

## PILOTAGE D'UN JEU : DEFINITION D'UNE ARCHITECTURE DE COMMANDE

Ronan CHAMPAGNAT, Guylain DELMAS, Michel AUGERAUD

L3i – Université de La Rochelle

Pôle science et technologie

17 041 La Rochelle cedex 1

{ronan.champagnat, guylain.delmas, michel.augeraud}@univ-lr.fr

**RESUME :** *L'objectif de cet article est de définir une architecture logicielle permettant la gestion de l'exécution adaptative du déroulement d'une application interactive de type jeu. Cette architecture est définie par rapport aux fonctionnalités décrites dans le cadre de la commande des systèmes complexes événementiels. Elle fait apparaître deux niveaux hiérarchiques. Le niveau pilotage local qui consiste à utiliser la flexibilité du scénario pour proposer une histoire cohérente vis-à-vis du scénario défini a priori et en fonction des décisions du joueur. Et le niveau pilotage global chargé de choisir le meilleur scénario à exécuter en fonction de l'évolution du jeu.*

*Cet article commence par présenter l'architecture générale d'un jeu en soulignant les deux approches existantes (orientée scénario et émergente). Puis nous décrivons les spécificités des jeux qui viennent de la difficulté d'autoriser le joueur à laisser son empreinte sur le déroulement du jeu tout en ne mettant pas en péril la structure du récit définie au préalable.*

**MOTS-CLES :** *pilotage, architecture des systèmes, scénarios, exécution adaptative*

### 1 INTRODUCTION

L'informatique ubiquitaire est souvent décrite par la profusion de support et de données accessibles. Les systèmes d'information font apparaître un nouveau rapport à l'utilisation de l'informatique ainsi que de nouvelles interactions entre l'homme et la machine. En particulier, les interfaces sont multiples et permettent d'obtenir une mesure plus fine ou, du moins, une information diverse sur l'utilisateur du système.

L'environnement (qui est fortement informatique) est dit sensible et devient capable d'interagir et de s'adapter à l'utilisateur et au contexte d'utilisation. Cette évolution de la capacité sensorielle d'un système informatique couplée avec la capacité d'adaptation de l'application à son utilisateur constitue ce que nous définissons par Interactivité Numérique.

À titre d'exemple nous pouvons citer les environnements informatiques pour l'apprentissage humain (EIAH) dont l'objectif consiste à accompagner l'acquisition de connaissances et de savoirs faire. Dans ce cadre, une activité est définie par un pédagogue (ou un expert du domaine) et l'apprenant (ou l'utilisateur) interagit avec l'application. Nous considérons plus particulièrement le cas des micromondes de type simulation. C'est-à-dire que l'EIAH modélise un environnement complexe, dangereux ou cher et permet à l'apprenant d'être mis en situation (sans conséquences pour lui ou le système). Au cours de cette expérimentation l'apprenant va pouvoir

élaborer des stratégies, les essayer et, ainsi acquérir des connaissances de manière plus profonde.

Il est désormais possible, via des acquisitions vidéo, d'interpréter le comportement de l'apprenant et d'adapter le déroulement de l'activité d'apprentissage en fonction de l'analyse de la situation, des objectifs pédagogiques et des capacités du système.

Un autre exemple concerne les jeux vidéo. Les nouvelles consoles rentrent pleinement dans l'Interactivité Numérique en utilisant de plus en plus de périphériques nouveaux et en développant l'interprétation du comportement du joueur. Mais les éditeurs de jeux travaillent de plus en plus sur l'adaptation du jeu pendant le déroulement de la partie.

En effet, un joueur est face à un jeu qui, généralement, constitue un modèle d'un processus avec lequel il interagit. Dans le cadre des jeux d'exploration-aventure, le joueur se retrouve dans un univers inconnu qu'il doit explorer et comprendre. Le niveau étant généralement terminé lorsque le joueur a découvert les règles de celui-ci.

Dans le cadre des jeux, il est intéressant de souligner que les jeux, tout comme les EIAH, permettent au joueur de développer des stratégies et de les expérimenter. Contrairement au monde réel auquel le joueur est confronté, lors d'un jeu, il peut être mis en échec et élaborer d'autres stratégies. Une règle communément admise en Game Design est de considérer que le jeu est

un ensemble de problèmes à résoudre, les premiers étant facile puis les problèmes deviennent de plus en plus complexes (nécessitent une dextérité ou une connaissance de l'environnement du jeu plus grande) mais le joueur finit toujours par terminer le jeu (et donc résoudre le problème très complexe).

Cette gestion du déroulement du jeu, l'adaptation de l'application en fait, constitue un élément de plus en plus important dans la création du jeu et dans la mesure de sa qualité. Par conséquent il n'est pas rare que les nouveaux jeux s'annoncent à la fois libres et scénarisés. Ils ne sont généralement que l'un ou l'autre. Nous trouvons ainsi d'une part des jeux aux scénarios très développés, mais dans lesquels le joueur se trouve contraint de suivre l'histoire proposée, et d'autre part des jeux offrant une réelle liberté au joueur, mais qui peinent à fournir une histoire construite et intéressante.

La gestion de l'exécution adaptative est rendue complexe de par le fait que l'utilisateur dispose d'une grande influence sur le déroulement de l'application grâce à ses interactions avec l'application. Or ces applications sont spécifiées par un expert (game designer, pédagogue, thérapeute) qui définit un ou plusieurs déroulements en fonction de ses objectifs (compétences à acquérir, contenu à parcourir, scénario, courbe dramatique...).

L'objectif de cet article est de définir une architecture logicielle permettant la gestion de l'exécution adaptative du déroulement d'une application interactive de type jeu. Dans (Champagnat et al., 2005) nous avons posé les bases d'une architecture adaptative. L'adaptativité de l'exécution se fait au travers d'un moniteur (agent logiciel ou personne) qui intervient pendant le déroulement afin de s'assurer que l'utilisateur suit une exécution conforme.

Dans ce cadre, nous cherchons à maîtriser l'interaction pour permettre une évolution adaptative du déroulement de l'application en :

- fonction du comportement du joueur ;
- respectant les consignes du concepteur ;
- tenant compte de l'état des ressources.

