



HAL
open science

Simuler des erreurs de communication au sein d'une équipe d'agents virtuels en situation de crise

Lauriane Huguet, Nicolas Sabouret, Domitile Lourdeaux

► To cite this version:

Lauriane Huguet, Nicolas Sabouret, Domitile Lourdeaux. Simuler des erreurs de communication au sein d'une équipe d'agents virtuels en situation de crise. Rencontres des jeunes chercheurs en Intelligence Artificielle (RFIA 2016), Jun 2016, Clermont-Ferrand, France. hal-01342668

HAL Id: hal-01342668

<https://hal.science/hal-01342668v1>

Submitted on 6 Jul 2016

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.

Simuler des erreurs de communication au sein d'une équipe d'agents virtuels en situation de crise

L. Huguet^{1,2}

N.Sabouret²

D.Lourdeaux¹

¹ Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, 60203 Compiègne Cedex

² Limsi, CNRS, Univ. Paris Sud, Université Paris Saclay, F – 91405, Orsay

lauriane.huguet@hds.utc.fr

Résumé

Dans le contexte de la formation de leaders d'équipe médicale, nous présentons un modèle de communication qui permet de reproduire des erreurs au sein d'une équipe d'agents virtuels. Ce modèle est élaboré à partir des travaux du domaine des systèmes multiagents et de la théorie des actes de langage, et d'un corpus de dialogues collectés sur le terrain lors de séances de formation. Notre modèle permet de simuler quatre types d'erreurs de communication (malentendu, mauvaise interprétation, incompréhension et non-réponse) introduites à différentes étapes du cycle de communication.

Mots clefs

Intelligence Artificielle, Environnement virtuel d'entraînement, Système multiagent, Communication, Stress, Travail d'équipe

Abstract

In the context of training medical team leaders, we present a multiagent communication model that can produce errors in a team of agents. This model is built from existing work from the literature in multiagent systems and speech act theory, but also from a corpus of dialogues collected during actual field training for medical teams. Our model supports four types of communication errors (misunderstanding, misinterpretation, non-understanding and absence of answer) that appear at different stages of the communication process.

Key Words

Artificial Intelligence, Virtual Environment for training, Multiagent Systems, Communication, Stress, Teamwork

1 Introduction

Le métier de secouriste est en train d'évoluer : les interventions, aussi bien en milieu urbain avec des populations civiles que dans le contexte militaire, se déroulent

souvent en terrain non sécurisé ou avec un afflux de victimes trop important par rapport aux ressources disponibles. Ces situations sont donc sources de stress [Lazarus and Folkman, 1984] et donc plus propices aux erreurs, comme l'ont montré [Driskell and Salas, 2013]. Pour gérer ce stress, il est nécessaire d'être préparé pour faire face à différentes situations « exceptionnelles », de manière à ce qu'un certain nombre de situations imprévues [Hollnagel, 2004] ne soient pas des situations impensées mais des situations possibles, du même ordre que celles auxquelles on a déjà été confronté en théorie ou en pratique.

C'est pourquoi, dans le contexte de la formation des équipes de secouristes, les organismes comme la Brigade de Sapeurs Pompiers de Paris ou l'École du Val de Grâce réalisent des simulations « grandeur nature ». Ces formations requièrent l'intervention d'un grand nombre d'opérateurs et mobilisent beaucoup de matériel. Elles ne peuvent donc pas être multipliées à l'infini. Pour palier ce problème, le projet VICTEAMS¹ a pour objectif de proposer un environnement virtuel d'entraînement dans lequel l'utilisateur interagit avec une équipe d'agents virtuels secouristes *autonomes*, implémentée à l'aide d'un système multiagent. Ce type d'approche en environnement virtuel permet non seulement d'augmenter la fréquence des formations mais aussi de confronter les apprenants à des situations que l'on ne peut pas reproduire en simulation réelle. Dans le projet VICTEAMS, nous ne nous intéressons pas à la formation aux gestes techniques : nous travaillons sur la formation des compétences non-techniques, c'est-à-dire les compétences cognitives, sociales et les ressources personnelles qui complètent les compétences techniques d'une personne et contribuent à la sécurité et à une meilleure efficacité dans l'exécution de la tâche [Flin et al., 2008]. En effet, des études [Flin et al., 2009] ont montré que plus de 70% des erreurs médicales sont dues à des erreurs liées aux

1. Virtual Characters for team Training : Emotional, Adaptive, Motivated and Social. <https://victteams.hds.utc.fr/>



FIGURE 1 – Premier aperçu de l’environnement virtuel du projet VICTEAMS, développé par Reviatech.

compétences non-techniques et plus particulièrement à la communication au sein de l’équipe médicale.

Dans ce contexte de formation, nous proposons un modèle de communication capable de reproduire dans le simulateur des erreurs comme il s’en produit au sein d’une équipe de véritables secouristes. Pour cela, nous allons déterminer les différentes erreurs à reproduire et à quelles étapes les introduire. Par exemple, nous souhaitons que nos agents puissent ne pas communiquer une information ou ne pas la transmettre au moment opportun ou encore mal interpréter le message qu’ils reçoivent.

