



DAMIEN MONDOU, ARMELLE PRIGENT, ARNAUD REVEL

Modélisation et supervision d'applications robotiques interactives destinées
au milieu culturel

Volume 3, n° 3-4 (2022), p. 311-344.

http://roia.centre-mersenne.org/item?id=ROIA_2022__3_3-4_311_0

© Association pour la diffusion de la recherche francophone en intelligence artificielle
et les auteurs, 2022, certains droits réservés.



Cet article est diffusé sous la licence

CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.

<http://creativecommons.org/licenses/by/4.0/>



*La Revue Ouverte d'Intelligence Artificielle est membre du
Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org

Modélisation et supervision d'applications robotiques interactives destinées au milieu culturel

Damien Mondou^a, Armelle Prigent^a, Arnaud Revel^a

^a La Rochelle Université – Laboratoire Informatique, Image, Interaction (L3i) 23

Avenue Albert Einstein, 17000 La Rochelle

E-mail : damien.mondou@univ-lr.fr, armelle.prigent@univ-lr.fr, arnaud.revel@univ-lr.fr.

RÉSUMÉ. — De nombreuses approches de conception d'applications robotiques ont été proposées ces dernières années. Si elles s'avèrent pertinentes pour le problème qu'elles résolvent, la plupart sont dépendantes de la plateforme ou du langage utilisé. Pour résoudre ce problème, l'utilisation de modèles formels pour la génération de scénarios ou d'activités est essentielle pour garantir une expérience de haute qualité.

Dans ce contexte, l'utilisation des réseaux de Petri ou des automates est très répandue. À partir de ce constat, le modèle formel CIT (Content Interaction Time) [44], consacré au développement d'applications robotiques interactives et basé sur des réseaux d'automates temporisés à entrées/sorties a été proposé.

Nous proposons donc, dans cet article, deux plateformes logicielles, CELTIC (Common Editor for Location Time Interaction and Content) et EDAIN (Execution Driver based on Artificial INtelligence), permettant une modélisation simplifiée et générique d'applications robotiques et leur supervision sur la base du modèle CIT.

Cette approche a été utilisée pour concevoir un jeu sérieux avec un robot humanoïde Nao. Ce jeu permet de faire découvrir de manière ludique au jeune public l'exposition dédiée à l'ethnographie au Muséum d'Histoire Naturelle de La Rochelle et l'exposition consacrée à l'archéologie au Musée Sainte Croix de Poitiers.

À l'issue de ces expérimentations, des problématiques d'adaptation dynamique du scénario ont été identifiées. Nous présenterons donc le début de nos réflexions sur l'intégration d'un algorithme d'apprentissage par renforcement permettant d'optimiser la valuation de certains paramètres contrôlant l'exécution du scénario.

MOTS-CLÉS. — Modélisation formelle, réseaux d'automates, robotique, interaction homme-robot.

1. INTRODUCTION

Les musées sont continuellement en recherche de nouvelles manières d'améliorer l'expérience des visiteurs. L'utilisation de jeu sérieux a démontré son potentiel pour créer de l'engagement et amener les joueurs à améliorer leur compréhension d'un sujet [25]. Les avantages de l'apprentissage par le jeu ont donc été démontrés [21, 56]

et de nombreuses institutions se sont équipés de fonctionnalités ludiques dans cet objectif [11, 14, 16, 62].

Ces enjeux sont particulièrement forts pour le jeune public qui peut montrer un désintérêt pour les musées. Les pratiques culturelles des jeunes sont, en effet, connues pour être éloignées de celles que peuvent avoir leurs aînés, du fait de la généralisation des technologies numériques rendant l'accès à l'information et à la culture facilité et ce à tout instant de la journée, sans contrainte de grille horaire ou de programmation. Les moins de 30 ans sont ainsi 82 % à déclarer que la culture représente une place importante dans leur vie⁽¹⁾ (Figure 1.1 à gauche), mais sont paradoxalement 77 % à déclarer aller seulement de 1 à 2 fois ou moins par an dans les musées⁽¹⁾ (Figure 1.1 à droite). Ces chiffres rejoignent ceux du ministère de la culture [8] qui, dès 2007, affichaient une baisse des visites de musée et d'exposition pour les jeunes de 15-24 ans, tout en montrant une hausse pour les 25-34 ans.

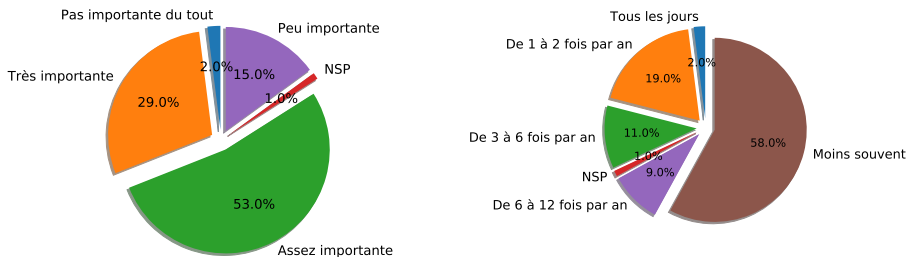


FIGURE 1.1. À gauche : Place de la culture chez les jeunes. À droite : Fréquence des visites de musée chez les jeunes

Fort de ce constat, les lieux de culture, en particulier les musées, s'emparent de plus en plus des nouvelles technologies afin de transmettre leur savoir non plus de façon unilatérale et verticale mais de façon participative et ludique. C'est bien ce que souligne Philippe Chantepie [48] : *les instances de transmission culturelle [...] sont appelées à revisiter leur modèle de médiation pour l'adapter aux jeunes générations.*

Ainsi, les robots investissent aujourd'hui les musées pour constituer une alternative aux dispositifs de diffusion d'information classiques. L'intérêt de cette nouvelle approche tient dans plusieurs points. Elle permet d'une part une interaction plus naturelle avec les visiteurs (il est plus aisé de visiter un musée sans avoir à être équipé d'une tablette, d'un téléphone ou d'autres équipements qui pourraient interférer avec la visite et la découverte des œuvres). D'autre part, elle ouvre la voie à une dynamique sociale entre les visiteurs (pouvant échanger entre eux quand bien même ils sont de simples spectateurs de l'interaction). Enfin, les systèmes de vision par ordinateur intégrés au robot permettent de détecter la présence de visiteurs et le robot peut alors les interpeller

⁽¹⁾D'après l'étude opinionway de 2016 pour Agefa PME, réalisée sur 807 jeunes de moins de 30 ans, *Les jeunes et la culture*, https://musiquesactuelles.net/wp-content/uploads/2016/02/opinionway_-_agefa-pme_-_les_jeunes_et_la_culture_-_fevrier_2016_version_de_diffusion.pdf

pour les inviter au jeu. Ainsi, un simple visiteur qui n'a pas fait la démarche de jouer sur tablette ou de s'équiper d'un dispositif de réalité virtuelle peut devenir un joueur pour une partie de sa visite.

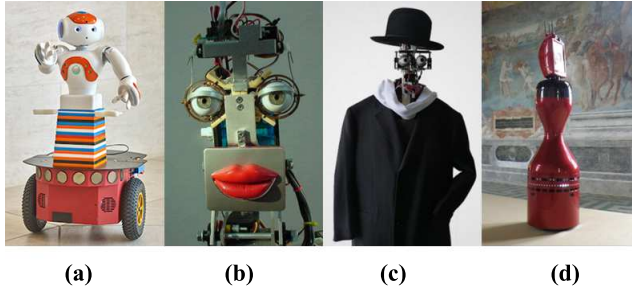


FIGURE 1.2. Expériences avec des robots dans les musées

1.1. LA TECHNOLOGIE AU SERVICE DE LA MÉDIATION MUSÉALE

Les audio-guides interactifs et personnalisés sont très répandus dans les musées afin que les visiteurs puissent obtenir davantage d'informations sur le patrimoine qu'ils visitent. Également, des équipements de médiation immersifs avec l'utilisation de la réalité augmentée [4, 41], de la 3D [54] et d'hologrammes apportent aux visiteurs une expérience riche et innovante (comme au British Museum ou au musée d'art moderne de New York par exemple).

Depuis quelques années, les robots intègrent aussi bien des œuvres artistiques [3, 32, 61] que les lieux de culture et constituent une alternative aux dispositifs numériques de diffusion d'information ou de création d'engagement auprès des visiteurs. Plusieurs expérimentations ont déjà été menées en ce sens afin d'analyser la faisabilité et l'impact de l'introduction des robots dans les lieux de culture. Ainsi, depuis 2014, deux robots sont en charge d'accueillir les visiteurs au Musée National des Sciences et Innovations émergentes à Tokyo et peuvent également leur donner les dernières nouvelles. Dans le même registre, le robot Inkha (Figure 1.2(b)) du King's College de Londres est en charge de la réception et de l'orientation des visiteurs.

Lors de l'exposition *Eppur Si Muove* au Musée d'Art Moderne de Luxembourg, un robot Nao couplé à une base mobile (Figure 1.2(a)) est une œuvre à lui seul mais guide également le public en leur présentant les œuvres exposées [26]. En France, dans le cadre d'un projet sur l'esthétisme artificiel, le robot Berenson (Figure 1.2(c)) a été déployé au musée du quai Branly. Ce robot se comporte comme un critique d'art ; il exprime une émotion lorsqu'il se trouve devant une œuvre. Son opinion évoluera grâce aux réactions des visiteurs présents autour de lui [12].

De plus, certains musées, en raison de leur architecture, ne sont pas accessibles aux personnes à mobilité réduite (et ne peuvent réaliser les travaux nécessaires). Le musée du château d'Oiron en France a appréhendé ce problème et, au premier étage

du musée, a présenté le robot Norio (Figure 1.2(d)) qui peut être contrôlé à distance par un joystick à partir du rez-de-chaussée, ce qui permet aux personnes handicapées de découvrir les œuvres qui leur sont inaccessibles [28].

Les agents conversationnels animés ont également émergés et été introduits dans diverses applications. Campano *et al.* [13] propose par exemple un personnage virtuel qui visite un musée en interagissant avec les visiteurs et permet d'apporter des informations et de les questionner. D'autres applications utilisent d'ailleurs ce type d'agents comme par exemple dans le cadre de la rééducation de patients cérébro-lésés [45] ou encore dans la recommandation de films [49].

Nous avons, au travers de notre projet « Musées 3.0 » et en collaboration avec le musée d'Histoire Naturelle de La Rochelle et le musée Sainte Croix de Poitiers, développé une expérience interactive pour les jeunes visiteurs au travers d'un jeu sérieux. Ce jeu leur permet de découvrir, d'une manière intéressante et ludique, une partie des collections, au travers d'un quiz oral proposé par un robot Nao. Le robot Nao, initialement développé par la société française Aldebaran (aujourd'hui Softbank Robotics), dispose d'un ensemble de capteurs (notamment une caméra, un sonar et des capteurs tactiles) qui lui permettent de percevoir l'environnement dans lequel il évolue. Ce robot est également capable d'interagir verbalement avec l'humain au travers de ses microphones et haut-parleurs (le robot intégrant les API de reconnaissance et de synthèse vocale). Le jeu, quant à lui, consiste en un jeu de piste dirigé par le robot qui pose des questions aux jeunes joueurs. Ces derniers partent alors dans le musée à la recherche de la réponse et retournent vers Nao pour répondre lorsqu'ils l'ont trouvée.

1.2. PROBLÉMATIQUES SCIENTIFIQUES

Ainsi, un premier prototype de jeu sérieux a été développé en juin 2016. Développé en collaboration avec le Musée d'Histoire Naturelle, le scénario consistait en un ensemble d'interactions visant à mener les visiteurs dans un parcours progressif dans le musée. Ces interactions ont notamment été produites au travers de l'outil propriétaire de la société Aldebaran (Choregraphe). Si les résultats de ces trois jours d'expérimentation se sont avérés positifs, la méthode de modélisation a montré ses limites. La complexité de modification de l'ordonnancement des actions et de modification des dialogues du robot ont fait apparaître le besoin de disposer d'un outil de scénarisation de plus haut niveau.

