

Temps et processus

Anne NICOLLE
GREYC-ISLanD UMR CNRS 6072 & Pôle Modescos de la MRSH
Université de Caen – CNRS
14032 Caen Cedex
Mail : anne.nicolle@info.unicaen.fr

*Avec le temps,
Avec le temps,
Va, tout s'en va
...
Léo Ferré*

*Dis, quand reviendras tu ?
Dis, au moins le sais tu ?
Car tout ce temps qui passe
Ne se rattrape guère,
Car tout ce temps perdu
Ne se rattrape plus.
Barbara*

Résumé :

Les processus informatiques sont dans des rapports variés au temps : temps compté, temps représenté, temps social, temps propre. Ils sont dans le temps présent de leurs usagers lorsqu'ils gèrent une interaction à but pratique. Ils sont dans un temps passé lorsqu'ils simulent des processus passés à partir d'un modèle pour tester ce modèle. Ils sont dans des temps futurs lorsque la simulation a pour but de faire des prévisions, ou lorsqu'ils ont pour objectif de faire un plan d'action. Ils sont dans un temps imaginaire quand il s'agit d'un jeu. Mais ces temps s'entrecroisent, se rencontrent, se superposent. Les rapports des processus au temps servent à les classer : fonctions, processus cycliques, processus gérés par les événements, agents, processus apprenants, et à donner des caractérisations originales de certains types de processus interactifs comme les systèmes multi-agents par leurs rapports aux différentes sortes de causes.

1. Introduction

Bergson présente le temps matériel comme la détente d'un élan et le temps mental comme le déploiement en éventail d'une mémoire. La mémoire des machines, comme espace de stockage a peu de points communs avec la mémoire d'un être vivant. La mémoire des humains est à la fois une mémoire d'être vivant et une mémoire sociale. La mémoire sociale est instrumentée par des objets techniques et sémiotiques dont les machines informatiques sont un aspect récent. Nous avons déjà présenté la mémoire comme un processus plutôt que comme un lieu [Nicolle & al., 2001] et montré comment ce point de vue s'applique aux agents logiciels. La présente étude sur les rapports des processus informatiques au temps tient compte principalement du temps comme « détente d'un élan » et contribue à la construction d'une théorie des processus interactifs de durée indéfinie (PII). Le but de cette

théorie est de rendre compte de la pratique informatique qui conçoit et met en route des PII qui transforment notre société en instrumentant tous les rapports sociaux.

L'informatique s'est d'abord intéressée à des processus terminatifs, aux fonctions calculables par les ordinateurs, en référence à une théorie du calcul. Le temps de calcul prime alors sur le temps du calcul. En parallèle, elle s'est intéressée à des processus périodiques, pour gérer les données des entreprises en suivant des rythmes journaliers, hebdomadaires, mensuels ou annuels. Elle a mis alors l'accent sur la constitution, l'utilisation et la mise à jour des données. Le temps du calcul est articulé sur le temps social et le temps de calcul doit être compatible avec les échéances. Le paradigme objet a permis de voir ces deux points de vue de manière unifiée en considérant les types d'objets comme comportant à la fois une structure de données et des procédures de manipulation de ces données. Il réunit ainsi le paradigme du calcul, sans effets de bord, et le paradigme des données persistantes et de leurs transformations, en structurant le logiciel par la notion de type de données. S'il prend en compte deux rapports au temps : le temps de calcul et le temps du calcul, il réifie le temps dans des classes pour en faire un élément observable des logiciels par eux mêmes (par exemple, les classes *Time*, *Date* et *Calendar* en Java¹).