La prochaine section présente une brève revue de la littérature qui met en évidence les modèles d’erreurs possibles. Nous présenterons ensuite notre modèle de communication intégrant des erreurs de communication (section 3) puis une évaluation préliminaire fondée sur un questionnaire (section 4).

2 Travaux connexes

Pour notre étude, nous avons été amenés à étudier plusieurs domaines de recherche autour de la communication, les théories de la communication en traitement de l’information et en sciences du langage d’une part et le traitement informatique des communications d’autre part.

2.1 Les théories de la communication

L’un des premiers modèles formels de la communication vient de Shannon [Shannon and Weaver, 1949] qui propose un modèle mathématique de la communication en traitement de l’information. Pour lui un message est encodé puis envoyé par un certain canal à un destinataire qui le décode. Pour garantir l’intégrité du message il est donc nécessaire d’avoir un canal de transmission non bruité et un codage de l’information partagé entre émetteur et récepteur. Ces éléments sont intéressants pour comprendre où l’on peut introduire des erreurs de communication : au niveau de l’émetteur (encodage), au niveau de l’environnement qui véhicule le message et au niveau du récepteur qui décode le message.

À l’issue de cette dernière étape, c’est-à-dire à la réception du message, on retrouve les trois types d’erreur définis par Mac Roy [McRoy, 1998].

1. Incompréhension (non-understanding) : le message reçu ne fait pas sens et le sens ne peut pas être réparé.
2. Mauvaise interprétation (misinterpretation) : le message a du sens mais est en désaccord avec les connaissances du récepteur.
3. Malentendu (misunderstanding) : le récepteur pense avoir compris le message qu’il reçoit mais ce qu’il comprend est différent de ce que voulait dire le locuteur.

À partir de l’étude d’un corpus vidéo collecté avec des médecins militaires, on a pu observer ces différents types d’erreurs. Dans le premier cas, l’erreur sur le message n’est pas sémantique, le message est trop dégradé pour être accepté (par exemple, un hélicoptère passe au moment où l’opérateur annonce le temps restant avant l’évacuation). Le récepteur est donc obligé de demander de répéter l’information.

Dans le deuxième cas, le message pose un problème sémantique au récepteur parce qu’il entre en conflit avec ses connaissances. Par exemple, lorsque le médecin coordinateur demande à l’infirmier comment va le bras de la victime et que l’infirmier ne comprend pas la question. Il n’avait pas eu l’information d’une douleur dans le bras du patient et s’était focalisé sur les autres traumatismes portés à sa connaissance.

Enfin dans le troisième cas, l’erreur n’est généralement pas visible immédiatement. Le récepteur pense avoir compris le message mais sa compréhension est différente de celle de l’émetteur. Il est même parfois possible de ne pas s’en rendre compte. Par exemple, dans l’une de nos vidéos, le médecin dit « on lui fait une coniotomie ? » à son infirmier qui le comprend non pas comme une question mais comme un ordre et répond simplement « oui ». Cette erreur n’a été détectée que lors du débriefing.

Ce dernier type d’erreur est intéressant, car il montre qu’un message peut être compris de différentes manières par les interlocuteurs. Le modèle carré de Schulz Von Thun [Schulz von Thun, 1981] donne un cadre intéressant pour conceptualiser ce phénomène. Il part du principe qu’un message porte toujours quatre composantes : un fait, une information sur l’émetteur, une information sur la relation entre l’émetteur et le récepteur et un appel à agir. Ces quatre composantes seraient toujours présentes mais de manière plus ou moins accentuée et plus ou moins intentionnelle. Ce modèle donne une explication pour les erreurs de type malentendu : d’après [St Pierre et al., 2008], les erreurs se produisent lorsque le récepteur et l’émetteur ne donnent pas la même importance aux différentes composantes du message. L’idée qu’un message comporte plusieurs interprétations possibles est une idée fondamentale pour notre modèle. Cependant, il nous a paru difficile de trouver exactement quatre significations correspondant aux quatre composantes définies par le modèle carré. C’est pourquoi dans notre modèle, nous autorisons un nombre quelconque de significations.

2.2 Les modèles informatiques

L'étude des théories de la communication nous a permis de mettre en valeur un certain nombre de possibilités pour introduire des erreurs dans les communications. Afin d'opérationnaliser ces éléments, il faut maintenant s'intéresser à ce qui est fait en informatique, et en particulier dans les systèmes multiagents.

Génération de messages. Les travaux en informatique reposent sur la théorie des actes de langage développée par Austin [Austin, 1975] et Searle [Searle, 1969]. Dans ce modèle, on distingue le locutoire (la forme du message en langage naturel), l'illocutoire (le sens du message envoyé) et le perlocutoire (l'effet du message sur le receveur). Pour décrire les phénomènes illocutoire et perlocutoire, un message est formé à partir d'un performatif qui indique le but de la communication, par exemple « Informer » ou « Ordonner » et un contenu qui permet de définir le sujet de la communication.