A. Problématiques liées à la scénarisation

❶ Les premières réflexions ont porté sur l'**extensibilité** d'un tel jeu. En effet, l'ajout ou la modification de contenus proposés dans le jeu peut s'avérer difficile de même que le déploiement pour d'autres musées. Disposer d'un outil de modélisation simplifiant la représentation et permettant une gestion externalisée des contenus est donc primordial pour faciliter l'extension de jeux.

❷ Le deuxième objectif d'un tel outil est de faciliter la **reproductibilité** d'un scénario pour d'autres lieux. L'objectif ici est de pouvoir disposer d'un ensemble

d'éléments réutilisables lors de prochaines modélisations afin de faciliter le travail du concepteur. Il s'agit donc ici d'avoir une vision modulaire de la conception.

③ Enfin, la **gestion du temps** dans les interactions, notamment homme-robot, est importante pour garantir une haute qualité de l'expérience. Il est en effet nécessaire de pouvoir gérer la dynamique de l'exécution en contraignant le temps alloué aux comportements ou aux attentes du robot. La plateforme de modélisation doit donc permettre de gérer le temps et disposer d'algorithmes de *model-checking* et de *time-checking*, afin de vérifier la cohérence temporelle du scénario.

Ces trois problématiques nous amènent à nous interroger sur la gestion des contenus (comment les représenter, les stocker, les adapter...), à l'interaction elle-même (comment représenter une interaction entre l'homme et la machine) et au temps nécessaire pour la réaliser ou pour réaliser un ensemble d'interactions.

B. Problématique liées à l'adaptation

④ À l'issue du travail de conception et d'expérimentation du scénario, une phase d'analyse de l'expérience est nécessaire pour déterminer si la session s'est déroulée dans de bonnes conditions. Cette analyse est possible grâce aux traces récoltées pendant l'exécution du scénario. L'utilisation d'algorithmes de parcours de graphe est alors nécessaire pour déterminer des parcours types et des parcours d'erreurs. Sur cette base, le scénario initial doit pouvoir être optimisé pour tenter de diminuer le nombre de parcours menant à un échec. Nous cherchons donc, dans un second temps, à disposer d'une architecture permettant de modifier dynamiquement le scénario par bouclage de pertinence, par l'utilisation de traces d'exécutions de sessions antérieures et par l'analyse du comportement de l'utilisateur courant.

1.3. ORGANISATION DE L'ARTICLE

Dans la suite de cet article, nous allons présenter en section 2 un état de l'art qui portera sur les méthodes de modélisations de la scénarisation et de leur supervision. Nous montrerons ici les enjeux et problématiques de la scénarisation d'expériences interactives. La section 3 est dédiée à la présentation du modèle CIT (Contenus, Interaction, Temps) et de la plateforme CELTIC qui est notre éditeur et générateur de scénario implémentant le modèle CIT. Notre plateforme de supervision, EDAIN, sera ensuite présentée en section 4. Elle permet d'exécuter le scénario conçu au travers de CELTIC. Ces travaux ont fait l'objet de plusieurs expérimentations dans des musées partenaires et seront présentées en section 5. Ces expérimentations nous ont permis de faire apparaître de nouvelles problématiques liées à l'adaptation du scénario au cours de son exécution. Nos premières réflexions sur ce sujet seront présentées dans la conclusion en section 6.

2. ÉTAT DE L'ART

Comme nous l'avons évoqué en introduction de cet article, nous nous intéressons ici à la problématique de conception et de supervision d'expériences interactives au travers de modèles permettant de piloter leur exécution. En effet, de nombreux travaux portent sur les architectures permettant de proposer à l'utilisateur une expérience de qualité tout en permettant au concepteur de garantir un contrôle fort sur les grandes étapes de la narration qu'il souhaite proposer [15, 47, 52]. Améliorer l'expérience du joueur au travers d'une expérience interactive revient donc à lui permettre une indépendance plus forte, c'est à dire lui rendre la liberté de contrôle sur le déroulement de l'histoire. Or, le designer souhaite lui pouvoir maîtriser le déroulé de la trame narrative et notamment se voir garantir que le joueur passera par des points narratifs précis dans un ordre qu'il décidera. L'enjeu est donc ici de proposer un modèle permettant de trouver un compromis entre le souhait de s'assurer que l'histoire vécue par le joueur est bien celle qu'il souhaite lui raconter et la volonté du joueur de maîtriser l'univers dans lequel il évolue (et de fait influencer l'histoire par ses choix).

Au delà de la seule problématique de l'interaction dans le contexte de la robotique, des travaux proposent des modélisations pour l'interaction au sens large dans d'autres contextes, comme par exemple celui des jeux. Les dimensions liées à l'interaction, les contenus, le temps et l'espace lèvent également des interrogations qui peuvent être communes avec le domaine de la robotique. En effet, un concepteur de jeu peut vouloir imposer qu'un niveau du jeu s'effectue dans un laps de temps et un lieu donné, comme il peut être nécessaire de concevoir un scénario robotique contraint sur la vitesse et les déplacements du robot.

Dans le domaine de la conception d'un système robotique complexe et comme le souligne McGanne dans [40], une approche monolithique peut difficilement être utilisée. La nécessité de diviser le modèle se fait alors ressentir. Les approches réductionnistes (également dites par composants) sont alors pertinentes et adaptées à cette problématique. Szyperski *et al.* [59] propose ainsi la définition suivante :

A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be developed independently and is subject to composition by third parties.

L'avantage majeur des approches par composants est la réutilisation. En effet, l'exécution d'un même composant peut se produire à différentes étapes de l'utilisation d'un logiciel mais dans des contextes particuliers. De même, la division par composants peut avoir des répercussions sur les coûts de production et sur les délais de mise sur le marché.

Notre état de l'art sur la modélisation de jeux et de plateforme robotique s'est donc orienté sur ces approches et plus particulièrement à deux types de modélisation, très étudiées dans la littérature [30, 37] :

- les méthodes semi-formelles avec en particulier UML (qui s'est imposé comme un standard dans la modélisation de systèmes complexes). Certaines

approches s'appuient donc sur ce formalisme pour la conception d'activités robotiques ;

- les méthodes formelles, qui utilisent un modèle mathématique pour la représentation, l'analyse, la vérification et la preuve de certaines propriétés. Ces aspects deviennent très importants et indispensables dans la conception d'activités robotiques et dans l'univers du jeu.

2.1. APPROCHES SEMI-FORMELLES - UML

Du fait de sa standardisation, UML (Unified Modeling Language) a été utilisé pour la conception d'activités robotiques, notamment sur deux plateformes, *RobotML* : *Robotic Modeling Language* et *RobotChart*.

Dans le cadre du projet de recherche français *PROTEUS* (Plate forme pour la Robotique Organisant les Transferts Entre Utilisateurs et Scientifiques), le langage de modélisation RobotML a été conçu pour faciliter la conception, la simulation et le déploiement d'applications robotiques [20]. Certaines approches étant dépendantes de la plateforme robotique, l'objectif principal de RobotML est de séparer le modèle de l'architecture robotique cible. Ainsi, un même modèle peut être déployé sur plusieurs plateformes cibles.

RobotML s'appuie sur des profils UML et des machines à états et est structuré autour de quatre packages (gestion de la trajectoire, observation de l'environnement, localisation et commande). La phase de création est réalisée grâce au plugin Eclipse dédié, dont un exemple est présenté figure 2.1.

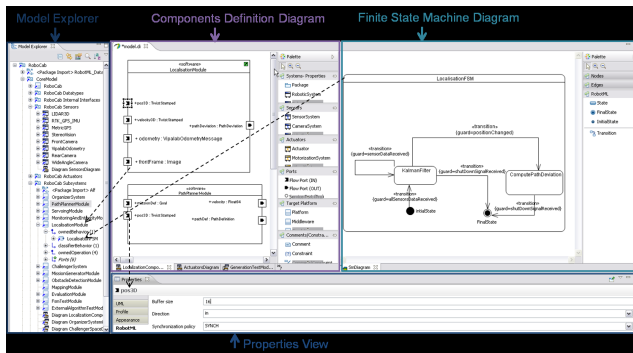


FIGURE 2.1. Environnement de modélisation de RobotML

Tout comme RobotML, RoboChart [42, 43] utilise une sémantique semi-formelle via une représentation UML (machines à états). Les primitives temporelles utilisées reposent sur la sémantique proposée dans les automates temporisés [1] et sur les CSP temporisés [57] (Communicating Sequential Processes). La spécification d'un système robotique sous RoboChart repose sur un ensemble de modules connectés à un ou plusieurs contrôleurs, dont la représentation semi-formelle est assurée par une

machine à états UML. RoboChart dispose d'un outil de modélisation : *RoboTool*. *RoboTool* est un ensemble de plugins pour l'éditeur Eclipse. Cet outil permet de concevoir et de visualiser le modèle robotique, sous sa forme textuelle et graphique et permet de traduire le modèle en CSP. La vérification du modèle est alors possible en utilisant l'outil FDR (Failures Divergences Refinement) [23].

Analyse globale des approches semi-formelles : RobotChart et RobotML utilisent les machines à états pour la modélisation de plateformes robotiques. L'utilisation de l'UML permet une rapide compréhension du modèle. De plus, la dissociation du modèle de l'architecture robotique est un atout majeur. Cependant, UML ne propose pas de sémantique formelle permettant la vérification de propriétés, ce qui en limite son utilisation dans notre cas.

2.2. APPROCHES FORMELLES

L'utilisation de méthodes formelles se justifie par le fait qu'elles utilisent, contrairement aux méthodes semi-formelles, un modèle mathématique pour l'analyse, la vérification et la preuve de certaines propriétés. Cette analyse est nécessaire afin de garantir une haute qualité de l'activité, au travers de propriétés fondamentales. Nous retrouvons également ces problématiques dans la conception de jeux.

L'ensemble des approches proposées mettent en évidence la nécessité de disposer d'outils performants, contrôlant notamment certaines propriétés de sûreté. En effet, l'utilisation de robots en interaction avec des humains peut engendrer des événements non désirables et potentiellement dangereux. De ce fait, nous nous attardons sur deux catégories particulières :

- les approches à base de réseaux de Petri, orientées ressources ;
- les approches à base de réseaux d'automates, orientées actions.

2.3. APPROCHES BASÉES SUR LES RÉSEAUX DE PETRI

La notion de composants, que nous avons décrite en introduction, est pertinente lorsque l'on se place dans le domaine de la conception d'un jeu. Celle-ci s'intègre parfaitement à travers certains concepts propres au domaine. En effet, contrairement à la modélisation des processus d'un logiciel quelconque, celle d'un jeu entraîne la spécification d'activités plus spécifiques, telles que les niveaux ou les quêtes. Des contraintes s'ajoutent également du fait de l'obligation ou non de réaliser certaines tâches pour atteindre un but. Ainsi, nous retrouvons dans la littérature des travaux où la modélisation d'un jeu est divisée en niveaux ou quêtes, que l'on peut comparer aux approches réductionnistes (ou par composant) quand on se place dans la conception d'activités robotiques.

De Oliveira *et al.* [18] propose, par exemple, une structuration du jeu sous forme de quêtes modélisées par Workflow nets [63]. Chacune des quêtes peut être obligatoire ou optionnelle et représente une suite d'activités pouvant être réalisées de manière séquentielle ou parallèle. Ces activités peuvent elles-mêmes être optionnelles. La

logique linéaire [24] est ensuite utilisée pour vérifier la robustesse du scénario modélisé. Barreto *et al.* [6] propose d'étendre l'approche précédente en ajoutant, à la modélisation des activités du joueur (par workflow net), une modélisation topologique de l'espace de jeu. Cette représentation est réalisée par un réseau de Petri où les places représentent les différentes régions où l'action doit se dérouler et le jeton, la position du joueur. Des places supplémentaires peuvent être ajoutées afin de modéliser les conditions d'un passage d'une zone de jeu à une autre (par exemple, l'obtention d'une clef). Si cette représentation s'avère efficace, elle n'est pas utilisable seule. En effet, une mise en relation des représentations des activités du joueur et de l'espace du jeu est nécessaire et permettra une gestion des interactions entre le joueur et l'univers virtuel.