Actuellement, les ordinateurs mettent en route et font tourner en parallèle des processus interactifs de durée indéfinie qui communiquent entre eux et avec des êtres humains : les systèmes d'exploitation des ordinateurs d'abord, et de nombreuses applications : traitements de texte, tableurs, gestionnaires de bases de données ou de mails, butineurs Web. La théorie du calcul, la théorie de l'information et la cybernétique ne rendent compte que très partiellement des processus interactifs de durée indéfinie : les deux premières parce qu'elles ne s'intéressent qu'à la structure des processus, la deuxième parce qu'elle voit les processus comme des boîtes noires. Le paradigme objet ne résout pas actuellement de manière convaincante la modélisation des processus interactifs informatiques. Pour en rendre compte, il a introduit des objets actifs : applets Java ou *threads* par exemple, qui exécutent une boucle de comportement de durée indéfinie, et la notion d'événement, qui permet de rendre compte des interactions entre processus, ou entre processus et usagers. Mais la distinction entre fonction et processus n'est pas explicite car les deux notions sont implantées par des méthodes. Le paradigme objet n'apporte pas assez d'attention à la dynamique des calculs, des actions et des interactions qui sont faits par les machines dans le temps réel, les objets actifs communiquant tout en ayant des temps propres différents.. « The dynamic semantics are described using natural language, although in a precise way so they can easily be understood. Currently, the dynamic semantics are not considered essential for the development of tools; however, this will probably change in the future. » *OMG-Unified Modeling Language, v1.4* 2001, p. 2-7² La modélisation des processus, des objets actifs, des agents, est donc un champ ouvert de théorisation.

Un processus interactif indéfini est par nature spatio-temporel et événementiel : il capte et lance des événements dans le temps et l'espace. Chaque processus a un rapport particulier au temps et à l'espace en fonction de sa situation. L'objectif de cet article est de répertorier et d'analyser les différentes formes des processus dans leur

¹ <http://java.sun.com/>

² <http://www.uml.org/>

rapport au temps. Le rapport à l'espace, espace du réseau, localisation des machines, mériterait une étude semblable. Cette question des processus interactifs indéfinis vaut pour les processus artificiels (PIIA) mais vaut aussi pour les processus naturels comme les êtres vivants (PIIV). Cette ressemblance n'est pas considérée ici comme une analogie, mais comme un principe de modélisation :

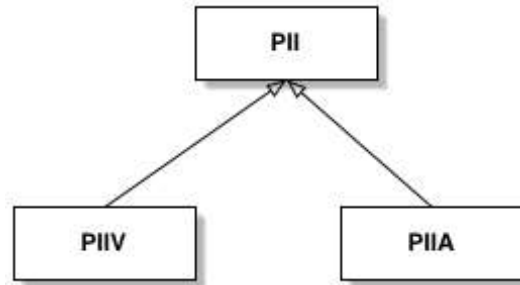


Figure 1 : Les classes de PII en UML

2. Le temps et les processus

Les processus interactifs sont perçus dans le temps des processus avec lesquels ils interagissent : autres processus informatiques, physiques, ou usagers. Pour décrire ce phénomène, on introduit la notion de « temps propre » d'un processus, par analogie avec la notion de « monde propre » en éthologie. Trois façons d'appréhender le temps sont envisagées ici : 1) le temps physique ou temps réel, qui est continu, 2) le temps social donné par les horloges et les calendriers, 3) le temps propre d'un processus, ou d'un usager d'un système informatique, pour qui le temps compte parce qu'il a des objectifs propres. C'est ce temps propre qui est en jeu dans les deux citations en exergue de cet article et c'est celui dont le modèle est le plus problématique. Le temps propre est structuré par la simultanéité, la succession et les rythmes des événements. La question de la durée relativement au temps propre, s'exprime par comparaison des durées de deux processus simultanés, par les pulsations propres (respiration, battements du cœur), ou par les rythmes produits ou perçus. Pour repérer les durées, on doit faire appel à des processus de référence ou à des appareils (sabliers, bougies, horloges). Le temps propre perçu permet de comparer des durées, pas de les mesurer, car il n'y a mesure que par l'action : si on n'a pas de réveil, on ne peut pas savoir combien de temps on a dormi, ni depuis combien de temps on a pas mangé.

Les processus informatiques sont dans le temps physique du processeur et de la mémoire des machines sur lesquelles ils s'exécutent. Leur temps est rythmé par le cycle de base du processeur, correspondant au temps de la plus petite opération exécutable. Ils sont dans un temps social car l'horloge des ordinateurs se règle sur une horloge de référence quand on accède au réseau. Il est à noter que ce temps social est discontinu puisqu'il s'exprime par un nombre décimal [Nicolle, 2003a]. Ce temps est accessible de l'intérieur des logiciels et permet de dater les documents, de calculer le temps utilisé par une fonction, d'appeler certaines fonctions à des heures ou des jours déterminés.