Dans les systèmes multiagents, c'est la norme FIPA² qui est la plus répandue. Elle propose, pour qu'une communication soit réussie, que l'illocutoire et le perlocutoire ne soient pas différenciés. C'est-à-dire qu'il n'y a pas d'erreur de communication possible à ce niveau, le message envoyé est bien le message reçu. De plus, la norme FIPA définit des protocoles de communication qui contraignent l'agencement des messages pouvant être reçus et la manière de les interpréter. Le but est donc de minimiser les erreurs.

C'est le cas aussi des systèmes de dialogue qui s'intéressent à générer un énoncé pertinent dans le contexte de la conversation que ce soit par planification réactive [Rich and Sidner, 1998] ou par initiative mixte [Ferguson and Allen, 1998]. Obtenir une bonne communication dans le cadre de la communication humain-agents dans un environnement virtuel est déjà une difficulté en soi car il faut supporter tous les types d'interactions humain-agent ou agent-agent et que tous les messages soient interprétables sans perdre en liberté et en expressivité [Barange, 2015].

Optimiser la communication dans le but d'une coopération efficace a donc été le principal point d'intérêt de la communauté. Notre travail cherche au contraire à introduire des erreurs dans la communication. Dans le but de sensibiliser l'utilisateur aux conséquences des erreurs de communication, ces erreurs doivent être générées de manière opportune et explicite.

Traitement des erreurs de communication. Certains travaux en traitement automatique des langues [Shin et al., 2002] se sont intéressés à l'étude d'erreurs de communication, et plus particulièrement aux erreurs syntaxiques. Les modèles obtenus sont essentiellement statistiques (ils portent sur des taux d'erreur au niveau lexical ou syntaxique), et ne s'intéressent généralement pas au contexte de l'interaction.

Or la prise en compte du contexte est importante dans notre modèle pour comprendre les erreurs au niveau sémantique. Par exemple, une erreur classique observée sur le terrain est d'échanger les termes « morphine » et « adrénaline ». Ce sont des termes sémantiquement proches car ce sont tous les deux des termes médicaux, désignant des produits pouvant être injectés et fréquemment utilisés par les secouristes. Cependant leurs utilisations sont contraires et la confusion des deux a de lourdes conséquences sur les victimes. La confusion qui semble lexicale est en fait une confusion au niveau sémantique liée au contexte médical. C'est pourquoi nous proposons un modèle sémantique capable de reproduire ce type d'erreurs de communication.

D'autre part, dans les systèmes de dialogue, il existe des travaux proposant des algorithmes de gestion des communications pour réparer les erreurs de type « mauvaise interprétation » [Ardissono et al., 1997]. Cependant, ils ne proposent pas, à notre connaissance, de modèle pour produire ces erreurs à réparer.

Dans la suite, nous allons exposer une première version de notre modèle de communication qui permet d'introduire des erreurs de communication à la génération, lors du transport et à la réception du message.

3 Proposition

3.1 Principe général

Nous nous plaçons dans un environnement virtuel peuplé d'agents autonomes interagissant entre eux et avec l'avatar de l'utilisateur à l'aide de messages structurés (voir section 3.2). Chaque agent est caractérisé par un nom, un rôle (leader, opérateur de transmission...) et un statut (médecin, infirmier...) permettant de l'interpeller. Ces différents éléments permettent de garder une certaine ambiguïté quant au destinataire réel du message.

L'agent dispose aussi de variables d'état. Parmi ces variables, nous nous intéresserons plus particulièrement à sa position dans l'environnement virtuel et à la direction dans laquelle l'agent regarde, dont nous verrons l'utilisation dans la section 3.3. Nous utilisons aussi une variable de « stress » qui prend sa valeur dans [0, 1] et qui permet de contrôler le taux d'erreur (voir section 3.5).

$$\text{Agent} = \langle \text{Nom, rôle, statut, position, orientation, seuil_stress, stress} \rangle$$

Comme cela se fait classiquement en système multiagent et comme nous l'illustrons sur la figure 2, chaque agent qui souhaite communiquer avec les autres agents ou l'avatar construit un message (cf (1.), figure 2), le dépose dans l'environnement qui le transmet aux agents (cf (2.), figure 2). Le transport des messages se fait au sein d'un environnement actif [Platon et al., 2005] permettant l'altération des messages et la distribution à d'autres agents que le destinataire initial. Enfin, chaque agent choisit d'ignorer ou de traiter les messages qu'il reçoit (cf (3.), figure 2).