Cet article n'aborde pas les aspects liés à l'écriture d'un scénario comme le font (Veissberg, 1999) ou (Propp, 1968), ni sur la structure d'un scénario (Juul 1999). Nous visons à mettre en place les mécanismes permettant d'obtenir un déroulement suivant une structure de récit établie au préalable.

À ce titre nous abordons le problème suivant un point de vu système et qui finalement est proche de la conduite d'un système flexible.

Nous commencerons par donner l'architecture générale d'un jeu. Nous montrerons que les deux approches existantes (orientées scénarios et émergentes) font appel

à une architecture similaire : un module interagit avec le modèle du jeu pour « contrôler » l'exécution du jeu.

Puis nous décrirons quels sont les besoins, en termes de fonctionnalité du contrôleur, pour le pilotage d'un jeu. Ces besoins viennent de la difficulté d'autoriser le joueur à laisser son empreinte sur le déroulement du jeu tout en ne mettant pas en péril la structure du récit définie au préalable.

Nous rapprocherons ces fonctionnalités de celles de la commande des systèmes flexibles, et donnerons les éléments pour une architecture de jeux à exécution adaptative. Nous détaillerons les aspects liés à la structure de commande et sa cohésion ainsi qu'au pilotage en temps-réel.

Mais tout d'abord, pour illustrer nos propos et les notions que nous manipulons, nous décrivons un exemple académique d'un jeu d'exploration-aventure basé sur un univers simple à découvrir.

## 2 EXEMPLE TYPIQUE DE JEU D'EXPLORATION/AVENTURE

Les jeux d'exploration-aventure, tels que « The Legend of Zelda » ou « The Hobbit », consistent à explorer un monde où l'on peut récolter des objets (comme récompense après avoir vaincu une difficulté) qui permettront de poursuivre dans l'aventure ou qui constituent des gratifications.

Dans cette section, nous présentons un jeu inspiré de ces mécanismes mais d'une complexité maîtrisable. Le joueur se trouve dans un monde composé de 9 pièces qu'il doit explorer pour trouver un trésor. Dans les pièces se trouvent des clefs que le joueur peut, en fait doit, prendre et qui lui serviront à ouvrir les portes lui permettant de passer d'une pièce à une autre.

Un personnage non joueur, un fantôme, évolue dans ce même monde. Il peut subtiliser des clefs pour entraver l'action du joueur ou, au contraire, perdre des clefs permettant d'offrir de nouvelles perspectives au joueur.

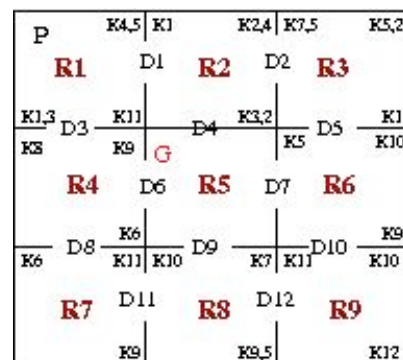


Figure 1. Topologie du jeu

De manière plus formelle, le joueur (P) doit explorer un monde composé de pièces (de R<sub>1</sub> à R<sub>9</sub> ). Pour cela, il

doit trouver les clefs ( $k_{1,2}...$ ) qui permettent d'ouvrir les portes (de  $D_1$  à  $D_{12}$ ) entre les pièces ( $K_{2,3}$  ouvrant à la fois  $D_2$  et  $D_3$ ). La partie est terminée lorsque le joueur a trouvé son objectif ( $G$ ).

Pour se déplacer d'une pièce à une autre le joueur doit franchir une porte. Pour cela il doit posséder la clef ouvrant la porte. Chaque clef peut ouvrir une porte donnée ou un ensemble de portes. Les clefs sont à usage unique, et la porte se referme après le passage du joueur. Une clef ne permet donc de passer qu'une seule fois à travers une porte.

### 3 ARCHITECTURE GENERALE D'UN JEU

Dans cette section, nous présentons les approches existantes dans le cadre du contrôle du déroulement d'histoire.

Mais avant d'introduire les travaux sur l'ontologie pour les jeux vidéo ainsi que les travaux liés à la construction d'histoire il nous semble nécessaire de faire un point sur le vocabulaire utilisé au cours de cet article.

En effet il est courant d'entendre parler de récit, de narration, d'histoire, de schéma narratif ou encore de scénario interactif. Nous ne considérerons que deux notions : l'histoire et le scénario.

L'histoire correspond, dans le cadre de cet article, à la suite chronologique des événements qui se déroulent au cours du jeu. Les événements pouvant être à l'initiative du joueur, par exemple ramasser une clef, ou à l'initiative du « jeu », par exemple le fantôme qui perd une clef.

Une histoire, dans le cadre de notre exemple académique, correspond à la suite d'événements suivants : « acquisition de la clef  $k_{1,3}$  par le joueur » ; « acquisition de la clef  $k_{4,5}$  par le fantôme » ; « franchissement de la porte  $D_1$  par le joueur » ; « acquisition de la clef  $k_{2,4}$  par le joueur » ; « franchissement de la porte  $D_2$  » ; « accomplissement de l'objectif par le joueur ». Cette histoire présuppose que le joueur se trouve dans la pièce  $R_1$ , ainsi que les clefs  $k_{1,3}$  et  $k_{4,5}$ , au commencement du jeu.

Le scénario, quant à lui, correspond à une description amont du jeu qui décrit la trame générale du déroulement du jeu et permet d'engendrer l'ensemble de toutes les histoires possibles (toutes les façons d'ordonner les événements du jeu).

La description d'un scénario est moins simple. Nous pouvons, par exemple, caractériser un scénario par l'état initial du jeu et un ensemble (partiellement ordonné ou non) des événements possibles.

### 3.1 Ontologie pour les jeux vidéo

Les travaux de (Zagal et al, 2005) dans le cadre du Game Ontology Project visent à définir un cadre pour décrire et analyser un jeu. Leur approche est basée sur l'étude de jeux existants dont ils ont extrait un ensemble de concepts et méthodes utilisés par les jeux.

L'ontologie proposée par les auteurs est structurée autour des notions d'interface, règles, objectifs, entités et entités jouables.

L'interface décrit l'ensemble des moyens que peut utiliser le joueur pour interagir avec le jeu. Et, ainsi, influencer sur le déroulement de l'histoire.