Le temps physique ou social est d'abord une ressource, un temps compté. Le temps de calcul des fonctions s'étudie par la complexité des algorithmes. Définir des fonctions et calculer leur complexité nécessite une notation pour les fonctions et une

notation pour le temps. Le temps noté t dans les fonctions des physiciens est un temps continu alors que le temps compté en complexité des algorithmes est un temps discret : $O(n)$, $O(n^2)$. C'est un temps abstrait qui se projette sur le temps concret en fonction de nombreux paramètres : cycle de base du processeur, efficacité de la compilation, choix de sécurité.

L'articulation entre le temps propre d'un processus et le temps physique de la machine sur lequel il tourne est complexe car plusieurs processus tournent en même temps sur une machine et se partagent le temps physique. Dans un processus, certaines fonctions comme les accès aux mémoires externes, les entrées et sorties, sont réalisés par des processus parallèles au processus principal, souvent sur des processeurs dédiés. Un processus peut avoir des sous processus du fait de sa programmation et l'attribution du temps aux divers sous processeurs n'est pas sous le contrôle du programmeur. Il peut juste fixer des priorités. Le temps physique est donc impropre à rendre compte du temps propre des processus, puisque les processeurs feuilletent le temps physique où les processus s'exécutent. En conséquence, calculer le temps d'exécution d'une fonction est du ressort du système d'exploitation, ni des usagers, ni du programme lui-même. Le temps de processus qui font des effets de bord, comme la lecture ou l'écriture dans des bases de données partagées est un temps qui doit s'articuler avec celui d'autres processus à cause des verrous posés par les processus sur les accès aux données pendant les modifications. Ce partage des ressources dans le temps ne nécessite pas la connaissance des processus concurrents et ils n'interagissent que de manière passive. Il s'agit encore d'un temps compté où les priorités prennent en compte l'urgence des différents travaux. Pour le partage des ressources physiques, il y a donc un transfert des raisonnements sur le temps vers des calculs de priorité qui évite de poser effectivement la question du temps réel.

Un processus qui capte des événements produits par d'autres processus est dans le temps car il ne peut capter qu'un événement à la fois et les événements non captés sont perdus. Être dans le temps est dit lorsque le rapport au temps ne se réduit pas à compter le temps ou à le représenter par une notation ou un mécanisme d'horlogerie. C'est un temps où des événements ont lieu, un temps qui compte, un présent, où le passé n'est plus là et ne peut pas être rejoué, et où le futur n'existe pas encore. Lorsque le temps compte, le déroulement d'un processus ne peut pas être identique d'une exécution à l'autre, puisqu'il dépend des événements reçus. Si le calcul d'une fonction a été interrompu pour des raisons quelconques, comme une panne de courant, on peut le recommencer, il se déroulera à l'identique. Les processus s'exécutant en arrière plan, comme l'horloge, ou le système d'exploitation, peuvent être relancés. Le temps perdu compte pour l'observateur, pas pour ces processus en eux-mêmes. Les processus interactifs sont dans un rapport effectif au temps, dans leur temps propre et dans le temps propre des processus avec lesquels ils interagissent. Le moment où ils produisent leurs sorties est signifiant, ainsi que celui où ils captent leurs entrées. C'est important pour eux-mêmes car les événements peuvent arriver en permanence et ceux qui ne sont pas captés sont perdus. C'est important pour les processus avec lesquels ils sont en interaction, qui vont pouvoir prédire l'arrivée d'événements par comparaison entre leur temps propre et le rythme des arrivées précédentes. Précisons les différences de rapport aux temps des

processus suivant le type de logiciel qu'ils exécutent.

3. Le rapport au temps des différents types de logiciels

Les processus naturels sont dans le temps réel et les processus périodiques servent de repères pour construire le temps partagé, ou temps social. Les théories du calcul sont de théories du temps compté (le temps de calcul). Les théories de l'information sont des théories qui ne tiennent pas compte du temps vécu, même si elle représentent le temps. Les temps linguistiques sont beaucoup plus riches que le temps vu comme position sur un axe. Ils distinguent non seulement le passé, le présent et le futur mais aussi le ponctuel et le duratif, le réel et l'imaginaire³ et rendent compte des places respectives des fenêtres temporelles [Gosselin, 96] évoquées par les récits. Ils servent de cadre à l'étude des rapports entre les processus informatique et le temps proposée dans cet article.