2. <http://www.fipa.org/repository/aclspecs.html>

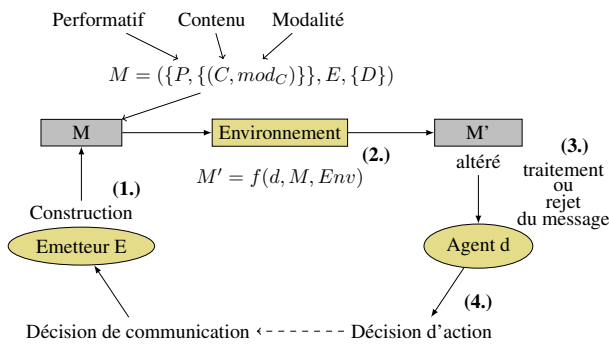


FIGURE 2 – Schéma récapitulatif du modèle

3.2 Structure des messages

Acte illocutoire. La structure d'un message est fondée sur la théorie des actes de langage [Searle, 1969] : elle comprend l'émetteur, les destinataires et les informations de l'acte illocutoire sous la forme (*performatif, contenu*). Les destinataires sont définis par leur nom, leur rôle ou leur statut. Il est aussi possible de ne pas préciser de destinataire dans un message (il est alors lancé à la volée). Les performatifs, qui caractérisent le traitement associé au message, ne sont pas contraints : ils peuvent par exemple être choisis parmi ceux définis par les normes FIPA ou DIT++ [Bunt, 2009], mais il est possible de définir d'autres performatifs propres à l'application considérée. Pour le moment, nous utilisons les quatre performatifs décrits sur la figure 3. Les contenus sont formalisés des requêtes SPARQL [Prud'Hommeaux et al., 2008] ou des ensembles de triplets RDF [Klyne and Carroll, 2006]. À titre d'exemple, informer de l'état d'un patient se fait à l'aide du performatif INFORMER et d'un ensemble de triplets RDF :

```
INFORMER { (patient-3 has-etat _:e),
            (_:e rdf:type Etat),
            (_:e cardio ...),
            ... }
```

Demander l'heure prévue d'évacuation se fait à l'aide de l'acte illocutoire :

```
DEMANDER ?x { (?x rdf:type Date),
              (e1 rdf:type Evacuation),
              (e1 has-date ?x) }
```

Actes perlocutoires. Pour simuler les interprétations multiples portées par un message, l'acte illocutoire est complété par d'autres couples (performatif, contenu), symbolisant les actes perlocutoires possibles. Par exemple, pour le locutoire « on intube ? » qui porte une ambiguïté de performatif, nous aurons (au moins) deux actes de langage : l'ordre avec le performatif ORDONNER et la question avec le performatif QUESTIONNER. De même, pour une erreur

Performatif	Utilisation
Informer	Transmettre une information
Questionner	S'enquérir d'une information
Ordonner	Donner un ordre
Interpeller	Attirer l'attention d'un agent

FIGURE 3 – Performatifs utilisés dans VICTEAMS.

d'interprétation de l'objet (« mets-le ici » où « le » peut désigner un objet différent pour l'émetteur et le destinataire), nous aurons plusieurs actes de langage, chacun avec un contenu différent.

Nous ne présentons pas ici de mécanisme automatique pour générer ces actes perlocutoires, car c'est un problème difficile que nous n'avons pas encore abordé. Dans la version actuelle du système, ces actes sont définis manuellement et à l'avance pour les différents *types* de mauvaise compréhension se produisant en situation réelle. Ces situations ont été extraites d'un corpus de communications recueillies lors de formations à l'École du Val de Grâce. Nous avons pu identifier deux catégories d'erreurs. La première, souvent liée au contexte, porte sur le contenu sémantique, comme dans l'exemple de confusion entre adrénaline et morphine. Celui-ci se produit si l'on parle d'adrénaline alors qu'il y a une seringue de morphine déjà prête à être utilisée par exemple. C'est donc la proximité physique de l'objet ou le fait qu'on vienne de s'en servir qui crée la confusion.

L'autre catégorie d'erreur observée porte sur les performatifs (prendre une question pour une affirmation ou l'inverse). Nous avons observé que ce type d'erreur est fortement liée au locutoire (par exemple, poser une question avec une tournure affirmative : « on y va ? »). Nous avons donc fait le choix de proposer systématiquement le même contenu avec les deux autres performatifs de la liste (INFORMER, QUESTIONNER, ORDONNER), le cas du performatif Interpeller étant un peu à part (il s'agit juste d'attirer l'attention d'un agent).

Modalités. En complément des éléments classiques présentés ci-dessus, chaque partie du message est associée à une modalité. Par exemple, considérons un secouriste qui dit à son collègue « Passe-moi le ... [geste désignant la trousse de secours] ». Le message est complet mais il est composé d'une première partie verbale et d'une seconde partie visuelle. Le contenu est donc découpé en un ensemble de couples (c, m) où c est un ensemble de triplets RDF, sous-ensemble du contenu global du message, et m la modalité qui y est associée.

Ainsi, dans notre modèle de communication, un message M est constitué d'un ensemble de significations S , d'un émetteur E , et d'un ensemble de destinataires D :

$$M = (\{S\}, E, \{D\}) .$$

Les destinataires D sont désignés soit par leur noms, par

exemple l'agent Toto :

```
D = { (_:a rdf:agent Agent),
      (_:a has-name Toto) } ;
```

soit par leur fonction, par exemple tous les infirmiers :

```
D = { (_:a rdf:agent Agent),
      (_:a has-status Infirmier) } .
```

Une signification S correspond à un acte perlocutoire. Elle est constituée d'un performatif P et d'un ensemble de couples (contenu, modalité) :

$$S = (P, \{(contenu, modalité)\}) .$$

Ce message ainsi formé est alors envoyé dans l'environnement.