Cette double modélisation permet également d'imposer aux joueurs d'être dans une zone particulière pour effectuer des actions. Les auteurs proposent donc une dernière modélisation pour faire le parallèle entre modélisation formelle des activités du joueur et de l'espace de jeu. Ce parallélisme est mis en œuvre par un réseau de Petri coloré par l'utilisation du concept des fusions de places. La vérification de propriétés de robustesse, notamment la vérification de l'accessibilité de chaque zone de jeu ainsi que l'exécution de toutes les activités du joueur, est mis en œuvre grâce aux algorithmes contenus dans *CPN Tools* [27]. Cependant, si ce n'est la gestion des ressources, aucun autre contenu n'est ici proposé. Les niveaux et activités ne sont également pas temporisés.

Les réseaux de Petri et leurs extensions temporisés ont été utilisés pour la modélisation de système à événements discrets et plus particulièrement au domaine de la robotique [29, 34]. Les architectures de contrôle basées sur des modélisations multicouches, simplifiant l'étape de conception, sont d'ailleurs répandues. Elles prennent en compte la planification, la coordination et l'exécution de tâches.

Ainsi, [17] propose une modélisation simplifiée et divisée en trois couches, où chacune décrit les éléments à différent niveau d'abstraction :

- **Environnement** : chaque état de l'environnement est représenté par un réseau de Petri marqué ;
- **Exécuteur d'actions** : l'ensemble des actions exécutées sur le robot et modélisées par un réseau de Petri. Une action a un impact sur l'environnement et peut être conditionnée (pre-conditions, running-conditions et success-conditions) ;
- **Coordinateur d'actions** : il planifie les actions du robot par un réseau de Petri de plus haut niveau.

Le modèle final est alors obtenu par la composition des réseaux de Petri issus des trois couches précédentes. Le temps nécessaire à la composition est proportionnel aux nombres d'états des différents réseaux. Cependant le temps d'analyse augmente de façon exponentiel par rapport aux nombres d'états. Cette analyse (probabilité qu'une condition soit respectée, temps nécessaire pour effectuer une action...) est réalisée par différents algorithmes proposés dans [7, 46, 64].

Analyse globale des approches à base de réseau de Petri : Nous venons de présenter plusieurs approches de conception de jeux et d'activités robotiques à base de réseaux

de Petri. À travers celles-ci, nous voyons que ce modèle est adapté lorsque l'on adopte une vision orientée ressources. Elles mettent en évidence l'efficacité des approches réductionnistes qui consistent à concevoir les activités indépendamment les unes des autres. Contrairement aux approches semi-formelles, des outils, tel que CPN Tools, ou la conversion en logique linéaire, permettent d'effectuer une analyse sur les modèles.

Nous allons maintenant nous placer du point de vue événements et actions en présentant les approches basées sur des réseaux d'automates.

2.4. APPROCHES BASÉES SUR LES RÉSEAUX D'AUTOMATES

L'utilisation des automates est également présente pour la conception de plateformes robotiques. Ainsi, [68] propose un système de planification multi-couches destiné à la robotique (figure 2.2) dans lequel les comportements du robot et son environnement sont modélisés par des réseaux d'automates temporisés.

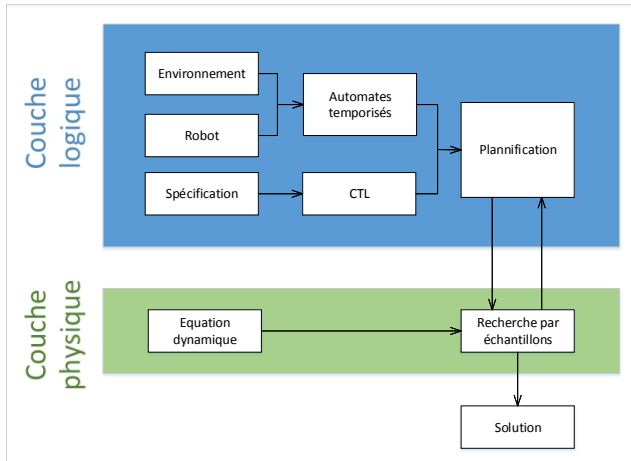


FIGURE 2.2. Architecture de l'approche de [68]

La couche logique est en charge de construire la représentation de l'environnement et des comportements du robot par des automates temporisés. Les propriétés de vérification sont exprimées en logique CTL et sont vérifiées avec l'outil UPPAAL [33]. Deux types de propriétés sont ici vérifiées :

- **propriétés d'accessibilité** : tous les robots doivent pouvoir atteindre leur destination ;
- **propriétés de sûreté** : l'évitement d'obstacles et la durée d'exécution.

La vérification de ces propriétés par UPPAAL permet de générer une trace, satisfaisant l'ensemble des contraintes. Ces traces sont alors utilisées dans la seconde couche, la couche physique, qui va générer une trajectoire pour chaque robot sur la

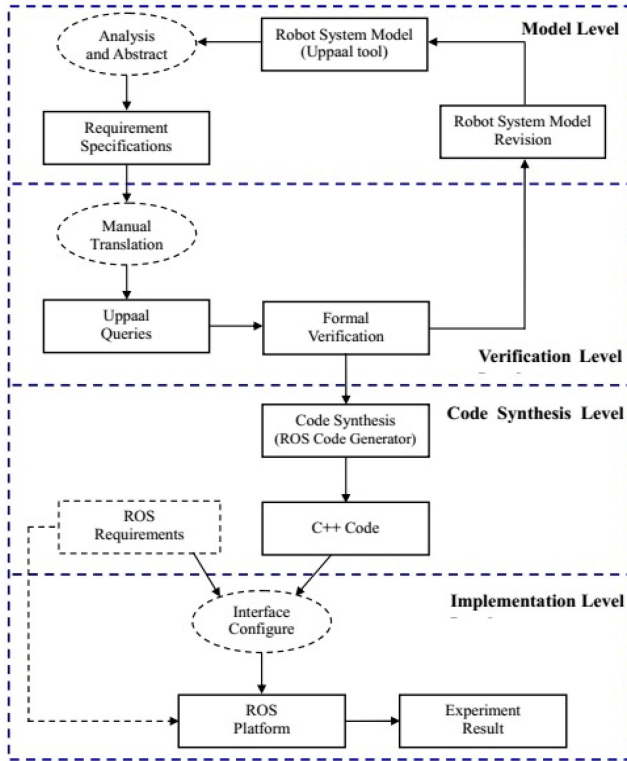


FIGURE 2.3. Conception d'un système robotique issu de [67]

base d'équations dynamiques. Une trace de contre exemple peut également être générée lorsqu'une contrainte n'est pas respectée.

Ces travaux ont été étendus dans [67]. En effet, jusqu'ici l'approche proposée est restée théorique et n'avait pas fait l'objet d'expérimentations. Une nouvelle approche en quatre couches a donc émergé (figure 2.3) :

- la couche « modèle » que l'on peut comparer à la couche logique de l'approche précédente ;
- la couche de vérification sous forme de requête CTL sous UPPAAL ;
- la couche de génération de code. Dans celle-ci, le générateur produit automatiquement le code C++ pour la plateforme ROS ;
- la couche d'implémentation permettant l'exécution du code sur le simulateur Gazebo.

Le principe de conception multi-couches a également été utilisé dans [51, 52]. Afin de réduire la complexité liée à la modélisation du scénario, cette approche propose une conception automatique du scénario par la définition d'un ensemble de comportements

génériques et de scènes d'exécutions. Le processus de création d'un scénario interactif est ici décomposé en trois couches (figure 2.4).

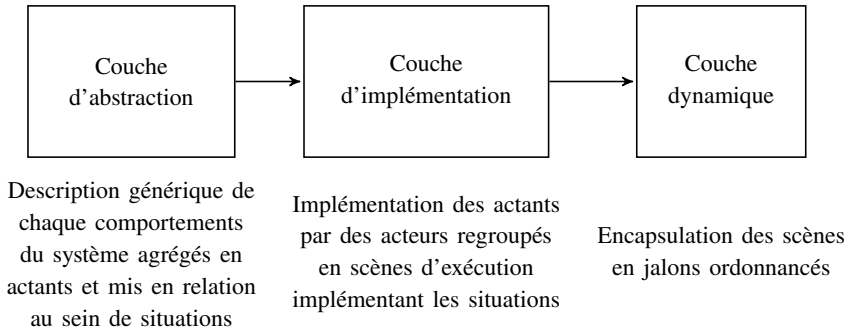


FIGURE 2.4. Modèle à trois couches de [51]

Un enjeu majeur de la conception de jeux porte sur la facilité d'extension ou de modification que ce soit au niveau de la structure ou au niveau des entités narratives. La structuration en trois couches de cette approche permet de répondre à cette problématique. Les concepts d'extensibilité et de réutilisation des entités génériques par héritage multiple sont un atout majeur. Bien que cette approche ait prouvé son efficacité, la conception d'un scénario nécessite de longues étapes qui peuvent être un frein pour le concepteur. De plus, la dimension temporelle n'est pas prise en compte, que ce soit au niveau des comportements ou au niveau des scènes pour le passage de l'une à l'autre. Enfin, la gestion des contenus n'est ici pas supportée.

Analyse globale des approches à base de réseau d'automates : À travers les approches que nous venons de présenter, nous remarquons que les automates, grâce aux différents mécanismes de composition et de synchronisation, sont adaptés à la modélisation modulaire des activités. Leurs formalisme permet également d'appliquer une vérification du modèle afin de prouver leur robustesse. Enfin, la possibilité de pouvoir contraindre temporellement les comportements est un atout majeur quand on sait l'importance de ces paramètres dans le domaine des interactions homme-robot.

2.5. POSITIONNEMENT

Nous venons de présenter un état de l'art dans lequel différentes méthodes de conception et de modélisation ont été détaillées. Bien que ces approches s'avèrent efficaces, nous avons pu établir que les modèles sur lesquels elles se basent n'adressent pas l'ensemble des critères énoncés dans notre problématique.

Les approches que nous trouvons dans l'état de l'art sont principalement basées sur des modèles spécifiques aux questions abordées (planification, narration...). Celles ci reposent sur des modèles formels permettant la vérification de propriétés. L'état de l'art montre la nécessité de disposer d'outils de modélisation efficace permettant une conception modulaire du système ou du scénario. Si la modélisation UML facilite la

compréhension et la lecture d'un système, la vérification de certaines propriétés telles que la sûreté ou l'accessibilité s'avère parfois difficile voire impossible. Les approches à base de réseaux de Petri sont, quant à elles, orientées ressources. Notre problématique nous imposant plutôt une représentation par actions des comportements robotiques, notre choix de formalisme pour la spécification des comportements robotiques s'est donc orienté vers une représentation par automates temporisés.

Concevoir de manière formelle une activité robotique garanti au concepteur d'obtenir un modèle répondant à ses attentes. Basée sur des mathématiques, une représentation formelle de l'activité permet également de s'assurer, à travers des techniques de *model-checking*, que le modèle répond à des critères de sûreté. Il est alors envisageable de piloter un processus quelconque et notamment une plateforme robotique.

La section suivante tient dans la proposition d'un modèle générique prenant en compte l'ensemble des dimensions de l'interaction : la représentation des contenus, l'interaction (au travers des comportements) et le temps. Un premier modèle, CITE, proposé dans [50] avait pour ambition de prendre également en compte l'espace dans la conception d'un scénario. Cette problématique constitue, entre autre, une intégration future à envisager pour notre modèle mais n'entre pas dans le cadre des travaux que nous présentons dans cet article.

3. CELTIC : ÉDITEUR DE MODÉLISATIONS D'EXPÉRIENCES INTERACTIVES

CELTIC (Common Editor for Location Time Interaction and Content) est un éditeur générique de production de modèle pour des expériences interactives reposant sur le modèle CIT (Content Interaction Time) [44]. Il peut s'agir d'un jeu, d'une application ou comme dans notre cas d'étude, d'une expérience de jeu avec un robot. L'objectif est de proposer une modélisation générique et modulaire qui permette de mettre en œuvre une supervision et une analyse des traces des utilisateurs. Ce modèle vise de plus à prendre en compte les différentes composantes de l'interaction : le temps, l'espace et le contenu.