Les logiciels informatiques assurent plusieurs types de fonctions : 1) les systèmes d'information représentent « un état actuel du monde » et permettent de le gérer, 2) les systèmes de contrôle de processus physiques commandent le fonctionnement des appareils électroménagers, des véhicules ou des chaînes de production des usines, 3) les systèmes de contrôle de flux captent les événements produits par des phénomènes et les traitent en temps réel, 4) les systèmes de simulation cherchent à reproduire ou à anticiper des états du monde passés ou futurs en faisant tourner un modèle du monde initialisé sur un état connu ou supposé, 5) les mondes virtuels décrivent et font exister des mondes imaginaires. Examinons le rapport au temps de chacun de ces types de logiciels.

3.1. Les systèmes d'information

Les systèmes d'information ou SI sont des moyens de représenter et d'organiser des connaissances pour manipuler, traiter, conserver et présenter l'information sous toutes ses formes, comme par exemple les SI des entreprises (bulletins de salaire, bons de commandes, factures, états). Ils assurent le partage des bases de données, présentent des interfaces unifiées pour s'adapter aux habitudes des usagers, échangent des flux d'information en interne et avec l'extérieur. Les SI font partie du monde social, ils sont situés dans l'espace et dans le temps. Ils représentent l'état d'un monde actuel et ils doivent conserver une correspondance entre ce monde et sa représentation tant au niveau conceptuel qu'au niveau historique, puisqu'ils doivent produire des documents qui vont y jouer un rôle. Ils sont dans le temps social, où chaque chose doit arriver à son heure et où les retards doivent être négociés. Les SI stratifient le temps en gardant en archive des états antérieurs des bases de données. Ils sont donc « le déploiement en éventail d'une mémoire ».

3.2. La commande de machines physiques

La commande de machines physiques est faite maintenant par des systèmes informatiques. Les usines, les équipements scientifiques comme le Gani⁴, sont contrôlés par ordinateur. Les pilotes automatiques d'avion ou de métro peuvent

³ exemple : « Si j'étais riche, je ferais le tour du monde ! »

⁴ Grand Accélérateur National à Ions Lourds : <http://ganinfo.in2p3.fr/>

remplacer le pilotage humain, et l'assistance au conducteur des voitures de développe. Les machines physiques contrôlées par informatique sont dans le temps et l'espace physique. Les programmes informatiques contrôlent donc des processus qui sont dans le temps réel et elles doivent respecter les contraintes temporelles des machines. La mise à jour de la connaissance du monde actuel concerné par l'équipement est faite par des capteurs, en continu, sans intervention humaine. La commande logicielle est beaucoup plus souple et plus fiable que la commande mécanique, elle permet de faire de nombreux contrôles, des raisonnements sur le temps, sur l'état de l'environnement, sur les temps de réaction des acteurs humains. Elle traite les exceptions et prévient les erreurs. Avec les algorithmes *anytime*,⁵ elle s'adapte au temps disponible pour ajuster au mieux la précision des calculs en fonction du temps disponible alors même qu'il n'est pas connu à l'avance.

3.3. Le contrôle de flux

Les flux de données peuvent provenir d'appareils enregistrant des données visuelles, sonores, des mesures de température, de pression. Ils peuvent provenir d'un périphérique comme le clavier. Ils peuvent être transmis par le réseau : textes, musique, images animées, séquences vidéo. Les flux arrivent de manière continue, il faut les traiter dans le fil du temps. Le cas de la surveillance de flux est bien caractéristique du rapport de l'informatique au temps socio-technique. Les événements intéressants étant trop rares ou trop fréquents, les observateurs humains risquent de relâcher leur vigilance. En couplant l'observation humaine avec des systèmes automatiques, pour analyser en temps réel les enregistrements d'une caméra dans un lieu public, pour analyser les dépêches d'agences de presse ou les cours des bourses, le résultat est plus fiable. Le but de l'analyse automatique de flux est d'attirer l'attention des observateurs sur des événements intéressants. Il peut-être aussi de décider ce qu'il faut mémoriser et sous quelle forme, ou de faire des statistiques.