3.3 Transport et altération des messages

L'environnement a une double fonction dans notre modèle : il détermine quels agents vont recevoir chaque message et avec quel degré de dégradation. La dégradation du message peut varier suivant le canal de transmission ou la modalité utilisée et peut être différente entre les différents récepteurs du message.

Si les contenus présents dans un message passent par deux canaux de transmission différents, on aura la possibilité, dans le cas où un seul canal de transmission est rompu, de ne supprimer du message que les contenus passant par ce canal. Par exemple, si le secouriste dit à son collègue « Passe-moi le ... [geste désignant la trousse de secours] » alors que le collègue est en train de regarder le patient, il ne verra peut-être pas la partie visuelle du message et recevra donc « Passe-moi le » qui est un message incomplet.

Les personnes susceptibles de percevoir un message sont celles vérifiant les conditions de transmission pour les modalités envisagées. Pour chaque agent d existant, l'environnement détermine si cet agent peut recevoir le message M envoyé par l'émetteur en fonction des modalités utilisées et de l'état des canaux de transmission pour chaque modalité $\{etat_{mod}\}$. Il construit alors un message M' altéré à partir de M :

$$M' = \text{percu}(M, d, \{etat_{mod}\})$$

Par exemple si la modalité est visuelle, le message sera transmis aux personnes positionnées au bon endroit et regardant dans la bonne direction ; si la modalité est vocale, le message sera transmis aux agents suffisamment proches et qui ne sont pas affectés par des bruits parasites.

Les agents qui reçoivent le message ne sont donc pas forcément dans la liste des destinataires du message, car l'environnement envoie un message, plus ou moins altéré, à tous les agents qui satisfont aux conditions de transmission d'au moins une des modalités utilisées. Cela permet de reproduire le phénomène d'écoute flottante (*overhearing*) [Balbo et al., 2004, Platon et al., 2005]. Ce phénomène permet aux agents de répondre à une question

qui ne leur est pas destinée mais dont ils pensent connaître la réponse ou d'inférer des choses sur les conversations entre leurs collègues car ils en ont ainsi connaissance, ce qui ne serait pas le cas si le message n'était transmis qu'aux destinataires intentionnels.

3.4 Reception des messages

Chaque agent à portée du message reçoit alors son message M' plus ou moins altéré et décide s'il l'ignore ou s'il le traite. S'il décide de le traiter, il lui faut ensuite choisir un acte perlocutoire, c'est-à-dire l'une des significations contenues dans le message M' , et agir en fonction, comme indiqué au point (4.) de la figure 2.

Le tableau de la figure 4 résume ainsi les différentes étapes du modèle avec les types d'erreurs générées à chaque étape et les fonctions proposées pour ces étapes. Dans la suite nous détaillons chacune de ces fonctions.

étape	type d'erreur	commentaire
(1.)	malentendu	génération des significations alternatives
(2.)	incompréhension	Algorithme 1
(3.)	non réponse	Algorithme 2
(4.)	mauvaise interprétation	Algorithme 2

FIGURE 4 – Récapitulatif des erreurs traitées par le modèle

3.5 Implémentation

Pour mettre en œuvre les mécanismes présentés ci-dessus, nous avons défini trois fonctions essentielles, comme illustré sur la figure 5 :

- Pour la distribution et l'altération du message par l'environnement, nous avons utilisé une fonction *perception* (cf : Algorithme 1) qui prend en argument un message et un agent et détermine si l'agent reçoit ou non le message, et avec quelle(s) altération(s).
- La fonction *niveau.degradation* permet à un agent de déterminer le taux de dégradation d'un message reçu.
- La fonction *acceptation* permet à cet agent, en fonction du niveau de dégradation et de son propre état interne (en particulier le niveau de stress), de décider s'il doit traiter le message, demander qu'on le lui ré-envoie ou simplement l'ignorer.

Ces trois fonctions sont détaillées dans les paragraphes suivants.

La fonction de perception du message et d'altération par l'environnement. Tous les messages, indépendamment des destinataires explicites, sont potentiellement transmis à tous les agents, conformément au principe d'écoute flottante. Ainsi, pour message M et pour chaque agent a , l'environnement construit un message altéré $M'(a)$ qui définit le message reçu par l'agent a .