Les règles déterminent les interactions élémentaires du joueur avec le jeu. Elles permettent de réguler le déroulement de l'histoire. Il est important de remarquer que les auteurs font la différence entre les règles du jeu et les règles de l'univers. Les règles de l'univers définissent le modèle « physique » du jeu (par exemple la gravité), alors que les règles du jeu sont des contraintes rajoutées aux règles de l'univers.

Cette discrimination entre les deux types de règles permet, par conséquent, de faire la distinction entre les jeux dit « abstraits » et les jeux de « simulation ».

Les objectifs correspondent aux conditions que le joueur doit remplir pour gagner une partie (trouver le trésor).

Enfin les entités et les entités jouables couvrent tous les éléments du jeu en les séparant entre éléments passifs (entités) et éléments manipulables (entités jouables).

Cette description permet de donner les éléments du jeu, mais elle ne donne aucune information sur la façon dont l'histoire se construit. Or le pilote est chargé de réaliser cette construction. Les paragraphes suivants décrivent les approches de construction courantes dans le cadre des jeux de type exploration-aventure.

### 3.2 Construction d'histoire dans les jeux vidéos

La construction d'histoire dans les jeux vidéo est divisée en deux grandes familles : les jeux avec histoire incorporée et les jeux avec histoire émergente. La première famille concerne des jeux basés sur une histoire et un scénario. Le travail du pilote du jeu consiste à s'assurer que les actions du joueur vont faire que l'histoire reste dans le cadre du scénario défini. Pour cela, ce type de jeux donne souvent des libertés mineures au joueur ou propose des cheminements alternatifs ou, encore, multiplie les histoires connexes à la trame principale. Toutefois le joueur ne dispose que de peu de degrés de liberté et son influence sur la structure de la narration est nulle.

Les jeux avec histoire émergente sont construits à partir d'un postulat opposé : au lieu de se baser sur une histoire, le jeu propose un univers riche, dans lequel tous les acteurs (humains comme virtuels) peuvent évoluer de manière libre et autonome. L'histoire va alors émerger des actions simultanées des personnages (Delmas et al, 2006). Il est évident que ce type d'exécution n'est basé sur aucune structure narrative et la qualité qui en découle, en termes de cohérence et de pertinence, est fortement dépendante du joueur et ne peut être garantie.

### 3.2.1 Histoires incorporées

Ces approches sont basées sur une réflexion autour de la définition de la notion scénario et de sa représentation, ainsi que sur des méthodes assurant la cohésion du scénario malgré les choix du joueur qui peut parfois s'y opposer.

Le Liquid Narrative Group (Université de Caroline du Nord) travaille sur un projet baptisé Mimesis (Young et al, 2004) consistant à piloter un moteur de jeu 3D (Unreal Engine) par un système d'exécution adaptative. Ils définissent un scénario comme un ensemble d'actions réalisées par différents acteurs et agencées par des liens de causalité. Le système proposé par ce groupe estime les effets des actions du joueur, les compare au plan scénaristique prévu et, en cas d'incompatibilité, peut soit redéfinir partiellement le plan, soit empêcher l'action du joueur de suivre son cours (en l'interrompant ou en la faisant déboucher sur un résultat non prévu).

Michael Mateas de l'équipe du Oz Project a orienté ses recherches sur le segment du drama interactif (vision théâtrale du récit, centrée sur l'évolution des relations et des états émotionnels des personnages) (Mateas, 1999). Dans son logiciel baptisé « Façade », il propose un scénario représenté comme un ensemble d'éléments atomiques dotés d'un ordre partiel, dans lequel le joueur va suivre un chemin différent à chaque partie.

L'équipe de l'Université du Michigan reprend cette notion de drama interactif (Magerko et al, 2003), et propose une architecture (IDA – Interactive Drama Architecture) basée sur un scénario écrit par un auteur humain puis formalisé sous la forme d'un graphe d'états avec un ordre partiel (Magerko, 2005). Le système comporte un agent directeur qui pilote le jeu en surveillant le joueur et les conséquences de ses actions sur l'évolution du scénario. En cas de détection « d'erreur », il altère l'état du monde de façon à le faire à nouveau coïncider avec le scénario prévu. Le principal intérêt de leur démarche est la capacité d'anticiper les problèmes éventuels en établissant des prédictions sur le comportement du joueur.

Avec le projet PAPOUS, l'Intelligent Agent and Synthetic Characters Group (Université de Lisbonne) travaille à la conception d'un conteur virtuel. Dans (Silva et al, 2003), ils rajoutent à ce conteur la capacité d'adapter son histoire en fonction de la perception de l'intérêt du pu-

blic (de jeunes enfants) capturé à l'aide d'un système d'acquisition vidéo. Ils conçoivent un modèle d'histoire comme un ensemble d'éléments atomiques (StoryBits) hiérarchisés. Chaque atome décrit un morceau de l'histoire et est doté de propriétés reprenant les fonctions de Propp (Propp, 1968). Un récit est un parcours de l'arbre dans lequel pour chaque niveau le directeur choisit l'élément le plus approprié en établissant des corrélations entre les propriétés de l'élément et les désirs perçus dans l'assistance.

### 3.2.2 Histoires émergentes

Les approches émergentes ont pour objectif de briser le carcan des approches orientées scénario en proposant des modèles dans lesquels la liberté de l'utilisateur est totale. Dans ces approches, l'histoire n'est pas représentée directement, mais émerge des comportements simultanés des acteurs.

Constatant le manque de liberté inhérent aux scénarii conventionnels, le Center for Virtual Environment de l'Université de Salford a développé les bases de l'Emergent Narrative (Aylett, 1999) : ne plus chercher à représenter directement l'histoire, mais décrire des personnages autonomes et faire émerger l'histoire de leurs interactions. S'inspirant des pratiques du jeu de rôle, ils proposent un modèle basé sur 3 niveaux de description : le « game master level » qui définit le schéma global des interactions entre personnages, le « groupe level » qui définit des dynamiques à l'intérieur d'un groupe de personnages liés et le « character level » qui précise la description et les motivations de chaque personnage individuellement. Ils constatent qu'avec ce modèle, l'histoire générée n'est jamais celle prévue par le concepteur, signe de l'impact réel du joueur dans l'histoire.