3.4. La simulation

Les logiciels de simulation représentent des états du monde⁶ non actuels, ils permettent d'expérimenter en étant hors du temps. Ils proposent une modélisation informatique qui est à la fois une description des phénomènes et une reproduction dynamique des processus par des moyens numériques. Les processus informatiques analogues aux processus physiques, biologiques ou sociaux, en changeant la substance des processus, conservent leur forme. Le but de cette modélisation est que leur description dans un matériau numérique explicite les raisons de leur fonctionnement et de leurs dysfonctionnements. Ils discrétisent le temps et l'espace, mais ils doivent assurer une correspondance entre le monde et sa représentation discrète. Cette correspondance au niveau des modèles n'implique pas une

⁵ Un algorithme *anytime* réalise un calcul d'approximation avec une précision d'autant meilleure qu'il a plus de temps. Il organise ses calculs de façon à pouvoir donner une réponse dès qu'on la lui demande et à continuer à améliorer les calculs tant qu'on ne lui demande pas de réponse.

⁶ ou plus exactement, d'un point de vue sur une partie du monde, dont on a fait un modèle qui doit être paramétré et validé par l'expérience.

correspondance historique comme pour les SI puisqu'il s'agit de concentrer le temps pour pouvoir faire revivre des phénomènes ou prédire leur évolution. Ils peuvent servir à expérimenter les modèles en les comparant aux phénomènes observés dans le passé, et à prédire leurs évolutions futures lorsque le modèle est validé. Ils sont couplés à des systèmes de graphiques, de statistiques, de comptage où le temps est représenté par un axe. Ils concentrent le temps passé ou futur sur un temps présent beaucoup plus court, le temps de l'observateur, introduisant par là des discontinuités dans le temps vécu. Ils peuvent implanter des processus décrits par des formules mathématiques mais ils cherchent aussi à dépasser les difficultés qui se font jour avec la modélisation mathématique des phénomènes complexes lorsqu'elle aboutit à des équations insolubles, en déplaçant la description du tout sur les parties (systèmes multi-agents).

3.5. Les mondes virtuels

On construit des mondes virtuels, pour montrer des mondes passés, futurs ou imaginaires : des monuments, des maisons, des ponts ou des espaces organisés (espace urbain, espace de travail). On peut y entrer, s'y promener, se déplacer pour les voir sous d'autres angles. C'est un jeu sur l'espace et le temps. Les mondes virtuels ont le même rapport au monde, à l'espace et au temps que les fictions littéraires ou cinématographiques. Ils sont dans le temps de celui qui les regarde et qui interagit avec eux, mais le temps qu'ils présentent est un temps imaginaire. Ils n'ont pas de contraintes de validité sur la représentation du monde aussi fortes que les logiciels de simulation. Par exemple, dans les jeux vidéo la gravité n'est pas toujours prise en compte. Par rapport aux maquettes construites par des moyens analogiques, comme le plan de Rome de l'Université de Caen⁷, les maquette numériques dites virtuelles sont dynamiques. Elles peuvent montrer l'état du monde à des moments différents du temps historique. Comme les simulations, les mondes virtuels créent des discontinuités dans l'espace et le temps vécus.

3.6. Récapitulation

Tous les types de logiciels présentés ci-dessus sont des processus interactif de durée indéfinie⁸ ou PII. Certains ne s'arrêtent jamais, ceux qui contrôlent les systèmes bancaires ou les centrales nucléaires par exemple. Ces processus ne s'arrêtent pas même quand il faut changer de version du logiciel, ils migrent sur la nouvelle version sans interruption. Ils ne s'arrêtent même pas quand on change les machines qui les font tourner. Ils passent sur la nouvelle machine quand elle est prête sans interruption de fonctionnement, car elle les reprend en charge dans l'état où ils sont. Il y a donc une certaine indépendance des processus aux processeurs. Bien entendu, ces processus font du calcul et ils traitent des informations, mais ces deux points de vue en donnent une vision partielle. Leur nature commune est d'être des processus de durée indéfinie, en interaction avec des usagers et avec des processus

⁷ www.unicaen.fr/rome/

⁸ Il serait plus classique de parler de processus infinis, mais le terme de durée indéfinie est plus précis puisque tout processus informatique finit par d'arrêter soit par lui-même, soit en erreur ou par une action « Quitter » de l'utilisateur, soit parce qu'il est remplacé par un processus plus performant.

physiques, biologiques, industriels. Ils sont dynamiques et en prise sur le temps présent. Ils font des actions physiques (ouvrir une vanne) ou signifiantes (poser une question). Ils modifient le monde socio-technique humain en instrumentant ses communications. Que peut-on en dire comme fondement à une théorie ?