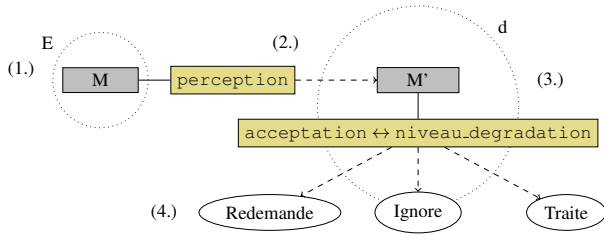


FIGURE 5 – Étapes du traitement d'un message

Deux cas particuliers sont à considérer. Si le message n'a pu être transmis par aucun canal entre l'émetteur et a alors il n'est pas distribué à l'agent a . L'autre cas est celui où le message a pu être transmis mais avec un contenu vide. Par exemple, imaginons deux secouristes qui se voient mais utilisent une radio pour communiquer. Si l'un des secouristes voit l'autre parler dans la radio et que celle-ci est en panne, il recevra un message mais de contenu vide.

Algorithme 1 `perception(msg, agent)`

```

pour signification  $\in$  msg.significations faire
  pour (contenu, modalite)  $\in$  signification faire
    si canal_rompu(emetteur, agent, modalite) alors
      Remplacer contenu par X
    fin si
  fin pour
fin pour

```

La fonction `canal_rompu` prend en entrée deux agents et une modalité et renvoie `True` si le canal véhiculant la modalité entre les deux agents est rompu.

Lorsqu'une partie du message ne peut être perçue par l'agent qui reçoit le message, nous avons fait le choix de remplacer les contenus correspondants par un mot clé X. Cela nous permet de repérer les parties endommagées du message pour le calcul du taux de dégradation présenté ci-après.

Le calcul de la dégradation. Dans la version actuelle du modèle, le calcul de la dégradation du message est un simple calcul du taux de contenu manquant. Il n'y a donc pas d'information sémantique sur la dégradation du message. Comme expliqué dans la section précédente, le calcul est rendu possible par le remplacement des contenus non perceptibles par le mot-clé X.

$$\text{niveau_degradation}(msg) = \frac{\text{nombre de X}}{\text{nombre de triplets}}$$

Ce calcul de dégradation n'est bien évidemment pas optimal car il est possible que l'altération se produise sur des informations superflues et que la partie perçue soit suffisante pour traiter le message. Cependant, nous voulions que dans une situation non stressée nos agents adoptent un comportement prudent. À terme, nous souhaitons définir

un mécanisme plus complet pour calculer le taux de dégradation qui prend en compte le contexte.

La fonction de décision : quand faire répéter. En s'appuyant sur une revue de littérature en sciences humaines, il apparaît que plus l'on est stressé plus l'attention se focalise [Easterbrook, 1959] (on parle de *tunnelisation*). Notre fonction d'acceptation des messages se décompose en deux parties. Si le niveau de stress de l'agent est supérieur à un seuil propre à l'agent alors il y a une probabilité α élevée qu'il ignore le message reçu (le paramètre α devant être calibré). Sinon, nous nous appuyons sur l'affirmation de Driskell et Salas [Driskell and Salas, 2013] qui expliquent que plus les agents sont stressés, plus ils sont enclins à faire des erreurs (c'est-à-dire le contraire de ce qu'ils devraient faire en situation optimale). Ainsi, lorsque l'agent est stressé (mais sous le seuil de tunnelisation), il devrait faire plus répéter les bons messages que les mauvais. Lorsque l'agent a un stress nul, la probabilité de faire répéter un message est proportionnelle à son taux de dégradation.

Ce point nous amène donc à choisir une fonction dite en « selle de cheval » de la figure 6. Grâce à une interpolation et en tenant compte des contraintes énoncées précédemment, nous avons obtenu l'équation :

$$\text{redemande}(s, d) = 0.625s^2 - 1.55sd + 0.125s + 0.95d + 0.04,$$

où s est le stress et d la dégradation.

La fonction de décision `acceptation` (*cf* : Algorithme 2), indique, en fonction du message reçu et de son état de stress, si l'agent ignore le message qu'il reçoit, s'il demande au locuteur de répéter ou s'il décide d'agir en fonction de ce qu'il peut inférer.

Algorithme 2 `acceptation(agent, msg)`

```

si stress > seuil ET rand() <  $\alpha$  alors
  retourner "ignore"
sinon
  msg' = percu(msg, agent)
  degradation = niveau_degradation(msg')
  r = redemande(stress, degradation)
  p' = rand()
  si p' < r alors
    retourner "redemande"
  sinon
    retourner "infère"
  fin si
fin si

```

4 Évaluation préliminaire

La génération de significations alternatives décrites précédemment devrait permettre de générer un grand nombre de possibilités. Il faut ensuite en sélectionner une ou deux à envoyer à l'agent pour éviter que les messages transmis dans le système ne soient trop lourds. De plus, les différentes solutions proposées ne sont pas de même difficulté technique.