L'équipe d'Interactive Storytelling de l'Université de Teeside propose elle un système de narration interactive par le biais d'agents autonomes dirigés par des réseaux de tâches hiérarchisées (Charles et al, 2003). Ce modèle permet de mettre en scène des sitcoms employant des agents autonomes et est complété par un système de planification des tâches qui assure un comportement cohérent des acteurs dans la durée. Ils introduisent également une possibilité d'interaction par l'utilisateur qui peut agir sur l'environnement et adresser des consignes vocales aux acteurs virtuels. Récemment, le système a été complété par un agent de gestion intelligente des interactions acteur-environnement qui détermine les réactions des « objets » manipulés par les acteurs en fonction des besoins de la narration (Lugrin et Cavazza, 2006).

L'équipe d'ID Tension, quant à elle, part du constat qu'une expérience narrative ne peut se résumer à la simple juxtaposition de personnages autonomes, mais doit répondre à des objectifs narratifs définis (Szilas, 2003). Elle propose donc un modèle dans lequel les actions des personnages sont guidées non seulement par leur comportement, mais aussi selon l'intérêt de chaque possibili-

té pour la narration en cours. Ainsi, à chaque étape importante de l'histoire, le système détermine le comportement de chaque acteur pour produire des effets narratifs.

### 3.3 Bilan des architectures existantes

Le tour d'horizon des approches de construction d'histoire dans les jeux vidéo nous montre que le pilotage d'un jeu s'appuie sur une entité contrôleur (un module ou un agent) rajoutée autour du modèle du jeu (défini par les règles de l'univers et les règles du jeu).

Toutes ces approches ont été développées de manière ad hoc en réponse à des problèmes spécifiques. L'architecture d'un pilote de jeu n'a fait l'objet d'aucune étude précise (contrairement au jeu pour lesquels une ontologie a été proposée).

### 3.4 Première version de l'architecture proposée

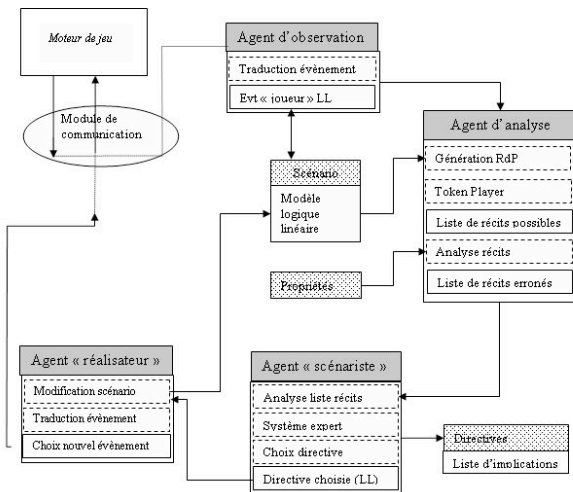


Figure 2. Première proposition d'architecture

Dans (Prigent et al., 2006) nous avons proposé une architecture pour l'adaptation de trames narratives (figure 2). Cette architecture est basée sur un modèle en Logique Linéaire du scénario qui est traduit en réseaux de Petri afin d'obtenir un modèle de commande qui est utilisé en émulation pour tester les décisions prises par un agent « réalisateur ».

Le point clef de cette architecture est l'articulation entre le modèle du scénario en Logique Linéaire et sa traduction en réseau de Petri. En effet, cela permet de modifier au cours de l'exécution la structure du réseau de Petri de commande. L'écueil principal étant qu'il est difficile de garantir la stabilité du système de commande. Pour corriger ce problème il faut mettre en place une structure hiérarchique de commande avec un niveau de pilotage qui possède une vision globale sur le déroulement du jeu.

Les expérimentations menées nous ont fait apparaître le besoin d'avoir un pilotage robuste. En effet, si le sorcier ne donne les clés pour passer au niveau suivant qu'à

condition que le joueur ait tué le dragon et que le joueur commence par tuer le dragon sans que le sorcier lui ait transmis la quête, le joueur doit pouvoir obtenir ces clés en récompense d'une autre quête. Il faut donc disposer d'un modèle exprimant la flexibilité du scénario et mettre en place des contrôleurs dont le but est de garantir la faisabilité du scénario (en terme de disponibilité des ressources)

Enfin le couplage entre le moteur de jeu et le système de commande apparaît comme un élément sensible de l'architecture car le niveau d'abstraction des entités manipulées n'est pas le même. Au niveau du moteur du jeu, il faut considérer toutes les actions du joueur (comme, par exemple, les coups portés lors d'un combat) alors qu'au niveau du pilotage seule la finalité du combat importe.

## 4 BESOINS POUR LE PILOTAGE D'UN JEU

Les architectures actuelles font clairement apparaître deux niveaux hiérarchiques. Le premier est en charge de réagir immédiatement aux actions (ou inactions) du joueur, le second possède une vision plus stratégique sur le déroulement de l'histoire (en s'assurant que le joueur a respecté des points de passage par exemple).

Nous voyons donc apparaître deux niveaux hiérarchiques de commande. Le premier, en prise directe avec le déroulement de l'histoire, est chargé suivre l'exécution du jeu et de proposer des solutions (ou des orientations) en réactions aux actions du jeu. Le second travail sur un horizon de temps plus long (à moyen terme) et gère l'évolution entre les objectifs prévisionnels et l'état d'avancement effectif du déroulement du jeu.

Plusieurs questions apparaissent. Comment garantir la cohérence entre les deux niveaux ? Quels mécanismes faut-il mettre en place pour assurer le suivi de l'exécution pour réaliser la prise de décision.

Ces questions sont classiques dans le cadre de la supervision de systèmes complexes. Nous les rapprochons plus particulièrement au cas des systèmes de production flexibles.

Dans le cadre du suivi de l'exécution d'un jeu nous devons garantir une cohérence de l'exécution vis-à-vis des choix antérieurs (cohérence locale) et des choix *a priori* du game designer (cohérence globale). Par exemple si le joueur, situé dans la pièce  $R_2$ , possède la clé  $K_{4,5}$ , il est impossible de lui empêcher de franchir la porte  $D_4$ . Alors que le game design du jeu avait fixé comme objectif de passer par la pièce  $R_6$  avant d'entrer dans  $R_5$ .

Ce problème est central dans le cas des applications interactives. Il s'agit de rendre actif l'utilisateur en lui permettant de laisser une trace dans le « réseau narratif » tout en évitant des typologies réductrices [VEI 99]. Et

c'est là la principale différence entre un système complexe flexible et un jeu.