4. Un modèle des PII

Avant de faire une théorie des PII, il fallait les décrire, ce que nous avons fait dans la section 3. Il faut maintenant en construire un modèle, à la fois descriptif et pour les processus artificiels, un modèle de réalisation. Sur cette base, le rôle d'une théorie est de rendre compte du modèle et de ces propriétés indépendamment des instances particulières et des exécutions particulières.

4.1. Modèle objet des processus et des systèmes

Un modèle de PII est une abstraction d'un processus concret du point de vue de son fonctionnement de processus et de ses interactions avec les autres processus. On distinguera trois étapes dans le fonctionnement des processus : la mise en route, la fin et le cœur du processus. Le cœur du processus fait souvent alterner des phases où le processus est actif et des phases où le processus est endormi. On peut donc le représenter dans son environnement informatique par un diagramme états/transitions où les transitions entre états sont causées par des événements (Figure 2).

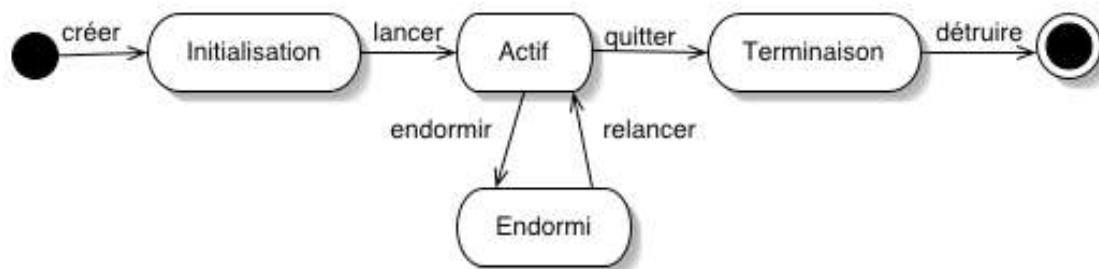


Figure 2 : les états d'un processus de durée indéfinie

Seul l'état actif du processus nous intéressera dans cette étude. Comme les changements d'état représentés dans ce schéma interviennent quand le processus capture des événements, ces changements d'état manifestent sa coopération avec les usagers et les autres processus, mais ni ses objectifs propres, ni les causes de son fonctionnement. Ce qui le caractérise en tant que PII particulier et ce qui détermine son rapport propre au temps est son fonctionnement dans l'état actif.

On peut distinguer deux classes de PII en fonction de leur fonctionnement dans l'état actif : 1) ceux qui traitent les événements (ou des flux) quand ils leur arrivent, comme les traitements de texte et les tableurs, et qui ne font rien si l'utilisateur ne fait rien. Ils sont dans le temps de l'utilisateur, dans le temps social en datant les objets qu'ils manipulent, mais ils n'ont pas de temps propre. 2) ceux qui ont des objectifs propres, qu'on appelle des agents. Ils exécutent une boucle de comportement : perception – délibération – action, avec rétroaction de l'action sur la perception. Ils filtrent dans les événements qui leur arrivent ceux qu'ils vont interpréter dans leur propre fonctionnement (c'est ce qu'on appelle la perception). Ils font des actions qui ne sont pas de simples réactions aux événements ou aux flux qu'ils ont reçus comme avec une table de transcodage. C'est pour ce type de processus que la question du

temps se pose comme temps propre ou temps vécu.

Certains processus ont un temps propre rythmé par une ou des pulsations : temps nécessaire pour réaliser une action élémentaire, temps de capture d'un événement par exemple. On les appellera des processus pulsés. La pulsation provient de la substance qui réalise le processus : les êtres vivants ont des pulsations qui leur sont propres comme la respiration ou les battements du cœur. Pour les artefacts mécaniques, la pulsation provient de la machine, qui la réalise par les fonctionnements cycliques (moteur, balancier d'horloge, ressort). Pour les artefacts informatiques, la pulsation est donnée par le cycle de base ou les tops d'horloge du processeur [Nicolle, 2003b].