3.1. PRINCIPE DU MODÈLE CIT

Notre approche est, elle aussi, basée sur des réseaux d'automates temporisés à états finis, très adaptés pour représenter des systèmes synchrones avec des contraintes de temps. Nous simplifions la tâche de modélisation par une méthode en deux étapes :

- tout d'abord, la description de l'expérience est réalisée de manière abstraite, en définissant les entités réutilisables (les comportements et les patterns), modélisées par des automates temporisés. Cette étape crée la *couche déclarative* ;
- par l'instanciation des entités génériques dans des agents, regroupés dans des contextes d'exécution ordonnés, la *couche d'implémentation* est alors définie.

Le processus de création d'un système interactif est illustré dans la figure 3.1 et 3.2 et est détaillé dans les sections 3.2 et 3.3.

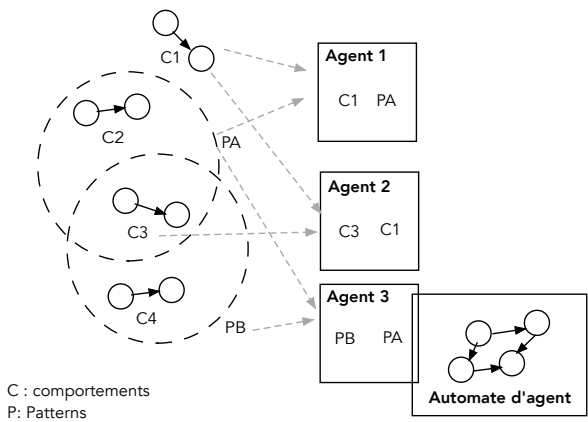


FIGURE 3.1. Principe de création des comportements, patterns et agents

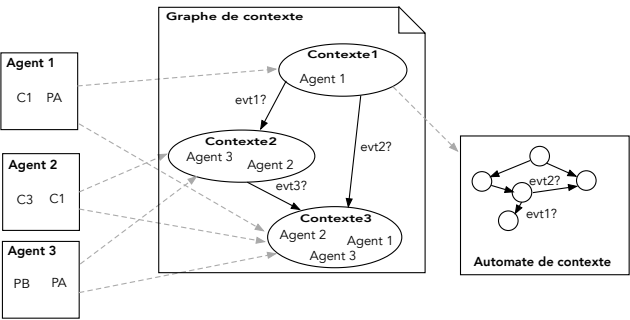


FIGURE 3.2. Principe de création des contextes et du graphe de contexte

3.2. LA COUCHE DÉCLARATIVE

La couche déclarative est celle dans laquelle nous définissons les entités génériques du système modélisé, à savoir les comportements atomiques qui peuvent être exécutés. Ces comportements sont associés à des variables déclaratives (des entiers, des chaînes de caractères ou des booléens) et à des messages (channels), qui sont notamment les signaux permettant de synchroniser plusieurs comportements pour la composition dynamique. Les comportements sont ensuite regroupés en patterns qui représentent un ensemble de comportements, spécifiques à l'entité, qui peuvent être réutilisés par le principe d'héritage multiple.

L'enjeu majeur de cette couche déclarative est de permettre la représentation de comportements atomiques et de patterns de comportements. Ceux-ci peuvent être réutilisés et composés dynamiquement au sein d'agents dans la couche d'implémentation.

3.2.1. Les comportements

Dans notre approche, un comportement atomique est une transition d'automate temporisé [33] qui viendra s'agréger avec tous les autres comportements possibles de l'agent. Par exemple, nous souhaitons représenter un premier comportement permettant au robot de se lever (dans un délai compris entre 2 et 5 unités de temps) et un second lui permettant de marcher pendant une durée de 3 à 10 unités de temps. Les modèles de ces comportements sont représentés sur la figure 3.3. Dans cet exemple, x représente le temps.

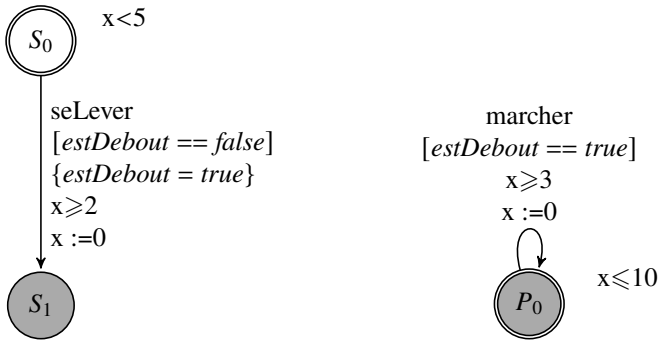


FIGURE 3.3. Automate temporisé des comportements *seLever* et *marcher*

3.2.2. Les patterns

Les patterns décrivent un ensemble de comportements susceptibles d'être exécutés conjointement au sein de l'application et sont implémentés par des agents. Un automate temporisé du pattern est alors obtenu par le produit synchronisé de l'ensemble des automates des comportements qui le composent (selon l'algorithme de synchronisation décrit dans [9]). Cette composition permet au système d'évoluer par la synchronisation de deux entités ou par leur évolution individuelle. Un exemple d'automate obtenu à partir des comportements définis précédemment est donné à la figure 3.4 (pattern *Promenade*).

3.3. LA COUCHE D'IMPLÉMENTATION

La couche d'implémentation utilise les comportements et patterns décrits en amont pour concevoir les entités impliquées dans l'expérience interactive : les agents et les contextes d'exécution. Il s'agit, dans un premier temps, de construire des agents qui implémentent des comportements et des patterns. Ces agents sont ensuite liés dans des contextes d'exécution représentant une situation spécifique dans lequel les agents interagissent entre eux.

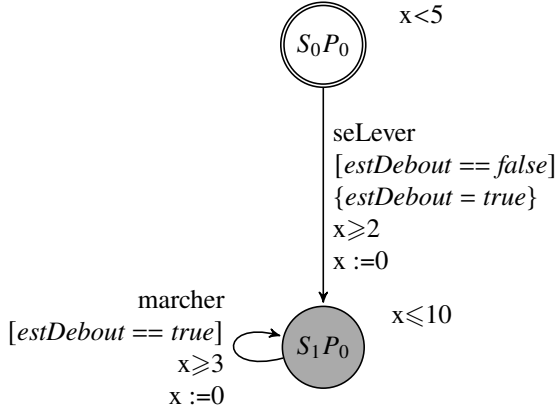


FIGURE 3.4. Automate temporisé du pattern *promenade*

3.3.1. Les agents

Un agent peut implémenter le rôle d'un ou plusieurs patterns grâce au mécanisme d'héritage multiple. Il permet aussi d'implémenter des comportements atomiques définis dans la couche déclarative. Ainsi, le concepteur peut spécialiser un agent grâce aux différents comportements qu'il implémente. L'automate de comportement de l'agent sera obtenu par synchronisation de l'ensemble des comportements. L'enjeu majeur de cette couche est de garantir la réutilisabilité des comportements. Un agent spécifique à une expérience interactive donnée peut utiliser des comportements atomiques conçus pour une autre expérience.

Les comportements, agrégés par un agent, proposent des valeurs par défaut lors de leur définition dans la couche déclarative. Ceux-ci peuvent ne pas correspondre à l'attente du concepteur pour un agent particulier. Il peut alors en modifier les valeurs. Enfin, l'agent permet de spécifier, pour chacun des comportements qu'il agrège, le contenu réel qui sera exécuté sur le processus piloté.

SPÉCIFICATION DES COMPORTEMENTS. — La spécification des comportements se réalise par l'utilisation d'une base de données externe (figure 3.5) intégrant l'ensemble des actions exécutables sur le processus piloté ainsi que les divers contenus, notamment textuel.

Exemple. — Nous définissons un agent *promeneur* implémentant le pattern *promenade* et le comportement atomique *asseoir*, qui déclenchera le signal *assis*. Le concepteur doit donc spécifier les trois comportements que l'agent implémente. Pour cela, il utilise la base de données et en particulier la table *Behavior* contenant l'ensemble des actions exécutables sur le processus. Ainsi, pour chaque comportement, il va spécifier l'action à exécuter. Pour être exécutée, certaines actions possèdent des paramètres (un dictionnaire et un temps d'écoute pour la reconnaissance vocale par exemple) et des sorties (le mot reconnu et un taux de confiance pour la reconnaissance

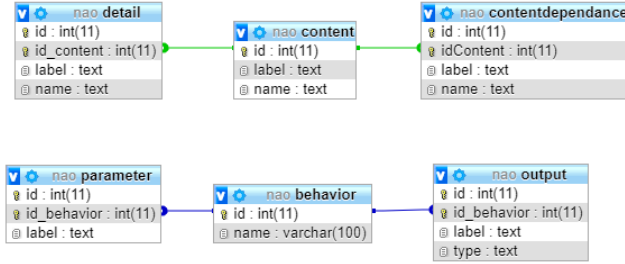
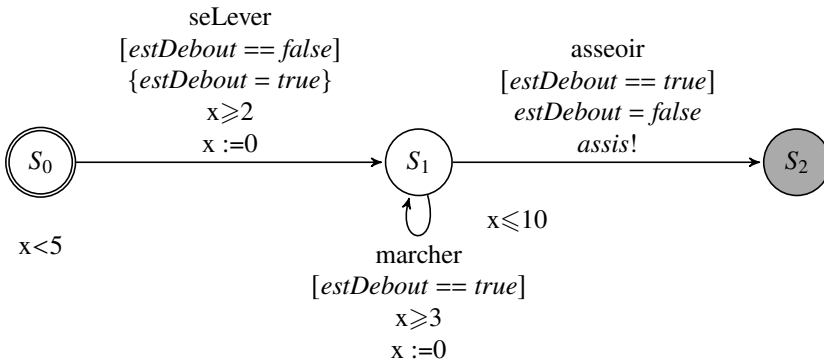


FIGURE 3.5. Base de données des contenus

vocale). Ces paramètres doivent également être spécifiés par le concepteur. Enfin, si le concepteur veut utiliser des contenus, par exemple textuels lors de l'utilisation de la synthèse vocale, le concepteur utilisera la table *Content*.

FIGURE 3.6. Automate temporisé de l'agent *promeneur*

3.3.2. Les contextes et le graphe de contexte

Le contexte décrit une situation dans laquelle un certain nombre d'agents va être impliqué. L'automate de contexte sera alors construit par synchronisation de l'ensemble des automates des agents présents. Le scénario global d'exécution est produit par le graphe de contexte. Il s'agit d'un automate temporisé de haut niveau qui représente les passages entre contextes d'exécution (un exemple est donné à la figure 3.7).

Cette couche d'implémentation agrège d'abord les comportements dans les agents, puis les agents dans les contextes et produit dynamiquement les automates de comportement de ces derniers par synchronisation.

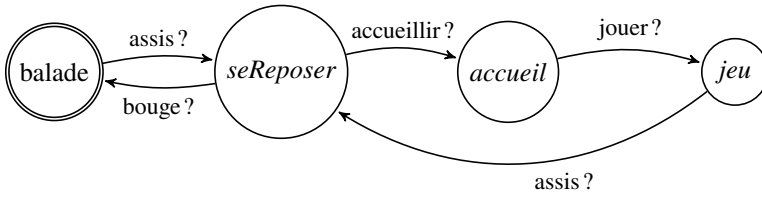


FIGURE 3.7. Graphe de contexte

3.4. L'ÉDITEUR

Notre plateforme se compose d'un éditeur web, utilisant le framework *Symfony*, pour la conception de scénario interactif. Nous avons souhaité rendre facilement accessible notre plateforme à la communauté scientifique. Le choix de la technologie web était donc évident.

Cet éditeur se compose de deux interfaces principales :

- La première est dédiée à l'édition du scénario. Dans cette interface, le concepteur du scénario va construire son expérience pas à pas grâce à notre modélisation à deux couches ;
- La seconde est dédiée à la gestion de la base de données et notamment aux différents contenus proposés lors de l'expérience.