Gérer la flexibilité d'un scénario devient un problème d'une grande complexité du fait de la prise en compte du joueur. Il est nécessaire de mettre en place des mécanismes garantissant une terminaison acceptable pour toutes les histoires. Mais également de prévoir une robustesse du scénario vis-à-vis du joueur dont une particularité est d'essayer de faire des choses impossibles ou non recommandées.

Dans (Delmas et al, 2007) nous proposons une gestion des histoires mêlant à la fois les concepts de liberté du joueur, comme pour les approches à histoires émergentes, en les complétant par un système de pilotage de l'exécution du scénario.

Ce pilotage a pour fonction d'observer le déroulement du jeu, d'analyser les actions du joueur et d'y apporter des réponses appropriées pour faire progresser le récit. Ce système ne se base pas sur un schéma d'exécution précis, mais sur un modèle générique de narration, dont il se sert comme référence pour générer son récit. Afin de ne pas entraver la liberté du joueur, les actions du système ne peuvent pas directement contredire ou invalider celles du joueur. Le système doit donc composer avec un champ d'action limité.

## 5 ELEMENTS POUR L'ARCHITECTURE DE COMMANDE D'UN JEU

Dans cette section, nous présentons les fonctionnalités mises en évidence pour la commande de système complexe, puis nous listons les fonctionnalités qui doivent figurer pour le pilotage du déroulement d'un jeu.

Commençons cette présentation par un exemple de modèle générique de narration : le « Périple du Héros » de Joseph Campbell.

### 5.1 Exemple de modèle générique pour la narration

J. Campbell (Campbell, 1968) a défini un modèle générique d'histoires. Ce modèle, basé sur l'étude des contes initiatiques en mythologie, retrace le périple d'un personnage tiré de son environnement familial et entraîné dans un univers étranger, le plus souvent fantastique. Il devra apprendre à maîtriser cet univers pour triompher d'épreuves qui jalonnent sa quête, puis retourner dans son monde, renforcé par la connaissance ainsi acquise.

Ce schéma (découverte d'un environnement inconnu, apprentissage, maîtrise, réutilisation dans un cadre familial) est particulièrement intéressant pour toute application visant à l'acquisition de nouvelles connaissances et compétences. De plus, la forte utilisation de cette structure en littérature et dans le cinéma lui confère un écho particulier auprès des joueurs.

La structure décrit l'histoire en trois parties :

- Le Départ : Le Héros commence l'histoire, plongé dans *le monde ordinaire*. Cela permet de présenter les problèmes propres au Héros (difficultés familiales, handicap...) qu'il devra surmonter lors de sa quête. Un événement (ou une série d'événements) le pousse à quitter le confort de son monde pour celui plus merveilleux qui s'offre à lui. Une fois que le Héros s'est engagé dans l'aventure, il reçoit une *Aide Surnaturelle*, sous la forme d'un allié ou d'un mentor qui lui délivre indices ou objets qui lui serviront par la suite. Il peut alors *Franchir le Seuil*, marquant son entrée dans le monde surnaturel, et se retrouver *Dans le Ventre de la Baleine*.
- L'Initiation : Notre Héros est désormais entré de plein pied dans l'aventure. Il va devoir passer une *Série d'Épreuves* qui viseront à tester son courage, sa détermination, et les vertus mises en avant dans sa quête. Ces épreuves se terminent par l'épreuve de l'initiation qui demandera d'employer toute sa maîtrise de la quête et lui offrira *l'Avantage Final* : la connaissance et l'expérience en récompense de ses efforts.
- Le Retour : Le Retour fait écho au Départ et clôt le récit. Le Héros, ayant désormais maîtrisé la quête, doit alors revenir la mettre à profit dans son monde.

Cette structure est utilisée dans le cadre du cinéma. (Delmas et al, 2007) en propose une adaptation dans le cadre des jeux. Il s'agit d'une simplification où l'on impose un déroulement de jeu de sorte que l'utilisateur ne tombe pas trop rapidement, ni trop lentement, sur la solution. Nous pouvons également imposer des points de passages (occurrence d'un événement particulier ou d'un nombre d'occurrence).

Par exemple dans le cadre du jeu d'exploration-aventure, nous pouvons observer les trois temps suivants :

- Le départ : positionné dans une pièce avec des clefs, le joueur doit comprendre qu'il doit se saisir des clefs pour franchir les portes.
- L'initiation : ramasser et franchir un ensemble de portes.
- Le retour : ouvrir le coffre qui contient sa récompense (une animation, ou le droit de passer au niveau suivant par exemple).

La phase d'initiation étant celle où l'on peut imposer un parcours particulier. Par exemple passer par la pièce  $R_6$  avant  $R_5$  et ne jamais se rendre dans la pièce  $R_8$ .

### 5.2 Fonctionnalités d'une commande de systèmes complexes

Les systèmes de production font apparaître la notion de flexibilité afin de répondre aux problèmes de variation

de la quantité demandé et de permettre au système de s'adapter aux perturbations (pannes, modifications des ressources).

Face à la complexité pour déterminer la commande d'un tel système et prendre en compte les différentes contraintes (niveau électrique d'un capteur et choix de la quantité de produit à fabriquer sur le long terme), le processus de commande a été divisé en une structure hiérarchique. Le but étant de ramener la résolution d'un problème de grande dimension aux caractéristiques variées à une suite de résolutions de problèmes de taille raisonnable ayant des caractéristiques plus homogènes.

Nous allons présenter les différentes fonctionnalités dans l'ordre croissant de la structure de commande en nous focalisant sur les niveaux supervision et ordonnancement.

Avant de présenter le niveau ordonnancement, nous allons décrire un mécanisme utile à notre architecture : le générateur d'évènements. Dans un contexte de surveillance de procédé et d'utilisation d'un réseau de terrain, il s'agit d'un mécanisme permettant d'extraire les évènements comme le dépassement d'une valeur particulière des mesures de capteurs cycliques. Ce générateur d'évènement est situé entre le réseau de terrain (synchrone) qui relaie les commandes et mesures du procédé aux contrôleurs locaux et le réseau de supervision (asynchrone) qui diagnostique et émet des informations de configuration au système de commande locale.