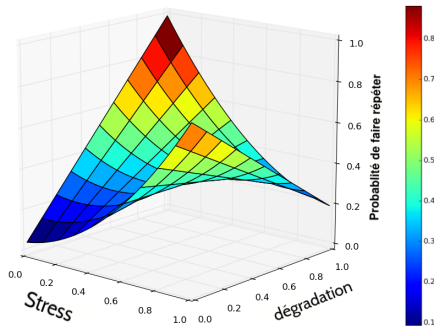


FIGURE 6 – Allure de la courbe de la fonction redemande(stress, dégradation)

Nous avons donc voulu confirmer les erreurs les plus fréquentes et les plus probables dans notre environnement virtuel avec les biais liés à la communication par messages écrits et non via synthèse vocale. Nous avons pour cela fait une étude subjective, à l'aide d'un questionnaire.

Le questionnaire proposait des messages en langage naturel mais écrits sans ponctuation pour ne pas contraindre l'interprétation des participants. Il était alors demandé de noter (de 0 à 4) les différentes significations alternatives proposées pour ce message en essayant de répondre assez rapidement. Les participants avaient aussi la possibilité de proposer une autre signification s'ils avaient compris le message d'une manière non proposée. Les messages étaient écrits et non audios car c'est le type d'interaction qui sera proposé dans un premier temps pour le prototype du projet VICTEAMS dans lequel s'inscrit ce travail.

Ce questionnaire a été rempli par 16 participants (non-experts) bien que ce ne soit pas suffisant pour faire une étude statistique, cela nous a permis de faire ressortir certains points. Premièrement, il y a très peu de consensus sur les réponses apportées par les participants comme on peut le voir sur la figure 7. Cela milite en faveur du fait de maintenir plusieurs significations alternatives.

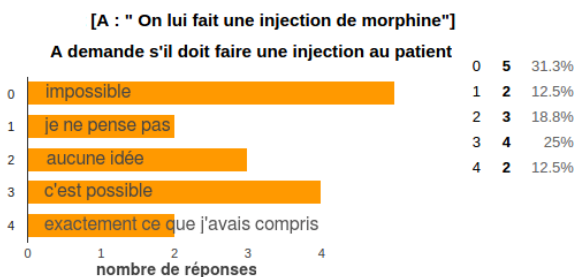


FIGURE 7 – Cas où les participants ne semblent pas d'accord entre eux - malentendu

Ensuite, nous avons pu confirmer que l'erreur sémantique (« adrénaline » à la place de « morphine ») est fréquente (figure 8). C'est pourquoi nous voulons, à terme, générer

ce type de significations alternatives grâce à des mesures de distances sémantiques dans des ontologies de domaine [Mazuel and Sabouret, 2008].

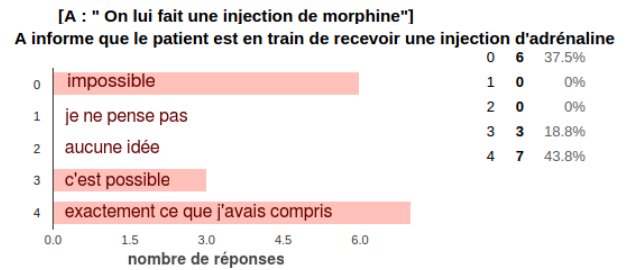


FIGURE 8 – Confirmation d'erreur de type « un mot pour un autre »

Enfin, nous observons sur certaines significations un écart important entre les participants, c'est-à-dire que certaines significations sont totalement rejetées (notées 0) par une majorité de participants et parallèlement jugées totalement plausibles (notées 4) par d'autres (figure 9). On peut donc imaginer que si ces participants se retrouvaient à travailler ensemble, ils auraient plus de mal à se comprendre.

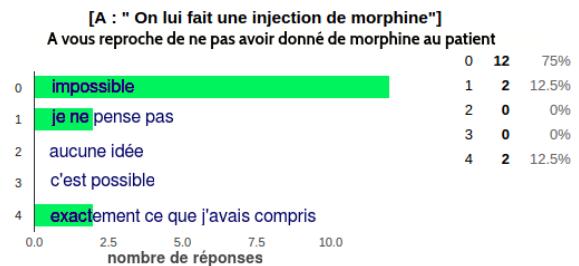


FIGURE 9 – Significations qui semblent réfutées

5 Conclusion et perspectives

Notre modèle permet d'introduire des erreurs de communication à quatre étapes différentes dans le cycle de traitement d'un message :

1. Lors de la création du message, des significations alternatives sont produites pour produire des malentendus. Nous avons proposé des significations alternatives potentielles d'un message en nous appuyant sur des relevés de conversations *in situ*.
2. Lors du transport du message par l'environnement, celui-ci l'altère en fonction des canaux de communication utilisés entre l'émetteur et chacun des récepteurs. L'altération est donc différente pour chaque récepteur du message. Nous pouvons ainsi simuler des erreurs d'incompréhension.
3. À la réception du message par un agent récepteur, nous reproduisons des erreurs de non-réponse :

l'agent décide s'il ignore le message ou non.

4. Au traitement du message, si l'agent récepteur n'a pas ignoré le message qu'il a reçu, il décide s'il agit ou s'il demande à l'émetteur de répéter. Lorsqu'il décide d'agir, il peut alors y avoir des erreurs de mauvaise interprétation.