3.5. GÉNÉRATEUR D'AUTOMATES

Sur la base du modèle décrit par l'éditeur et sérialisé au format XML, le générateur d'automates de la plateforme CELTIC est en charge de générer l'automate temporisé communicant de chaque entité du système. Ces automates sont ensuite combinés pour répondre aux spécifications du concepteur grâce aux différents algorithmes de composition qu'il intègre. À l'issue de la génération de chaque entité, le générateur sérialise à nouveau au format XML le scénario et produit ainsi le fichier de supervision. Ce fichier intègre l'ensemble des canaux de communication, les variables, l'automate temporisé de chaque contexte ainsi que le graphe de contextes de haut niveau.

4. EDAIN : L'ARCHITECTURE DE SUPERVISION

La supervision d'une tâche en robotique consiste à prendre les décisions adéquates en fonction des différents événements qui peuvent se produire. Lorsque les robots interagissent avec un humain, les tâches deviennent alors collaboratives et nécessitent la prise en compte de spécificités particulières. En effet, le succès de l'exécution de la tâche n'est alors plus seulement due au robot mais également à la réalisation de l'activité attendue de l'humain. Par exemple, lors d'un échange vocal entre le robot et l'humain, la tâche ne peut être réussie que si le robot est en phase d'écoute mais également si l'humain lui même décide d'interagir avec le robot en lui parlant. Il est donc nécessaire de prendre en compte toutes les issues possibles lorsque nous

souhaitons superviser une activité robotique. Nous détaillons donc, dans cette section, comment, à partir du modèle à base d'automates, est supervisée et pilotée l'activité dans notre approche.

EDAIN (Figure 4.1) est l'outil de supervision associé au modèle CIT. Il permet d'établir une connexion sur les robots Nao et Pepper. Le mécanisme de communication permettant le dialogue entre les robots et EDAIN repose sur une communication client-serveur. Un serveur développé en Python a été embarqué sur les robots tandis qu'un client JAVA est en charge d'établir la connexion depuis l'outil EDAIN.

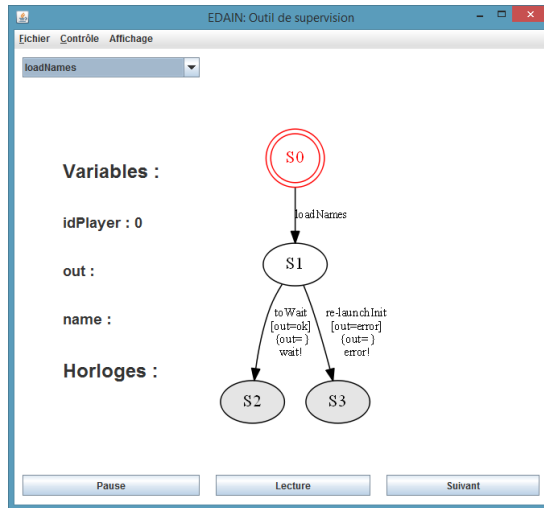


FIGURE 4.1. Interface du superviseur EDAIN

Après avoir renseigné les adresses de la base de données et du robot ainsi qu'un fichier où seront sauvegardés tous les logs, l'interface du superviseur EDAIN demande à l'utilisateur de charger le fichier de supervision. Les différents automates sont alors représentés graphiquement et l'outil charge le contexte initial (figure 4.1). L'ensemble des variables et horloges est alors affiché ainsi que l'automate du contexte actuel. EDAIN va alors exécuter, parmi les transitions possibles, celle répondant aux contraintes de temps et de variables.

Lorsqu'une action doit être exécutée sur le robot, EDAIN récupère en base de données le nom du module à exécuter et éventuellement les valeurs des contenus à proposer à l'utilisateur et va se connecter au serveur du robot pour lui transmettre les paramètres de pilotage. L'échange d'information se réalise au travers d'un socket et utilise le format JSON. Par exemple, lors de l'exécution du module de reconnaissance vocale, EDAIN envoie l'information ci dessous au robot :

```
{"speechReco":{"parameters":{"time":"15","dictionnary":"oui,non"}}
```


Du côté du robot, à la réception de ces informations, le système va exécuter le script *speechReco.py* en lui passant les paramètres associés. À l'issue de l'exécution de ce script, le robot renverra à EDAIN le mot reconnu ainsi qu'un taux de confiance associé dans le même format :

```
{"out":{"wordRecognized":"oui","confidence":"0.3216"}}
```

Grâce aux informations reçues, EDAIN met à jour le vecteur d'états des variables du modèle et peut ainsi choisir la prochaine transition à exécuter. Lorsqu'une synchronisation entre contexte est réalisée, le contexte cible est chargé sur l'interface et le même procédé est reproduit.

5. EXPÉRIMENTATIONS PUBLIQUES ET ANALYSES DES SITUATIONS TYPES

Dans cette section, nous allons présenter les différentes expérimentations que nous avons menées au Muséum d'Histoire Naturelle de La Rochelle et au Musée Sainte-Croix de Poitiers. Ces expérimentations nous ont permis d'éprouver la méthode de modélisation proposée et de vérifier la généricité de représentation d'une expérience interactive, la réutilisabilité des éléments de la couche déclarative d'une expérience à une autre et la facilité avec laquelle les contenus sont décorrelés du modèle (au travers de l'architecture basée sur la base de données présentée à la figure 3.5 de la section 3). Une autre validation porte sur l'exactitude de notre algorithme de composition dynamique des comportements pour construire les agents et les comportements du système au sein des contextes et sur la qualité du processus de supervision permettant d'exécuter le graphe de contexte.

La première expérimentation a eu lieu lors de la fête de la science au muséum d'Histoire Naturelle de La Rochelle. Le scénario créé pour l'occasion, en collaboration avec les équipes du musée, avait pour objectif de faire découvrir certaines œuvres exposées, sous forme de quiz proposés par les robots Nao (figure 5.1).



FIGURE 5.1. Expériences au muséum d'Histoire Naturelle de La Rochelle

La seconde expérimentation, dédiée au musée Sainte Croix de Poitiers, s'est déroulée lors de la Gamers Assembly. Nous proposons à cette occasion, un nouveau scénario interactif, faisant intervenir les robots Nao et des jeux développés sur tablettes pour l'occasion.

Nous détaillons dans les sections 5.1 et 5.2 ces deux expérimentations.

5.1. MUSÉUM D'HISTOIRE NATURELLE DE LA ROCHELLE

La première expérimentation s'est déroulée au Muséum d'Histoire Naturelle de La Rochelle, pendant trois après midi, lors de la fête de la Science. L'objectif était de présenter trois œuvres exposées au public sous forme de quiz. À chaque question, le joueur était invité à parcourir le musée pour trouver la réponse et revenir vers le robot. Les joueurs pouvaient interagir avec Nao à n'importe quel moment, par simple reconnaissance visuelle du badge qu'ils portaient. Nao continuait alors la séquence du jeu en interrogeant sa mémoire interne.

Nous avons proposé deux expérimentations, basées sur le même modèle (seuls les contenus textuels en base de données ont été modifiés) :

- la première, destinée aux enfants leur faisait découvrir l'exposition consacrée aux monstres, imaginaires ou réels ;
- la seconde, destinée aux adolescents et adultes, proposait de découvrir l'exposition permanente consacrée à l'archéologie.

5.1.1. *Présentation du scénario*

L'expérimentation débutait par la mise en attente de Nao. Lorsqu'un joueur voulait interagir avec lui, un contact tactile sur le capteur situé sur la tête de Nao, lui permettait de s'éveiller. Nao est capable de détecter la présence de personnes grâce à ses caméras, mais nous n'avons pas souhaité utiliser cette fonctionnalité du fait du grand nombre de visiteurs pouvant passer devant lui. Après avoir déclenché l'exécution du scénario, le visiteur était invité à présenter son badge à Nao pour s'identifier. Cette procédure était à renouveler à chacune des phases du jeu. Au début de la partie, Nao ne connaissait pas le joueur et lui présentait donc le déroulement de la partie. Il enregistrait alors en mémoire l'identifiant de son badge ainsi que l'identifiant de la question à laquelle il devait répondre (dans ce cas, la première). Lorsque Nao reconnaissait le joueur, il répétait la question du quiz et attendait la réponse (un exemple de contenu est proposé dans le tableau 5.1). Un indice était donné au joueur en cas de mauvaise réponse. Dans le cas contraire, des détails sur l'œuvre étaient énoncés et la question suivante était posée, avec sauvegarde en mémoire. À la fin du quiz, le joueur obtenait son temps de jeu par Nao ainsi que son classement provisoire.

Question	Dans les caisses en bois, trouve un grand objet qui aurait pu appartenir à une licorne. En réalité à quel animal appartient-il ?
Réponse attendue du joueur	Narval
Explication de l'œuvre par Nao	Ses grands doigts lui permettent de chercher les insectes qui creusent des galeries dans le bois des arbres.

TABLE 5.1. Exemple de contenu proposé lors de l'expérimentation au Muséum de La Rochelle

Ce scénario a été modélisé par notre outil **CELTIC** et est composé de huit contextes, comme indiqué sur la figure 5.2 :

- **Init** : le robot effaçait les anciennes parties de sa mémoire s’il y en avait. En cas d’erreur, il recommençait ;
- **Attente** : le robot se mettait en attente d’un joueur, qui lui appuyait sur la tête pour lancer le début du jeu. Il y avait un bouclage sur ce contexte toutes les quinze secondes, en cas d’inactivité ;
- **Accueil** : dans ce contexte, le robot accueillait le joueur par un petit mot de bienvenue et se présentait ;

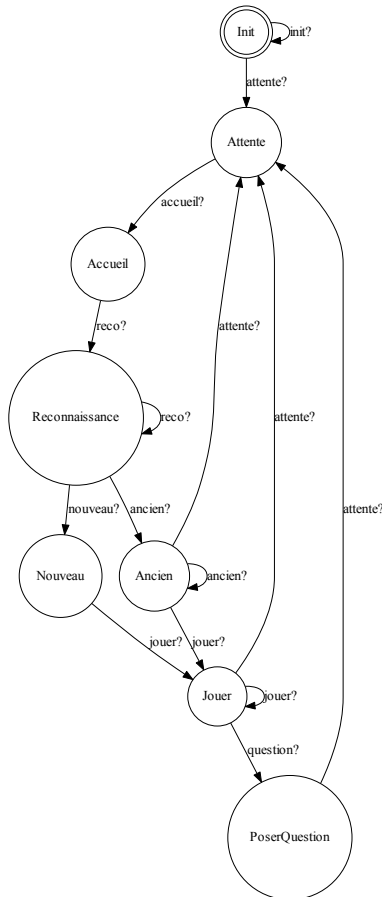


FIGURE 5.2. Graphe de contexte des expériences du Muséum d’Histoire Naturelle de La Rochelle

- **Reconnaissance** : après avoir été accueilli par Nao, le joueur devait s'identifier en présentant, au robot, le badge qu'il avait précédemment récupéré à l'accueil du musée. Sur celui-ci, un NaoMark unique lui permettait de s'identifier (chaque NaoMark possède un identifiant, représenté par un entier). Des tests de reconnaissance faciale ont été réalisés, mais l'absence d'une bonne luminosité dans le musée nous a contraint à abandonner cette approche ;
- **Nouveau** : lorsque Nao ne reconnaissait pas le joueur, le contexte *Nouveau* était exécuté. Celui-ci permettait d'expliquer le jeu et son déroulement. Si aucun NaoMark n'était détecté, le contexte était exécuté une nouvelle fois ;
- **Ancien** : dans ce contexte, Nao avait reconnu le joueur. Il lui reposait alors la question à laquelle il devait répondre et attendait la réponse. Si la réponse n'était pas correcte ou si un problème de reconnaissance vocale se produisait, le contexte était à nouveau exécuté. Dans le cas d'une réponse correcte, le robot donnait des détails supplémentaires sur l'œuvre répondant à l'énigme. Enfin, si le joueur répondait à la dernière question du quiz, le robot le félicitait, effaçait de sa mémoire l'identifiant du joueur (le badge pouvait alors servir à un nouveau joueur) et retournait en *Attente*. Dans le cas contraire, le couple <identifiant du joueur, numéro de la question suivante> était sauvegardé dans la mémoire du robot ;
- **Jouer** : cette étape consistait à demander au joueur s'il voulait continuer ou non la partie. Si oui, le contexte *PoserQuestion* était exécuté sinon, le robot se remettait en *Attente* ;
- **PoserQuestion** : une nouvelle question était posée au joueur puis le contexte *Attente* était à nouveau chargé.