### 5.2.1 Niveau supervision

La fonction principale du niveau supervision consiste à la prise de décisions en temps-réel. Le système doit respecter au mieux l'ordonnancement prévisionnel et doit être capable en cas d'anomalie de faire une redistribution des tâches. Les trois grandes fonctions sont : pilotage en temps-réel des flux, pilotage en temps-réel des ressources et sécurité (Valette, 1998).

Le *pilotage des flux* consiste à définir ce qu'il est souhaitable de faire. À partir de la surveillance du procédé (donc de l'état courant) et de la politique prévisionnelle on redéfinit ce qu'il est souhaitable que le système fasse.

Cette fonction consiste, dans les étapes amont, à analyser l'écart (détection et diagnostique) entre l'ensemble des contraintes prises en compte au niveau prévisionnel et celles qui apparaissent en temps-réel. Puis il faut élaborer les politiques de décision en temps-réel (utilisation de la flexibilité, c'est-à-dire de l'indéterminisme résiduel pour permettre de réagir aux imprévus, ou remise en cause de la politique prévisionnelle).

Le *pilotage en temps-réel des ressources* cherche à établir ce que le système peut encore faire. Cette fonction consiste à définir, en fonction de l'état courant du procédé, ce que l'on peut (encore) faire.

Cela nécessite une analyse des défaillances et de leurs effets (en particulier de leur propagation), leurs criticités, la probabilité d'occurrence et la probabilité de non-détection. Mais, également, une étude des divers modes de fonctionnement et les diverses façons de reprendre un fonctionnement nominal après une défaillance.

La détection est pratiquement toujours fondée sur une comparaison en temps-réel du comportement effectif du système surveillé avec celui d'un modèle de bon comportement.

La *sécurité* vise à définir ce que le système ne doit pas faire. Les problèmes sont proches de ceux posés dans le cas précédent mais le point de vue est différent : « il s'agit de s'assurer que le système ne passe jamais dans une configuration anormale ».

### 5.2.2 Niveau ordonnancement

La prise de décision en fonction de l'état courant du système correspond à la fonction pilotage en temps-réel. Cette fonction utilise des algorithmes d'ordonnancement temps-réel qui sont généralement basés sur l'application d'un ensemble de règles qui permettent de déterminer un indice de priorité pour chaque tâche. Le rôle du pilote temps-réel est alors de choisir parmi les tâches prêtes à s'exécuter celle dont la priorité est la plus importante.

Une autre stratégie consiste à ne proposer au décideur, à chaque fois qu'il a une décision à prendre, qu'un choix compatible avec les contraintes modélisées et à satisfaire. Il peut s'agir soit de calculer à partir d'un état donné l'ensemble des décisions correctes possibles soit de déterminer *a priori* une stratégie ne permettant de ne générer que des décisions satisfaisant les contraintes.

Cependant il est important de noter que la caractérisation d'un ensemble d'ordonnancement admissible est un problème difficile. Deux démarches sont couramment utilisées. La première démarche consiste à rechercher des conditions nécessaires d'admissibilité, que tout ordonnancement solution du problème doit satisfaire. La deuxième démarche consiste à rechercher des conditions suffisantes d'admissibilités telles que tout ordonnancement satisfaisant ces conditions est admissible.

Il est à noter que nous avons choisi de classer la fonction de pilotage en temps-réel dans le niveau ordonnancement, mais de nombreux auteurs la considèrent comme faisant partie de la fonction pilotage en temps-réel des flux.

L'ordonnancement consiste à programmer dans le temps l'exécution d'opérations de manière à réaliser un objectif donné. Bien souvent le problème est découpé en deux étapes (ordonnancement sans contrainte de ressources et affectation) qui peuvent être enchaînées dans un sens ou dans l'autre.

La détermination d'un plan de production en fonction de contraintes liés à la quantité de produit à fournir et fonction des capacités est un problème complexe, et bien souvent la résolution est coûteuse en temps de calcul, pour lequel on trouve une littérature fournie. Nous pouvons citer les approches suivantes de résolution des problèmes d'ordonnement : les approches par construction progressive qui proposent à chaque itération une solution partielle au problème d'ordonnement (méthodes sérielles...); les approches par voisinage qui permettent de générer à chaque itération, à partir d'une solution initiale une ou plusieurs solutions (tabou, recuit simulé...); les approches par décomposition qui consistent à décomposer le problème initial en sous problèmes plus simples faiblement dépendants (hiérarchique, spatiale, temporelle...), les approches par modification de contraintes qui construisent des solutions non réalisables permettant une évaluation par défaut du problème utilisables par des techniques de constructions progressive (méthodes de relaxation); les approches basées sur les techniques de représentation des connaissances et de résolution de problèmes issues de l'Intelligence Artificielle. Toutes ces méthodes visent à déterminer une solution optimisée vis-à-vis d'un critère en déterminant l'ordre de passage des produits sur les machines.

En plus de ces travaux sur la recherche d'un ordonnancement nous pouvons relever les travaux concernant le processus décisionnel. Ces travaux visent à étudier l'exécution d'un ordre au travers des différentes couches de la structure hiérarchique de commande et à étudier la remontée de l'information aux travers des différentes couches. Il s'agit d'étudier l'agrégation et la désagrégation des décisions en s'assurant de la robustesse et de la cohérence des décisions.

Une décision est robuste si elle permet au niveau inférieur d'élaborer une décision détaillée. Alors qu'une décision détaillée sera dite cohérente si elle est compatible avec les décisions agrégées.

### **5.3 Fonctionnalités pour le pilotage de l'exécution d'un jeu**

La section précédente nous donne les fonctionnalités d'un système hiérarchique de commande, nous allons reprendre les fonctionnalités en les appliquant au cadre des jeux puis nous terminerons cette section par la proposition d'une architecture pour le pilotage.

Nous faisons apparaître deux niveau de commande : le niveau pilotage local et le niveau pilotage global. Le niveau de pilotage local étant interfacé avec le moteur du jeu (qui suit les règles du jeu) via un générateur d'évènements pour les informations montantes et directement pour les commandes envoyées au moteur du jeu.

#### **5.3.1 Niveau pilotage local**

Ce niveau consiste à utiliser la flexibilité du scénario pour proposer une histoire cohérente vis-à-vis du scénario défini *a priori* et en fonction des décisions du joueur.

Tout d'abord, le pilotage en temps-réel d'applications adaptatives consiste à surveiller et réagir vis-à-vis du scénario prévu. La principale question est de définir en fonction de l'état courant, quel est l'événement souhaitable.