Ce modèle apporte deux originalités : d'une part l'utilisation de significations alternatives (actes perlocutoires) et d'autre part la prise en compte de la multi-modalité dans l'interaction.

La principale limite du modèle actuel est que les significations alternatives ne se font pas automatiquement. Nous nous sommes appuyés sur un questionnaire pour valider, de manière subjective et avec un nombre restreint de sujets, les erreurs que nous avons introduites dans le modèle. Nous souhaitons à terme automatiser la génération des significations alternatives et prendre en compte la probabilité d'apparition de ces significations, en nous appuyant sur les données des questionnaires.

Remerciements

Les auteurs remercient la Région Picardie et le Fond européen de développement régional (FEDER) 2014/2020 pour le co-financement de ce travail. Celui-ci a été réalisé dans le cadre du projet VICTEAMS (ANR-14-CE24-0027), financé par l'Agence Nationale de la Recherche (ANR) et la Direction Générale de l'Armement (DGA) et labélisé par le Labex MS2T.

Références

- [Ardissono et al., 1997] Ardissono, L., Boella, G., and Damiano, R. (1997). A computational model of misunderstandings in agent communication. In *AI* IA 97 : Advances in Artificial Intelligence*, pages 48–59. Springer.
- [Austin, 1975] Austin, J. L. (1975). *How to do things with words*. Oxford university press.
- [Balbo et al., 2004] Balbo, F., Maudet, N., and Saunier, J. (2004). Interactions opportunistes par l'écoute flottante. *Actes des Journées Francophones sur les Systèmes Multi-Agents*, 2004 :265–270.
- [Barange, 2015] Barange, M. (2015). *Task-Oriented Communicative Capabilities of Agents in Collaborative Virtual Environments for Training*. PhD thesis, Université de Bretagne Occidentale (UBO), Brest.
- [Bunt, 2009] Bunt, H. (2009). The dit++ taxonomy for functional dialogue markup. In *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 13–24.
- [Driskell and Salas, 2013] Driskell, J. E. and Salas, E. (2013). *Stress and human performance*. Psychology Press.
- [Easterbrook, 1959] Easterbrook, J. A. (1959). The effect of emotion on cue utilization and the organization of behavior. *Psychological review*, 66(3) :183.
- [Ferguson and Allen, 1998] Ferguson, G. and Allen, J. F. (1998). Trips : An integrated intelligent problem-solving assistant. In *Proceedings of the Fifteenth National Conference on AI (AAAI-98)*.
- [Flin et al., 2009] Flin, R., Winter, J., and Cakil Sarac, M. R. (2009). Human factors in patient safety : review of topics and tools. *Bulletin of the World Health Organisation*, page 2.
- [Flin et al., 2008] Flin, R. H., O'Connor, P., and Crichton, M. (2008). *Safety at the sharp end : a guide to non-technical skills*. Ashgate Publishing, Ltd.
- [Hollnagel, 2004] Hollnagel, E. (2004). *Barriers and accident prevention*. Ashgate.
- [Klyne and Carroll, 2006] Klyne, G. and Carroll, J. J. (2006). Resource description framework (rdf) : Concepts and abstract syntax.
- [Lazarus and Folkman, 1984] Lazarus, R. S. and Folkman, S. (1984). Stress. *Appraisal, and coping*, 725.
- [Mazuel and Sabouret, 2008] Mazuel, L. and Sabouret, N. (2008). *Semantic relatedness measure using object properties in an ontology*. Springer.
- [McRoy, 1998] McRoy, S. (1998). Preface : Detecting, repairing and preventing human—machine miscommunication. *International Journal of Human-Computer Studies*, 48(5) :547–552.
- [Platon et al., 2005] Platon, E., Sabouret, N., and Honiden, S. (2005). Overhearing and direct interactions : Point of view of an active environment. In *Environments for multi-agent systems II*, pages 121–138. Springer.
- [Prud'Hommeaux et al., 2008] Prud'Hommeaux, E., Seaborne, A., et al. (2008). Sparql query language for rdf. *W3C recommendation*, 15.
- [Rich and Sidner, 1998] Rich, C. and Sidner, C. L. (1998). Collagen : A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3-4) :315–350.
- [Schulz von Thun, 1981] Schulz von Thun, F. (1981). *Miteinander reden : Störungen und klärungen* [talk with each other]. Rowohlt, Reinbek.
- [Searle, 1969] Searle, J. R. (1969). *Speech acts : An essay in the philosophy of language*. Cambridge university press.
- [Shannon and Weaver, 1949] Shannon, C. E. and Weaver, W. (1949). *The mathematical theory of communication*. University of Illinois Press.
- [Shin et al., 2002] Shin, J., Narayanan, S. S., Gerber, L., Kazemzadeh, A., Byrd, D., et al. (2002). Analysis of user behavior under error conditions in spoken dialogs. In *INTERSPEECH*.
- [St Pierre et al., 2008] St Pierre, M., Hofinger, G., and Buerschaper, C. (2008). *Crisis Management in Acute Care Settings*. Springer Berlin Heidelberg.