Lorsqu'on cherche à caractériser les processus interactifs infinis pulsés, il n'y a pas à priori de temps universel, mais seulement la prise en compte d'un temps propre pour chaque processus pulsé. Le temps propre est premier, le temps partagé va intervenir comme conséquence des interactions, car il permet les synchronisations. La théorie doit rendre compte de l'apparition du temps commun aux processus ou temps social à travers les interactions entre les processus. Le temps universel est construit socialement, il peut être acquis par des processus informatiques comme résultat des interactions homme-machine ou machines-machines via un réseau, mais encore faut-il le référer à un temps propre pour qu'il soit signifiant. Étudier les processus interactifs suppose l'existence d'un système de processus en interaction dans un environnement. Certains de ces processus peuvent être des processus matériels ou des processus biologiques. Ils seront considérés par les processus informatiques en tant qu'ils sont instance d'une classe de processus au sens décrit ici.

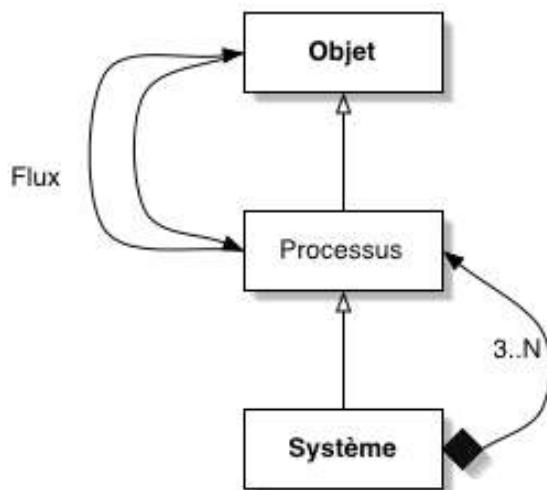


Figure 3 : relations entre les classes Objet, Processus et Système

4.2. Classement des processus interactifs en fonction de leurs causes

Aristote distinguait 4 types de causes : causes substantielle, efficiente, formelle et finale [Aristote]. John Stewart a proposé de réhabiliter la notion de cause en biologie [Stewart,]; cette section propose de l'utiliser pour comprendre les

opérations faites par les informaticiens dans la construction des logiciels.

Les causes substantielles relèvent de l'implantation d'un processus dans un processeur. La substance d'un processus est matière et n'est pas numérisable en tant que telle, mais la forme l'est, elle peut être transformée en signal. C'est pour cela qu'une trace de pas dessinée sur un papier ou sur un écran est toujours une trace de pas, parce que c'est la forme qui compte. On ne peut pas fumer la pipe dessinée par Magritte, mais c'est bien une pipe, quoi qu'il écrive en dessous du dessin. On ne peut pas dessiner une odeur ou un goût parce que c'est une trace indétachable de son support, un effet de la substance. Il existe donc des contraintes qui permettent ou non la transformation de la cause substantielle d'un processus en numérique et la possibilité de la simulation tient à la possibilité de transformer la substance sans transformer la forme, en trouvant des analogues numériques pour les causes formelle, efficaces et finales.

Un processus interactif numérisable est caractérisé par sa forme : les actions qu'il fait, les événements qu'il capte et les relations entre les actions et les événements. Si toutes les actions sont possibles à tout moment et sont choisies aléatoirement, le processus est dit aléatoire. Ces processus obéissent seulement à une cause formelle qui est le tirage aléatoire. Les processus déterministes décident de leurs actions en fonction des événements qu'ils ont captés et des actions qu'ils ont faites précédemment de manière univoque. Ils obéissent donc à des causes formelle (la fonction) et efficace (les actions et événements passés). Ils n'ont pas de cause finale (but, objectif interne). Les agents sont des processus dont la fonction de transition entre le temps t et le temps $t+1$ renvoie plusieurs résultats. Elle contraint les actions possibles au temps $t+1$, mais n'en décide pas. La décision effective d'action dépend d'une deuxième fonction, la fonction de choix, qui peut être aléatoire ou dépendre d'une cause finale. La cause finale est un principe de viabilité, un but ou des objectifs hiérarchisés. Elle est implantée par une méthode de choix indépendante de la fonction de transition : méthode heuristique, oracle. Un observateur peut prédire ce qui est impossible de faire à chaque instant à cause des contraintes, il ne peut pas prédire ce qui sera effectivement fait car les raisons internes évoluent au fur et à mesure qu'on s'approche du but, où qu'on s'éloigne des conditions de viabilité.