Il faut analyser l'écart entre l'ensemble des contraintes prises en compte dans le scénario et celles qui peuvent se produire lors de l'exécution (choix de l'utilisateur, état des personnages et élément du jeu, état des ressources systèmes). Il faut donc évaluer les besoins de flexibilité et de réactivité, en amont, puis il faut élaborer des politiques de décision en temps-réel.

Le pilotage des ressources correspond aux activités de surveillance et de réactivité vis-à-vis des personnages, des éléments du jeu (voire des ressources système) et de leurs défaillances. On recherche, en fonction de l'état courant de l'histoire, les actions que l'on peut encore faire.

Enfin, la fonction de sécurité qui cherche à empêcher l'histoire d'atteindre certains états et, dans une situation anormale, doit forcer l'exécution de l'histoire tout en vérifiant un ensemble de contraintes très strictes.

La criticité du système en termes de risque pour l'utilisateur n'est pas grande, par contre la robustesse en termes d'utilisation doit être grande car la confiance de l'utilisateur envers l'application est fragile d'autant plus que cette application est utilisée dans un contexte autonome. Il s'agit donc d'un problème qui ne doit pas être négligé dans cette industrie.

#### **5.3.2 Niveau pilotage global**

Ce niveau gère l'exécution et l'ordonnement des actions de l'histoire. Les décisions qui sont prises visent à appliquer un scénario en fonction d'objectifs de gestion de l'intensité dramatique ou de progression pédagogique.

#### **5.3.3 Proposition d'architecture**

La figure 3 présente l'architecture que nous avons définie pour des jeux interactifs à exécution adaptative. Cette architecture est structurée en deux niveaux hiérarchiques (pilotage local et pilotage global).

Le pilotage local observe à la fois les informations en provenance du jeu (via le générateur d'évènements) mais

également les informations en provenance du pilotage global.

Trois modules concernent l'analyse de l'état de l'exécution et proposent une action à réaliser chacun en fonction de son point de vue (fonctionnement nominal, fonctionnement dégradé et fonctionnement sûr). Chaque module dispose d'un modèle du jeu et effectue une détection et un diagnostic des divergences constatées entre l'exécution et le modèle du jeu. Le module de pilotage des flux possédant en plus un modèle exprimant la flexibilité du scénario. Le module décision choisit au final l'action à réaliser.

Le pilotage global choisit le scénario à réaliser en fonction des informations qu'il observe et des scénarios pré-établis. Il peut, également, être conduit à choisir un autre scénario si le pilotage local lui impose une remise en cause du scénario.

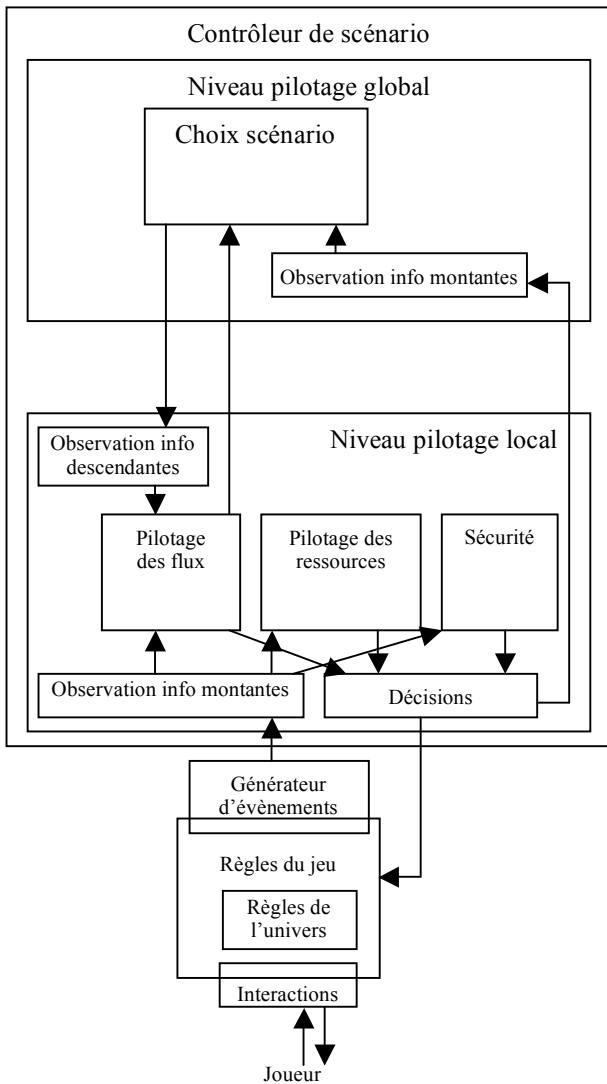


Figure 3. architecture pour un jeu à exécution adaptative

## 6 CONCLUSION

Dans cet article, nous avons présenté la problématique du pilotage du déroulement d'un jeu que nous avons rapproché de celle du pilotage de systèmes complexes événementiels.

Nous avons proposé une architecture pour le pilotage d'histoires dans le cadre des jeux vidéo. Cette architecture est en cour de développement dans le cadre du pilotage de l'histoire d'un jeu éducatif, et fait suite à une version précédente de l'architecture développée dans le cadre d'un jeu académique.

Les perspectives de ces travaux portent à la fois sur la définition d'une architecture générique pour la gestion du déroulement de l'exécution pour les applications interactives à exécution adaptatives et à la définition de l'ontologie pour le pilotage de telles applications.