5.1.2. Limites de l'expérimentation

Lors de ces trois jours d'expérimentations, nous avons rencontré des problèmes à la fois sur le plan technique et sur le plan de la conception même du scénario. Du fait de la configuration du Muséum, il était impossible d'isoler Nao dans une pièce. Cet inconvénient a eu pour conséquences des difficultés de reconnaissance vocale. En effet, Nao était placé au milieu des œuvres dans un bruit ambiant élevé. La qualité de l'interaction entre le joueur et le robot a été impactée. De plus, l'exécution du scénario est dépendante de certains paramètres contrôlant l'activité. La valuation de ces paramètres (de temps ou seuils de reconnaissance vocale) est réalisée par le concepteur et peut, dans certains cas, ne pas être adaptée à l'environnement ou au joueur.

Notre outil de supervision **EDAIN**, présenté dans la section 4 intègre un système d'observation des comportements de l'utilisateur et du scénario, afin de générer des traces d'exécution. Les logs obtenus lors de ces expérimentations font apparaître les problématiques que nous avons présentées précédemment. Chaque événement produit par l'outil de supervision était sauvegardé dans un fichier plat ainsi que l'ensemble des retours issus de Nao (Figure 5.3). Pour l'expérimentation réalisée au Muséum de La Rochelle, nous avons obtenu 6 500 lignes de logs.

```

1 31 mar. 2018 14:50:47; joueur:4 nouveauJoueur; parameters: [gesture: animations /
   Stand/BodyTalk/Speaking/BodyTalk_11, speed: 90, text: Je ne pense pas te
   connaitre. Afin de mieux faire connaissance, peux tu me donner ton prenom.
   Attends bien le petit son que je vais emettre avant de me repondre.]
2 31 mar. 2018 14:51:02; joueur:4 nouveauJoueurBack; parameters: [out: ok]
3 31 mar. 2018 14:51:03; joueur:4 ecouterPrenom; parameters: [time: 15]
4 31 mar. 2018 14:51:12; joueur:4 ecouterPrenomBack; parameters: [confidence:0
   .500299990177, name:theophile]
5 31 mar. 2018 14:51:13; joueur:4 donnerPrenomNouveau; parameters: [gesture:
   animations / Stand/BodyTalk/Speaking/BodyTalk_11, speed: 90, text: theophile]
6 31 mar. 2018 14:51:18; joueur:4 donnerPrenomNouveauBack; parameters: [out: ok]
7 31 mar. 2018 14:51:19; joueur:4 sauvegarderPrenom; parameters: [id: 4, name:
   theophile]
8 31 mar. 2018 14:51:20; joueur:4 sauvegarderPrenomBack; parameters: [out: ok]

```

FIGURE 5.3. Exemple de traces d'exécution enregistrées lors des interactions entre un joueur et Nao

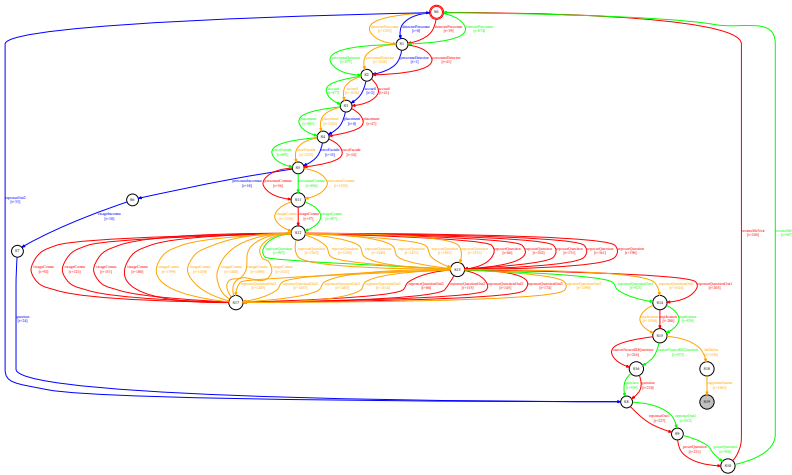


FIGURE 5.4. Graphe d'un joueur ayant eu des problèmes de reconnaissances vocales avec Nao

- Présentation du jeu && annonce de la première question
- Réponse à la première question && annonce de la deuxième question
- Réponse à la deuxième question && annonce de la troisième question
- Réponse à la troisième question && fin du jeu

Nous pouvons ainsi, a posteriori, reconstruire les automates temporisés de chaque joueur. Les problèmes liés à la reconnaissance vocale apparaissent alors par de nombreuses boucles (exemple figure 5.4).

5.1.3. Analyse

Victime de son succès, le robot Nao a suscité un grand intérêt, provoquant des cas d'utilisation non désirés. Le détail est donné dans le tableau 5.2.

Partie terminée	Problème de reco. vocale	Erreur d'identification	Abandon
18	13	3	25

TABLE 5.2. Vérité terrain de l'expérience réalisée au Muséum de La Rochelle

Si la première après midi, consacré à une classe de collégiens s'est bien déroulée (7 groupes de 2 ou 3 élèves), les phases de jeux ouvertes au grand public n'ont pas permis d'aboutir à une expérience de jeu optimale pour les joueurs. En effet, si les deux dernières après midi ont permis à 36 joueurs de venir découvrir le Muséum et jouer avec Nao, seulement 42 % d'entre eux ont pu terminer leur partie. Durant les différentes parties, nous pouvons remarqué grâce aux graphes de chaque joueur que beaucoup d'erreurs de reconnaissance vocale ont eu lieu. Ce nombre élevé a peut être conduit certains joueurs à abandonner en cours de partie. Deuxième explication, le nombre de joueurs simultanés qui a fait augmenter le temps d'attente pour jouer avec Nao. Certains parents n'ont pas voulu rester d'avantage de temps pour terminer la partie. Ce phénomène a été particulièrement noté lors des sessions du dimanche. Les erreurs de reconnaissance vocale sont dues, en partie, au bruit ambiant présent dans le musée et à la résonance de la pièce. Le seuil de reconnaissance était de 40 % pour les expériences de vendredi et samedi. Ce seuil a été abaissé à 30 % pour les sessions du dimanche, mais n'a pas permis d'augmenter significativement la qualité de la partie.

5.2. MUSÉE SAINTE CROIX DE POITIERS

La seconde expérimentation s'est déroulée au Musée Sainte Croix de Poitiers lors des *offs* du festival de la Gamers Assembly. Dans ce cadre, nous avons organisé une session de jeu permettant au public de venir découvrir le musée de manière ludique

Le jeu se déroule en quatre étapes, comme indiqué sur la figure 5.5.

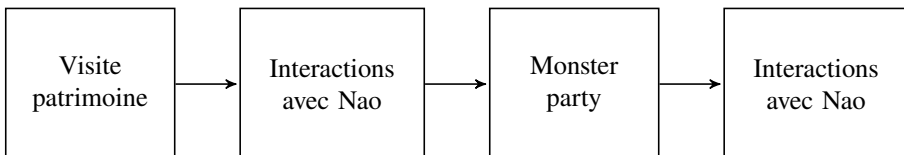


FIGURE 5.5. Étapes du jeu réalisé au musée Sainte Croix

5.2.1. *Visite patrimoine*

La première partie du jeu se déroule à l'extérieur du musée, en utilisant l'application *Visite patrimoine Poitiers*. Cette application, développée par des étudiants de La Rochelle Université, propose de découvrir certains monuments de la ville. Pour notre expérience, une version personnalisée de l'application a été développée par des étudiants de l'IUT de La Rochelle.

Les joueurs se sont vus proposer un parcours à l'extérieur du musée, leur faisant découvrir des monuments comme la cathédrale Saint Pierre, le Baptistère et l'extérieur du musée. À chaque étape du parcours, une énigme leur était posée dont la réponse leur servait d'indice. À l'issue du parcours extérieur, les joueurs devaient reformer un mot code à partir des différents indices récupérés et ainsi passer à la deuxième étape du jeu. Le mot code ainsi obtenu, les joueurs se dirigeaient vers le musée où Nao les attendait.

5.2.2. *Lancement du jeu : premières interactions avec Nao*

Nao était chargé d'accueillir les joueurs au musée. L'objectif de ces premières interactions étaient de vérifier le mot code des joueurs ainsi que de leur expliquer le déroulement de la seconde partie du jeu. La mécanique d'identification du joueur est identique à celle utilisée lors des expériences au Muséum. Lorsqu'aucun joueur n'est présent devant Nao, il se met en attente et s'assoit. La première action à effectuer est donc de lui appuyer sur la tête pour l'éveiller et lui permettre d'accueillir le visiteur. Nao demandait alors au joueur de présenter son badge. Chaque badge ayant un *NaoMark* unique, il était alors possible de déterminer quel joueur était en train d'interagir avec Nao et à quelle étape du jeu il en était.

Contrairement à l'expérimentation effectuée au Muséum de La Rochelle, et dans la mesure où un nombre trop important de joueurs pouvaient diminuer la qualité de l'expérience, une gestion des flux a été mise en place. En effet, dans le cadre de l'expérimentation du musée Sainte-Croix, une inscription préalable auprès des services de la mairie était nécessaire. Sur la base, de la liste des joueurs ainsi obtenue, nous avons donc ajouté une interaction, au cours de laquelle Nao demandait le prénom du joueur. Le dictionnaire des prénoms a permis une référence symbolique vers le prénom ayant le taux de confiance le plus élevé.

À l'issue de cette interaction, une sauvegarde du couple identifiant du badge/prénom était effectuée en mémoire de Nao. Cette information sera utilisable dans la suite du scénario.

Enfin, après cette sauvegarde, Nao vérifiait le mot code proposé par le joueur. Si le mot code était incorrect, le robot donnait un nouvel indice jusqu'à obtenir la bonne réponse. Lorsque la réponse était correcte, la seconde partie du jeu était expliquée au joueur. La *monster party* pouvait donc débiter.



FIGURE 5.6. Extrait du jeu *Monster party*

5.2.3. *Monster party*

Ce jeu sur tablette, développé par des étudiants de l'IUT, avait pour objectif de faire découvrir l'étage consacré à l'archéologie. Pour ce faire, le jeu proposait de collecter des armes pour capturer des monstres, associés à des œuvres. Grâce à la technologie de réalité augmentée, les joueurs devaient retrouver 3 armes et capturer 3 monstres (Figure 5.6). À chaque monstre capturé, un indice leur était proposé afin de déterminer le mot code final. À chaque événement produit dans le jeu (découverte d'un outil, capture d'un monstre ou récupération d'indice), une sauvegarde sur un serveur était réalisée.

Dans le cadre d'un projet de recherche sur l'analyse des déplacements des joueurs [10], chaque badge porté par les joueurs permettait d'enregistrer son parcours dans le musée, grâce à la technologie des iBeacons. Des Raspberry (mini ordinateur) ont été déployés sur l'étage où se déroulait ce jeu et sur lesquels les badges iBeacons se connectaient lorsqu'ils se trouvaient à proximité. Ainsi, un timestamp et l'identifiant du Raspberry étaient enregistrés sur un serveur afin d'identifier le parcours du joueur. À la fin de ce jeu, le joueur était invité à se rendre devant Nao pour vérifier le mot code final qu'il avait obtenu grâce aux différents indices récupérés.

5.2.4. *Fin du jeu : dernières interactions avec Nao*

Pour terminer la partie, le joueur devait retourner auprès de Nao pour lui donner le mot code final. Après s'être identifié avec leur badge, Nao récupérait sur le serveur l'ensemble des indices obtenus par le joueur, via son identifiant. Ces indices ainsi récupérés étaient redonnés au joueur. Celui-ci propose alors le mot code final au robot.