Type de processus	Causes
aléatoire	Formelle
déterministe	Formelle et efficace
agent	Formelle, efficace et finale

Figure 4 : Classement des processus actifs en fonction de leurs causes

Cette catégorisation à partir des causes permet de clore les catégories de processus pour des raisons générales, externes au fonctionnement des processus.

Tout logiciel a un but pour son commanditaire ou son acheteur. L'écriture d'un programme déterministe transforme le but, qui est une cause finale, en cause formelle et le lancement du programme active la cause efficace. L'écriture de cette cause formelle dans un matériau numérique permet de l'exécuter en activant une cause substantielle numérique. Pour les agents, la cause finale reste explicite, mais la planification est la transformation située des causes finales en causes formelles.

Lorsque le processus planifie ses actions, c'est pour maîtriser le temps réel par un raisonnement qui a lieu dans un temps différé.

La question des causes finales est liée à la question du temps propre. Lorsque la cause finale est transformée en cause formelle, il n'y a plus de temps propre du processus, parce que tous les déroulements possibles ont été prévus à l'avance. C'est un avantage pour les raisonnements théoriques sur les processus, mais aucun apprentissage n'est plus possible.

5. Conclusion

On peut distinguer plusieurs types de processus selon leur rapport au temps. Les processus terminatifs comme le calcul de fonctions ont un début et une fin inscrites dans leur déroulement. Leur but est implicite et ils s'arrêtent quand le but est atteint. Les processus non terminatifs ont une durée indéfinie parce que leur but implique qu'ils continuent à exister. Ils peuvent être opportunistes, attendant des événements pour les traiter, comme les traitements de textes. Ils peuvent être périodiques, cycliques comme une horloge s'ils se reproduisent périodiquement à l'identique. Le comportement des agents peut se reproduire périodiquement avec des modifications de comportement : adaptation ou apprentissage.

Tous les processus informatiques sont à la croisée du temps physique, parce qu'ils sont exécutés sur des processeurs ayant un cycle de base, du temps social parce qu'ils se synchronisent avec une horloge de référence et du temps propre des usagers avec lesquels ils interagissent. Ils utilisent le temps comme une ressource et comme un repère. Il y a rencontre entre le temps propre d'un processus et un autre temps : physique, social ou temps propre d'un autre processus, à la réception d'un événement. La plupart du temps, programmer, c'est gérer les causes finales d'un processus comme des causes formelles et par là même supprimer le temps propre du processus qui en résultera. Mais cette façon de faire supprime aussi les possibilités pour les processus d'évoluer à partir de leur expérience. Concevoir des logiciels plus sommaires et plus robustes, mais capables d'adaptation pourrait être une solution pour des applications trop complexes pour être totalement maîtrisées d'avance.

6. Références

Laurent Gosselin, 1996, Sémantique de la temporalité en français, Un modèle calculatoire et cognitif, Duculot, Louvain-la-neuve

Klaus Mainzer, 2002, The little book of Time, Copernicus Books, Springer-Verlag

Anne Nicolle, Pierre Beust, Vincent Perlerin, 2001, Un analogue de la mémoire pour un agent logiciel interactif, Revue In Cognito, n° 21, pp. 37-66

Anne Nicolle, 2003a, Le continu, le discontinu et le discret en informatique, Espace-Temps, Les cahiers, Association Espace-Temps, Paris, à paraître

Anne Nicolle, 2003b, Étude préliminaire à une théorie des processus interactifs infinis, Les cahiers du Greyc, n° 3, (18 p.)

John Stewart, La Modélisation en Biologie, in *Enquête sur le concept de modèle*, ed. Pascal Nouvel, PUF, 2002