien-fondé du modèle, et les contributeurs de la prise en compte effective de leurs contributions. L'ontologie PROV-O [6] utilise les concepts d' acteur, d'opération, de donnée, et des relations pour décrire la provenance d'une donnée comme un graphe représentant sa genèse. Par exemple, [4] transforment et résument ensuite de tels graphes pour permettre la réutilisation de données produites en exécution. Nous nous rapprocherons de ces travaux en nous limitant aux besoins de notre projet.

Le contenu de chaque contribution représente un point de vue instantané sur le système à modéliser. Ce contenu est susceptible d'être révisé au vu des contributions venant le compléter ; il peut aussi faire l'objet de plusieurs doutes (sur la terminologie employée, sur des ambiguïtés, etc.) pour le transcripteur qui vise un modèle le moins ambigu possible. Le travail du transcripteur est donc entaché d'incertitude : l'outil destiné à l'aider doit en tenir compte pour pouvoir ajuster au mieux la part d'autonomie de décision du

6. https://www.w3.org/TR/prov-o/

transcripteur et celle des fonctions automatiques. Pour cela, nous pourrons utiliser une technique de représentation des connaissances et de raisonnement avec incertitude ([1]).

6 Conclusion et perspectives

Nous constatons que la complexité des systèmes actuels ne permet plus à un humain d'en réaliser une modélisation exploitable pour en assurer la maîtrise et l'évolution. Nous proposons un processus de transcription d'un jeu de contributions partielles et hétérogènes décrivant un système complexe de manière imparfaite et incomplète. Ce processus, qu'il est possible d'outiller, vise à générer une représentation formelle consensuelle entre les contributeurs. Celle-ci doit répondre aux besoins de ses parties prenantes, dont une gestion sûre de la confidentialité et le calcul d'indicateurs accompagnant leurs décisions sur le système.
Pour outiller ces travaux, nous utilisons un prototype (WEIRD, introduit dans [3]) qui devra être étendu. Ce travail sera enrichi par la prise en compte de divers formats de contribution, le traitement d'autres formes d'ambiguïté sémantique et l'utilisation d'une théorie de l'incertitude.

Remerciements

Ce travail a bénéficié à ses débuts de la participation de Laurence Cholvy (ONERA). Il est en partie financé par l'Agence de l'Innovation de Défense (AID) du Ministère des Armées français (convention de recherche CONCORDE N° 2019 65 0090004707501).

Références

[1] Salem Benferhat, Thierry Denoeux, Didier Dubois, and Henri Prade. Représentations de l'incertitude en intelligence artificielle. In Pierre Marquis, Odile Papini, and Henri Prade, editors, *Panorama de l'Intelligence Artificielle*, volume 1 : Représentation des connaissances et formalisation des raisonnements - Chapitre 3, pages 65–121. Cépaduès Editions, 2014.

[2] Antoine Beugnard, Fabien Dagnat, Sylvain Guerin, and Christophe Guychard. Des situations de modélisation pour évaluer les outils de modélisation. In *INFORSID 2014 : 32ème congrès de l'INFormatique des ORganisations et Systèmes d'Information et de Décision*, pages 181–196, Lyon, France, May 2014.

[3] Pierre Bieber, Frédéric Boniol, Guy Durrieu, Olivier Poitou, Thomas Polacsek, Virginie Wiels, and Ghilaine Martinez. MIMOSA : Towards a model driven certification process. In *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, TOULOUSE, France, January 2016.

[4] Alban Gaignard, Hala Skaf-Molli, and Khalid Belhajjame. Découvrabilité et réutilisation de données produites par des workflows : un cas d'usage en génomique. In Maxime Lefrançois, editor, *Journées Francophones d'Ingénierie des Connaissances (IC) PlateForme Intelligence Artificielle (PFIA'21)*, pages 73–80, Bordeaux, France, June 2021.

[5] Sébastien Gérard, Jordi Cabot, Robert Clarisó, Marco Brambilla, and Sébastien Gerard. Cognifying Model-Driven Software Engineering. In *Software Technologies : Applications and Foundations*, pages 154–160. Springer, January 2018.

[6] Fahad R. Golra, Antoine Beugnard, Fabien Dagnat, Sylvain Guerin, and Christophe Guychard. Using free modeling as an agile method for developing domain specific modeling languages. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, MODELS '16, page 24–34, New York, NY, USA, 2016. Association for Computing Machinery.

[7] Fahad Rafique Golra, Antoine Beugnard, Fabien Dagnat, Sylvain Guerin, and Christophe Guychard. Addressing Modularity for Heterogeneous Multi-model Systems using Model Federation. In *MODULARITY 2016 : 15th International Conference on Modularity*, pages 206 – 211, Malaga, Spain, March 2016.

[8] Esther Guerra and Juan de Lara. On the quest for flexible modelling. In *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, MODELS '18, page 23–33, New York, NY, USA, 2018. Association for Computing Machinery.

[9] Kaitlin Henderson and Alejandro Salado. Value and benefits of model-based systems engineering (mbse) : Evidence from the literature. *Systems Engineering*, 24(1) :51–66, January 2021.

[10] Ana Meštrović. Semantic matching using concept lattice. *Concept Discovery in Unstructured Data, CDUD*, 871 :49–58, 01 2012.

[11] Olivier Poitou, Pierre Bieber, Joël Ferreira, and Ludovic Simon. Formal architecture modeling for documenting and assessing Aeronautics Maintenance : A case study. In *ERTS 2018*, 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), Toulouse, France, January 2018.

[12] A. L. Ramos, J. V. Ferreira, and J. Barceló. Model-Based Systems Engineering : An Emerging Approach for Modern Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1) :101–111, January 2012. Conference Name : IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews).

[13] Roxana Saavedra, Luciana C. Ballejos, and M. Ale. Requirements quality evaluation : State of the art and research challenges. In *XIV Simposio Argentino de Ingeniería de Software (ASSE) - JAIIO 42*, pages 240–257, 2013.

[14] Claire Saurel, Olivier Poitou, and Laurence Cholvy. Assessing the usefulness of information in the context of coalition operations. In Éloi Bossé and Galina L. Rogova, editors, *Information Quality in Information Fusion and Decision Making*, pages 135–154. Springer International Publishing, Cham, 2019.