Si la réponse est incorrecte, le robot donne un indice supplémentaire jusqu'à ce que le joueur donne la bonne réponse. Enfin, lorsque la bonne réponse était proposée, Nao invitait les joueurs à se diriger vers l'accueil pour obtenir un lot.

5.3. RÉSULTATS

Pour palier les difficultés rencontrées lors de l’expérimentation effectuée au Mu-séum de La Rochelle, liées au flux important de joueurs, des inscriptions ont été mises en place par les services de la mairie de Poitiers. Ainsi, nous avons un contrôle du flux de joueurs voulant interagir avec Nao, qui pour cette occasion, était isolé du reste de l’exposition. Au total, nous avons comptabilisé 13 joueurs (ou équipes). Bien que le nombre de participants fût faible, cette expérimentation nous a permis d’étendre et de proposer un jeu sérieux plus abouti que celui proposé au Muséum de La Rochelle. En effet, dans cette expérience, nous avons couplé trois jeux, sur tablettes et sur le robot Nao. Contrairement à l’expérimentation de La Rochelle, aucun joueur n’a abandonné la partie. Des erreurs de reconnaissance vocale persistent cependant, mais leur nombre a été réduit par rapport à l’expérimentation de La Rochelle. Avoir isolé Nao et contrôlé le flux des joueurs a permis de limiter le bruit ambiant, améliorant ainsi la qualité de l’interaction. Un questionnaire évaluant la qualité du jeu a été proposé à chacun des joueurs à l’issue de leur partie. Les résultats sont présentés dans le tableau 5.3 et sur les figures 5.7, 5.8, 5.9.

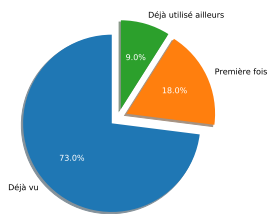


FIGURE 5.7. Connaissiez-vous Nao avant le début du jeu ?

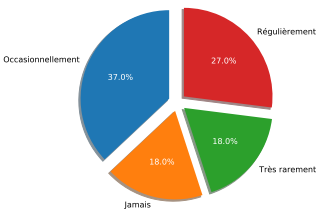


FIGURE 5.8. Étiez-vous déjà venu au musée Sainte Croix ?

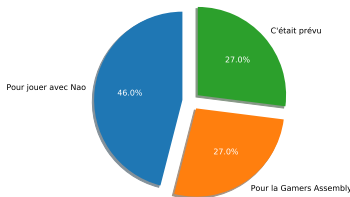


FIGURE 5.9. Pourquoi êtes-vous venu ?

Cette expérience a été qualifiée d’originale et de positive par une grande majorité des joueurs mais reste complexe à appréhender pour les plus jeunes, notamment avec le jeu en réalité augmentée sur tablette. Cependant, ces résultats montrent également

	Simple	Ennuyeuse	Originale	Positive
Tout à fait d'accord	9 %	0	82 %	64 %
D'accord	28 %	0	9	36 %
Pas d'accord	45 %	18	0	0
Pas du tout d'accord	0	82 %	0	0
Sans opinion	18 %	0	9 %	0
Total	100 %	100 %	100 %	100 %

TABLE 5.3. Comment qualifieriez-vous cette expérience ?

que les joueurs sont satisfaits de leur expérience au musée Sainte Croix. Plus du tiers des joueurs n'ont pas pour habitude de visiter des musées et près de la moitié sont venus pour jouer avec Nao. Bien que ces résultats ne portent que sur 13 joueurs ou équipes, ils démontrent que l'usage des nouvelles technologies peut inciter le jeune public à découvrir le patrimoine exposé dans les lieux de culture.

6. CONCLUSION

Nous avons présenté ici notre modèle de conception d'expériences interactives et la mécanique de supervision permettant de l'utiliser pour piloter le système. Des expériences publiques ont été menées qui nous ont permis de valider notre approche de modélisation et de supervision.

De plus, ces expérimentations nous ont prouvé à quel point la prise en compte du temps est importante pour obtenir une interaction de qualité. En effet, plus encore que dans les systèmes informatiques classiques, le paramétrage des temps de parole du robot et des délais au cours desquels il écoute une réponse verbalisée par le public est primordiale. Par exemple, nous avons constaté que la rapidité de réponse d'un utilisateur est la plupart du temps dépendante de son âge. Ainsi, un enfant va répondre très rapidement (parfois même avant que la séquence d'enregistrement du robot soit déclenchée), alors qu'une personne plus âgée prend davantage de temps avant de verbaliser sa réponse (et souvent alors que cette même séquence d'enregistrement est terminée). De plus, le choix des délais d'attente du robot avant de réagir à la présence d'un visiteur, sa vitesse de parole ou de mouvement doivent être paramétrés avec précaution car ces éléments ont un impact fort sur la perception du public [53]. C'est dans cette dynamique que nous avons intégré au framework une prise en compte des contraintes temporelles pour les comportements des agents et pour la gestion des passages entre contextes.

Au delà de cela, nous débutons les travaux d'adaptation en utilisant les traces utilisateurs pour déterminer par apprentissage par renforcement le paramétrage optimal de certains paramètres et ainsi mettre en place un raffinement dynamique du modèle via un bouclage de pertinence.

L'adaptation des comportements d'un système lors de son exécution est une tâche difficile et largement développée ces dernières années dans l'état de l'art [31, 39]. La difficulté principale est due à l'absence de méthodes précises pour la description et la modélisation des systèmes. Dans le domaine de la robotique, de nombreuses approches utilisent l'apprentissage par renforcement pour cette problématique et notamment pour les robots assistants et domestiques [60, 65].

L'objectif de l'apprentissage par renforcement est de trouver, par essais et erreurs, l'action optimale à effectuer. Chaque action du système est associée à une récompense (positive, négative ou nulle). Cette récompense permet d'évaluer la qualité d'une action particulière. L'objectif pour un agent est de maximiser la récompense accumulée au cours de l'exécution du système.

Nous nous intéressons particulièrement aux méthodes de différence temporelle [58] qui constituent les méthodes les plus utilisées en robotique. Elles combinent deux avantages, à savoir :

- apprendre à partir d'expériences sans connaissance a priori de l'environnement (méthodes de Monte-Carlo);
- utilisent les estimations des états successeurs pour estimer la valeur d'un état (programmation dynamique).

La principale méthode est celle du **Q-Learning** [69] et utilise le maximum de la fonction de valeur sur les actions suivantes et ne repose donc plus sur l'action effectivement réalisée. La fonction de valeur ne repose pas seulement sur l'état mais sur le couple état/action et est définie par :

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{a_t} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (6.1)$$

avec $0 \leq \alpha \leq 1$, le facteur d'apprentissage; $0 \leq \gamma \leq 1$, le facteur d'actualisation; s , l'état précédent; s_{t+1} , le nouvel état dans lequel le système se trouve après l'exécution de l'action a_t , associée à la récompense r_{a_t} .

Nous souhaitons donc utiliser cet algorithme sur les traces récoltées lors de l'expérimentation réalisée au Musée d'Histoire Naturelle de La Rochelle. L'intérêt d'utiliser ce jeu de données repose principalement sur le fait qu'il contient beaucoup d'exemples de cas d'erreurs susceptibles d'être évités par l'utilisation de notre technique de renforcement. Sur la base de ces logs, nous allons entraîner notre modèle pour déterminer les différents Q-valeur associées à chaque couple (localité, transition). Nous utiliserons dans un second temps la valuation obtenue pour chaque couple (localité, transition) pour adapter dynamiquement les paramètres contrôlant l'exécution du scénario.

BIBLIOGRAPHIE

- [1] R. ALUR & D. DILL, « The theory of timed automata », in *Real-Time: Theory in Practice: REX Workshop Mook, The Netherlands, June 3–7, 1991 Proceedings* (J. de Bakker, C. Huizing, W. de Roever & G. Rozenberg, eds.), Lecture Notes in Computer Science, vol. 600, Springer, Berlin, Heidelberg, 1992, p. 45-73.

- [2] M. ARAÚJO & L. ROQUE, « Modeling Games with Petri Nets », *Breaking New Ground: Innovation in Games, Play, Practice and Theory – Proceedings of DiGRA 2009* (2009).
- [3] E. ASPORD, J. BECKER & E. GRANGIER, *Link human/robot*, Van Dieren eds, 2014.
- [4] F. BANTERLE, F. A. CARDILLO, L. MALOMO, P. PINGI, F. GABELLONE & G. AMATO, « LecceAR: An Augmented Reality App », in *Digital Presentation and Preservation of Cultural and Scientific Heritage DiPP2015*, Institute of Mathematics and Informatics - Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Block 8 Sofia (1113), Bulgaria, September 2015, p. 99-108.
- [5] F. M. BARRETO, J. JESKE & S. JULIA, « A Timed Petri Net Model to Specify Scenarios of Video Games », in *Information Technology - New Generations*, Springer International Publishing, 2018, p. 467-473.
- [6] F. M. BARRETO & S. JULIA, « Modeling and Analysis of Video Games Based on Workflow Nets and State Graphs », in *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, CASCON '14, IBM Corp., Riverton, NJ, USA, 2014, p. 106-119.
- [7] F. BAUSE & P. KRITZINGER, *Stochastic Petri Nets – An Introduction to the Theory*, Vieweg, 2013.
- [8] P.-J. BENGHOZI, « Approche générationnelle des pratiques culturelles et médiatiques », *Culture Prospective* **3** (2007), p. 1-32.
- [9] J. BENGTTSSON & W. YI, « Timed Automata: Semantics, Algorithms and Tools », in *Lectures on Concurrency and Petri Nets*, Springer Berlin Heidelberg, 2004.
- [10] G. BERNARD, C. FAUCHER & K. BERTET, « Towards Reconstruction of Human Trajectories in Indoor Environments », in *Proceedings of the EKAW 2018 Posters and Demonstrations Session co-located with 21st International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018)*, Nancy, France, 2018, p. 37-40.
- [11] C. BOLOGNESI & D. AIELLO, « Learning through serious games a digital design museum for education », *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **XLIII-B5-2020** (2020), p. 83-90.
- [12] S. BOUCENNA, P. GAUSSIER, P. ANDRY & L. HAFEMEISTER, « A Robot Learns the Facial Expressions Recognition and Face/Non-face Discrimination Through an Imitation Game », *International Journal of Social Robotics* **6** (2014), n° 4, p. 633-652.
- [13] S. CAMPANO, C. CLAVEL & C. PELACHAUD, « “I like this painting too”: when an ECA shares appreciations to engage users », in *14th International Conference on Autonomous Agents and Multiagent Systems AAMAS'15* (Istanbul, Turkey), 2015.
- [14] M. CARROZZINO & M. BERGAMASCO, « Beyond virtual museums: Experiencing immersive virtual reality in real museums », *Journal of Cultural Heritage* **11** (2010), n° 4, p. 452-458.
- [15] R. CHAMPAGNAT, P. ESTRAILLIER & A. PRIGENT, « Adaptive Execution of Game: Unfolding a Correct Story », in *International Conference on Advances in Computer Entertainment Technology* (New York), ACE 06, ACM, 2006.
- [16] T. COENEN, L. MOSTMANS & K. NAESSENS, « MuseUs: Case Study of a Pervasive Cultural Heritage Serious Game », *J. Comput. Cult. Herit.* **6** (2013), n° 2, article no. 8 (19 pages).
- [17] H. COSTELHA & P. LIMA, « Robot task plan representation by Petri nets: modelling, identification, analysis and execution », *Autonomous Robots* **33** (2012), n° 4, p. 337-360.
- [18] G. W. DE OLIVEIRA, S. JULIA & L. M. SOARES PASSOS, « Game modeling using WorkFlow nets », in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 2011, p. 838-843.
- [19] R. DE VIRGILIO, « AML: A modeling language for designing adaptive web applications », *Personal and Ubiquitous Computing* **16** (2012), n° 5, p. 527-541.
- [20] S. DHOUBI, S. KCHIR, S. STINCKWICH, T. ZIADI & M. ZIANE, « RobotML, a Domain-Specific Language to Design, Simulate and Deploy Robotic Applications », in *Simulation, Modeling, and Programming for Autonomous Robots* (I. Noda, N. Ando, D. Brugali & J. J. Kuffner, eds.), Springer Berlin Heidelberg, 2012, p. 149-160.
- [21] D. DIETZE, *Playing and learning in early childhood education*, Wadsworth Publishing Company, 2011.
- [22] B. J. FREY & D. DUECK, « Clustering by Passing Messages Between Data Points », *Science* **315** (2007), n° 5814, p. 972-976.
- [23] T. GIBSON-ROBINSON, P. ARMSTRONG, A. BOULGAKOV & A. W. ROSCOE, « FDR3 – A Modern Refinement Checker for CSP », in *Tools and Algorithms for the Construction and Analysis of Systems* (E. Ábrahám & K. Havelund, eds.), Springer Berlin Heidelberg, 2014, p. 187-201.
- [24] J.-Y. GIRARD, « Linear logic », *Theoretical Computer Science* **50** (1987), n° 1, p. 1-101.