## REFERENCES

- Alexiev V. Application of Linear Logic to Computation: an overview. *Bull. Of the IGPL*, vol. 2, n°1, 1994, p. 77–107
- Andreu D. Commande et supervision des procédés discontinus : une approche hybride. *Thèse de doctorat de l'Université de Toulouse III*, novembre 1996.
- Artigues C. Ordonnancement en temps-réel d'ateliers avec temps de préparation des ressources. *Thèse de doctorat*, Université de Toulouse III, décembre 1997
- Aylett R. Narrative in virtual environments – towards emergent narrative, *proceedings of the AAAI Fall Symposium on Narrative Intelligence*, 1999
- Billaut J.-C. Prise en compte de ressources multiples et des temps de préparation dans les problèmes d'ordonnancement en temps-réel. *Thèse de doctorat*, Université de Toulouse III, décembre 1993
- De Bonneval A. Mécanismes de reprise dans les systèmes de commande à événements discrets. *Thèse de doctorat*, Université de Toulouse III, 1993
- Campbell J. A hero with a thousand faces, *Princeton University Press*, 1972
- Cavazza M., Pizzi D., Narratology for interactive storytelling, a critical introduction, *4<sup>th</sup> International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt (Allemagne), 2006
- Carlier J. Chretienne P. Problèmes d'ordonnancement : modélisation-complexité-algorithmes. Masson, 1988
- Chaillet-Subias A. Approche multi modèles pour la commande et la surveillance en temps-réel des systèmes à événements discrets. *Thèse de doctorat*, Université de Toulouse III, décembre 1995
- Champagnat R. Optimisation d'une séquence de franchissement de transitions dans un réseau de Petri t-temporisé. *MOSIM'03*, Toulouse (France), avril 2003, p. 94– 100

- Champagnat R. Prigent A. Estraillier P. Scenario building based on formal methods and adaptative execution. *ISAGA*, Atlanta (USA), juin 2005
- Charles F. Lozano M. Mead J. Bisquerra F. Cavazza M. Planning formalisms and authoring in interactive storytelling, *1<sup>st</sup> International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt (Allemagne), 2003
- Collé F. Champagnat R. Prigent A. Scenario analysis based on Linear Logic. *ACE*, Valence (Espagne), juillet 2005
- Delmas G. Champagnat R. Augeraud M. Plot monitoring for interactive narrative games, *international conference on Advances in Computer Entertainment*, Salzburg (Autriche), 2007
- Deschamps J. Gestion hiérarchisée de cellules flexibles d'assemblage : concepts, modèles et simulation. *Thèse de doctorat*, Université de Toulouse III, 1994
- Dormans J. The hacker : new mythical content of narrative games, *4<sup>th</sup> International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt (Allemagne), 2006
- Eladhari M. Object oriented story construction in story driven computer games, *master thesis*, université de Stockholm (Suède), 2002
- Erschler J. Fontan G. Mercé C. Approche par contrainte en planification et ordonnancement de la production. *RAIRO APII*, vol. 6, no 27, 1993, p. 669–693
- Girard J.-Y. Linear Logic. *Theoretical Computer Science*, vol. 50, 1987, p. 1–102
- Girault F. Formalisation en logique linéaire du fonctionnement des réseaux de Petri. *Thèse de doctorat*, Université de Toulouse III, décembre 1997
- GOTHA. Les problèmes d'ordonnancement. *Recherche Opérationnelle*, vol. 1, 1993, p. 77–130
- Jenkins H. Game design as narrative architecture. *First person: new media as story, performance and game*, Cambridge, 2003
- Juul J. A clash between game and narrative, 1999
- Küngas P. Using Linear Logic planning to make knowledge bases reactive. *Seventh Symposium on Programming Languages and Software Tools*, Szeged (Hongrie), juin 2001, p. 135–148
- Küngas P. Linear Logic programming for AI planning, mai 2002
- Lugrin J.-L. Cavazza M. AI-based world behaviour for emergent narratives, *international conference on Advances in Computer Entertainment Technology*, Hollywood (Etats-Unis), 2006
- Magerko B. Story representation and interactive drama, *1<sup>st</sup> Artificial Intelligence and Interactive Digital Entertainment Conference*, Los Angeles (Etats-Unis), 2005
- Magerko B. Laird J. Building an interactive drama architecture, *1<sup>st</sup> International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt (Allemagne), 2003
- Mateas M. Interactive drama on computer : beyond linear narrative, *AAAI Fall Symposium on Narrative Intelligence*, 1999
- Niel E. Rezg N. Dioso J. Fravel J. Système de commande-supervision orienté sécurité opérationnelle. *APII*, vol. 5, no 28, 1994, p. 539–549
- Pradin-Chezalviel B. Valette R. Künzle L.-A. Scenario durations characterization of t-timed Petri nets using linear logic. *PNPM'PP*, Zaragoza (Espagne), septembre 1999
- Pradin-Chezalviel B. Valette R. Accessibilité marquage et logique linéaire dans un réseau de Petri t-temporel. *Journées Formalisation des Activités Concurrentes*, Toulouse (France), mai 2000, CERT-IRIT-LAAS/CNRS, p. 123–134
- Prigent A. Champagnat R. Estraillier P. Exécution adaptative de trame narrative. *Chapitre 9 du livre IA et jeux*, Hermès sciences, 2006
- Propp V. Morphology of the folktale *2<sup>nd</sup> edition university of Texas press*, 1968
- Silva A. Raimundo G. Paiva A. Tell me that bit again... Bringing interactivity to a virtual storyteller, *international conference on virtual storytelling*, 2003
- Szilas N. IDtension : a narrative engine for interactive drama, *1<sup>st</sup> International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt (Allemagne), 2003
- Tammet T. Proof Strategies in Linear Logic. *Journal of Automated Reasoning*, vol. 12, 1994, p. 273–304, Also available as Programming Methodology Group Report 70, Chalmers University, 1993
- Valette R. Supervision et réactivité pour les procédés discrets. *Séminaire CNRS-PROSPER maîtrise des risques et sûreté de fonctionnement*, Nancy, mars 1998.
- Valette R. Künzle L.-A. Réseaux de Petri pour la détection et le diagnostic. *Journée d'étude s3 : détection et localisation de défaillances*, Paris (France), novembre 1994
- Valette R. Pradin-Chezalviel B. Champagnat R. Raisonnement temporel pour la supervision et l'ordonnancement des systèmes dynamiques à événements discrets. *École thématique document et évolution du groupe de recherche i3 du CNRS*, vol. II, Cépaduès éditions, 2000, p. 103–117
- Veissberg J.-L. Présences à distance, No 2-7384-7678-3, Éditions l'Harmattan, 1999
- Young M. Riedle M. Brandy M. Martin J. Saretto J. An architecture for integrating plan-based behavior generation with interactive game environments, *journal of game development*, 2004
- Zagal JP. Mateas M. Fernandez-Vara C. Hochhalter B. Lichti N. Towards an Ontological Language for Game Analysis. *DiGRA 2005 Conference : Changin Views – Worlds in play*, Vancouver (Canada)