- [25] J. HAMARI, D. J. SHERNOFF, E. ROWE, B. COLLIER, J. ASBELL-CLARKE & T. EDWARDS, « Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning », *Computers in Human Behavior* **54** (2016), p. 170-179.
- [26] P. HENAFF, « Entre Art et Science : Guido, un robot guide espion au musée d'art moderne de Luxembourg », 2015, session vidéo, Journées Nationales de la Recherche en Robotique (JNRR).
- [27] K. JENSEN, « An introduction to the practical use of coloured Petri Nets », p. 237-292, Springer Berlin Heidelberg, 1998.
- [28] M. KHLAT, « Norio, the robot guide of the Oiron Castle », Publié sur le site de Tourmag le 11/12/2014, http://www.tourmag.com/Norio-the-robot-guide-of-the-Oiron-Castle_a71190.html.
- [29] K. KOBAYASHI, A. NAKATANI, H. TAKAHASHI & T. USHIO, « Motion planning for humanoid robots using timed Petri net and modular state net », in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 6, 2002.
- [30] D. KORTENKAMP, R. SIMMONS & D. BRUGALI, « Robotic Systems Architectures and Programming », in *Springer Handbook of Robotics*, Springer International Publishing, Cham, 2016, p. 283-306.
- [31] C. KRUPITZER, F. M. ROTH, S. VANSYCKEL, G. SCHIELE & C. BECKER, « A survey on engineering approaches for self-adaptive systems », *Pervasive and Mobile Computing* **17** (2015), p. 184 - 206, 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian.
- [32] M. LAPEYRE, P. ROUANET & P.-Y. OUDEYER, « Poppy: a New Bio-Inspired Humanoid Robot Platform for Biped Locomotion and Physical Human-Robot Interaction », in *Proceedings of the 6th International Symposium on Adaptive Motion in Animals and Machines (AMAM)* (Darmstadt, Germany), 2013.
- [33] K. G. LARSEN, P. PETTERSSON & W. YI, « Uppaal in a Nutshell », *International Journal on Software Tools for Technology Transfer* **1** (1997), p. 134-152.
- [34] A. LEHMANN, R. MIKUT & T. ASFOUR, « Petri Nets for Task Supervision in Humanoid Robots », in *Proc., 37th International Symposium on Robotics (ISR 2006)*, 2006, p. 71-73.
- [35] A. LIAPIS, H. P. MARTÍNEZ, J. TOGELIUS & G. N. YANNAKAKIS, « Adaptive game level creation through rank-based interactive evolution », in *IEEE Conference on Computational Intelligence in Games (CIG)*, 2013, p. 1-8.
- [36] Y. LOUCHART & R. AYLETT, « Emergent narrative, requirements and high-level architecture », in *Proceedings of the 3rd Hellenic Conference on Artificial Intelligence*, 2004, p. 298-308.
- [37] M. LUCKCUCK, M. FARRELL, L. A. DENNIS, C. DIXON & M. FISHER, « Formal Specification and Verification of Autonomous Robotic Systems: A Survey », <http://arxiv.org/abs/1807.00048>, 2018.
- [38] B. MAGERKO, « Building an interactive drama architecture », in *First International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003, p. 226-237.
- [39] G. S. MARTINS, L. SANTOS & J. DIAS, « User-Adaptive Interaction in Social Robots: A Survey Focusing on Non-physical Interaction », *International Journal of Social Robotics* **11** (2019), n° 1, p. 185-205.
- [40] C. MCGANN, K. RAJAN & A. GARCIA OLAYA, « Integrated Planning and Execution for Robotic Exploration », in *International Workshop on Hybrid Control of Autonomous Systems*, 2009.
- [41] T. MIYASHITA, P. MEIER, T. TACHIKAWA, S. ORLIC, T. EBLE, V. SCHOLZ, A. GAPEL, O. GERL, S. ARNAUDOV & S. LIEBERKNECHT, « An augmented reality museum guide », in *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2008, p. 103-106.
- [42] A. MIYAZAWA, P. RIBEIRO, W. LI, A. CAVALCANTI, J. TIMMIS & J. WOODCOCK, « RoboChart: a state-machine notation for modelling and verification of mobile and autonomous robots », *Tech. Rep.* (2016).
- [43] ———, « RoboChart: modelling and verification of the functional behaviour of robotic applications », *Software & Systems Modeling* **18** (2019), p. 3097-3149.
- [44] D. MONDOU, A. PRIGENT & A. REVEL, « A Dynamic Scenario by Remote Supervision: A Serious Game in the Museum with a Nao Robot », in *Advances in Computer Entertainment Technology* (A. D. Cheok, M. Inami & T. Romão, éd.), Springer International Publishing Cham, 2018, p. 103-116.
- [45] M. L. MORALES-RODRÍGUEZ, « Modèle d'interaction sociale pour des agents conversationnels animés : Application à la rééducation de patients cérébro-lésés », Thèse, Université Paul Sabatier – Toulouse III, 2007.
- [46] T. MURATA, « Petri nets: Properties, analysis and applications », *Proceedings of the IEEE* **77** (1989), n° 4, p. 541-580.

- [47] S. NATKIN & L. VEGA, « A Petri Net Model for the Analysis of The Ordering of Actions in Computer Games », in *GAME ON 2003* (France), Jan 2003, London, October 2003.
- [48] S. OCTOBRE, « Pratiques culturelles chez les jeunes et institutions de transmission : un choc de cultures ? », *Culture prospective* **1** (2009), n° 1, p. 1-8.
- [49] F. PECUNE, S. MURALI, V. TSAI, Y. MATSUYAMA & J. CASSELL, « A Model of Social Explanations for a Conversational Movie Recommendation System », in *Proceedings of the 7th International Conference on Human-Agent Interaction* (New York, NY, USA), HAI '19, Association for Computing Machinery, 2019, p. 135-143.
- [50] A. PRIGENT & A. REVEL, « CITE – Content Interaction Time and spacE: a hybrid approach to model man-robot interaction for deployment in museums », *EAI Endorsed Transactions on Creative Technologies* **4** (2017), n° 13.
- [51] N. REMPULSKI, « Synthèse dynamique de superviseur pour l'exécution adaptative d'applications interactives », Thèse, Université de La Rochelle, 2013.
- [52] N. REMPULSKI, A. PRIGENT, V. COURBOULAY, M. PERREIRA DA SILVA & P. ESTRAILLIER, « Adaptive Storytelling Based On Model-Checking Approaches », *International Journal of Intelligent Games & Simulation (IJIGS)* **5** (2009), n° 2, p. 33-42.
- [53] A. REVEL & P. ANDRY, « Emergence of structured interactions: from a theoretical model to pragmatic robotics », *Neural Networks* **22** (2009), n° 2, p. 116-125.
- [54] B. RIDEL, P. REUTER, J. LAVIOLE, N. MELLADO, N. COUTURE & X. GRANIER, « The Revealing Flashlight: Interactive Spatial Augmented Reality for Detail Exploration of Cultural Heritage Artifacts », *Journal on Computing and Cultural Heritage* **7** (2014), p. 1-18.
- [55] M. O. RIEDL & V. BULITKO, « Interactive Narrative: An Intelligent Systems Approach », *AI Magazine* **34** (2012), n° 1, p. 67-77.
- [56] I. P. SAMUELSSON & M. FLEER, *Play and learning in early childhood settings: international perspectives*, Springer Dordrecht ; London, 2008.
- [57] S. SCHNEIDER, *Concurrent and Real-time systems*, John Wiley and Sons, 2000.
- [58] R. S. SUTTON, « Learning to Predict by the Methods of Temporal Differences », *Machine Learning* **3** (1988), n° 1, p. 9-44.
- [59] C. SZYPERSKI, D. GRUNTZ & S. MURER, *Component Software: Beyond Object-Oriented Programming*, 2nd éd., ACM Press and Addison-Wesley, 2002.
- [60] A. TAPUS, C. ȚĂPUȘ & M. J. MATARIĆ, « User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy », *Intelligent Service Robotics* **1** (2008), n° 2, p. 169-183.
- [61] O. THÖRN, P. KNUDSEN & A. SAFFIOTTI, « 2020 Human-Robot Artistic Co-Creation: a Study in Improvised Robot Dance », in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, p. 845-850.
- [62] S. TORSI, C. ARDITO & C. REBEK, « An Interactive Narrative to Improve Cultural Heritage Experience in Elementary School Children », *Journal on Computing and Cultural Heritage* **13** (2020), n° 3, article no. 22 (14 pages).
- [63] W. VAN DER AALST & K. VAN HEE, *Workflow Management: Models, Methods, and Systems*, MIT Press, Cambridge, MA, USA, 2004.
- [64] N. VISWANADHAM & Y. NARAHARI, *Performance Modeling of Automated Manufacturing Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
- [65] K. WADA & T. SHIBATA, « Robot therapy in a care house - its sociopsychological and physiological effects on the residents », in *Proceedings 2006 IEEE International Conference on Robotics and Automation*, 2006, p. 3966-3971.
- [66] C. WANG, D. Z. WANG & J. L. LIN, « ADAM: An adaptive multimedia content description mechanism and its application in web-based learning », *Expert Systems with Applications* **37** (2010), n° 12, p. 8639-8649.
- [67] R. WANG, Y. GUAN, H. SONG, X. LI, X. LI, Z. SHI & X. SONG, « A Formal Model-Based Design Method for Robotic Systems », *IEEE Systems Journal* **13** (2018), n° 1, p. 1096-1107.
- [68] R. WANG, P. LUO, Y. GUAN, H. WEI, X. LI, J. ZHANG & X. SONG, « Timed automata based motion planning for a self-assembly robot system », in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, p. 5624-5629.

- [69] C. J. C. H. WATKINS & P. DAYAN, « Q-learning », *Machine Learning* **8** (1992), n° 3, p. 279-292.

ABSTRACT. — Many approaches to the design of robotic applications have been proposed in recent years. If they are relevant to the problem they solve, most of them are platform or language dependent. To solve this problem, the use of formal models for scenario or activity generation is essential to ensure a high quality experience.

In this context, the use of Petri nets or automata is very widespread. Based on this observation, the formal model CIT (Content Interaction Time) [44], dedicated to the development of interactive robotic applications and based on networks of input/output timed automata has been proposed.

In this paper, we propose two software platforms, CELTIC (Common Editor for Location Time Interaction and Content) and EDAIN (Execution Driver based on Artificial INtelligence), allowing a simplified and generic modeling of robotic applications and their supervision based on the CIT model.

This approach was used to design a serious game with a humanoid robot Nao. This game allows young people to discover in a playful way the exhibition dedicated to ethnography at the Natural History Museum of La Rochelle and the exhibition dedicated to archaeology at the Sainte Croix Museum in Poitiers.

After these experiments, problems of dynamic adaptation of the scenario were identified. We will thus present the beginning of our reflections on the integration of a reinforcement learning algorithm allowing to optimize the parameters controlling the scenario's execution.

KEYWORDS. — Formal Modeling, Automaton, Robotics, Human-Robot Interaction.

Manuscrit reçu le 15 mai 2021, révisé le 10 septembre 2021, accepté le 7 décembre 2021.