



HAL
open science

Gestion des configurations dans les industries sous pression innovationnelle : le problème non-résolu de l'identité des systèmes

Gilbert Giacomoni

► **To cite this version:**

Gilbert Giacomoni. Gestion des configurations dans les industries sous pression innovationnelle : le problème non-résolu de l'identité des systèmes. *Revue française de gestion industrielle*, 2022, 36 (2), pp.75-97. 10.53102/2022.36.02.911 . hal-04398717

HAL Id: hal-04398717

<https://agroparistech.hal.science/hal-04398717>

Submitted on 18 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gestion des configurations dans les industries sous pression innova-tion-nelle : le problème non-résolu de l'identité des systèmes

Giacomoni Gilbert, Université Paris-Saclay, AgroParisTech, INRAE, Paris-Saclay Applied Economics, 91120, Palaiseau, France, gilbert.giacomoni@agroparistech.fr, Orcid: <https://orcid.org/0000-0003-2648-857X>

RÉSUMÉ : Gérer les évolutions de configuration d'un système en constante évolution est déterminant pour pouvoir prédire son comportement dans des environnements référentiels, les imprévus étant potentiellement lourds de conséquences (aéronautique, logiciel, santé). L'utilisation de solutions informatiques appropriées est donc stratégique pour connaître parfaitement l'identité des systèmes et anticiper tout problème d'invariance, de discernabilité et d'interchangeabilité. Pour que les formalismes spécifiques proposés par ces solutions informatiques soient adaptables à des systèmes et à des environnements référentiels évoluant sous pression innova-tion-nelle, la difficulté à surmonter n'est pas seulement pratique mais d'abord théorique. Notre recherche met en évidence un problème syntaxique et sémantique non-entièrement résolu dans la relation entre extension des connaissances et innovation. C'est un sujet important pour la recherche sur la gestion des connaissances et des systèmes d'informa-tion.

Mots clés: Innovation ; interchangeabilité ; configuration ; identité ; systèmes d'information

Configuration changes management in industries under the pressure of innovation: the unsolved problem of systems' identity

Abstract: The way in which a constantly evolving system is configured is a key issue in making predictions on its behavior in reference environments, with potentially major implications (aeronautics, advanced computer engineering, healthcare). The use of suitable IT applications is strategic to ensure comprehensive knowledge to deal with systems' identity, discernibility and interchangeability. Specific formalisms proposed by such IT applications are seeking to find an alignment with fast changing environments and systems coupled with innovation. The difficulty is theoretical and not solely practical. Our approach highlights an unsolved syntactic and semantic problem in the relationship between knowledge extension and innovation. It is an important topic for research on knowledge management and Information Systems.

Keywords: Innovation ; interchangeability ; configuration ; identity ; information systems ;

1. INTRODUCTION

En juin 2020, à la demande d'Airbus "*Production – Logistics & Transportation*", Toulouse (France), nous avons engagé une recherche de plusieurs mois. Le problème se présentait initialement de la manière suivante: "*In order to support upcoming ramp-up, we need to define efficient IT logistics solutions to answer production needs (...) both internal and external benchmark showed that all plants meet the problem without finding any satisfying solutions so far [i.e. working as planned; standardizing to avoid delayed differentiation and resulting fluctuation; not requiring a lot of surveillance; not consuming time and cost; etc.]*" (N.B. Head of the projects). En novembre 2015, une problématique de recherche similaire était discutée avec Rolls-Royce "*Configuration and Design Data Management Department – Trent 7000 & Airbus Projects*", Derby (UK), en s'accordant sur un résultat démontrable : en l'état de nos connaissances, il n'existait pas de solutions applicables permettant un traitement automatisé mais plutôt une aide à la décision (Giacomoni & Sardas, 2014). Nous pourrions bien remonter plusieurs années en arrière. Déjà en 2011, la complexité du problème était mise en évidence (Giacomoni & Sardas, 2011): toutes les potentialités des solutions informatiques (PLM, ERP, bases de données) étaient contrariées par l'existence de cette problématique: comment adopter une syntaxe et une sémantique permettant de mémoriser l'identité (ou l'équivalence ou l'invariance ou l'interchangeabilité) des systèmes, sachant que cette dernière ne peut être établie de façon universelle et absolue, tel un état de la nature, mais plutôt comme une relation artificielle, relative, conditionnelle et temporaire, suspendue à un état des connaissances (expériences reproductibles dans des environnements référentiels)? L'objet du présent article est de discuter l'origine théorique et les fondements syntaxiques et sémantiques de la problématique opérationnelle rencontrée.

Identifier les configurations d'un système en constante évolution est déterminant pour pouvoir suivre les innovations absorbées et donc les changements d'états survenus, afin de partager cette connaissance à des fins décisionnelles et d'amélioration de l'efficacité collective (Andersson et al., 2008, NASA¹ 2022). Il est attendu d'une science -- tout particulièrement de la logique, définissable comme la recherche des conditions formelles d'une vérité – qu'elle apporte des réponses (un corps de connaissances) à propos des relations entre un sujet (un concepteur ou un groupe de concepteurs) et un objet (un système) dans son environnement référentiel. Ces réponses sont indispensables pour pouvoir prédire les comportements d'un système à l'aide de solutions informatiques appropriées (en conception, en production, etc.). Au préalable, ce qu'il faudrait entendre par relations d'identité et de dépendance entre les propriétés d'un système et son environnement référentiel, relativement à un certain état des connaissances, doit être clarifié : s'agit-il de propriétés caractéristiques d'une situation donnée ou bien de propriétés universellement vraies et immuables dans le temps, toujours identiques à elles-mêmes ? A défaut, les relations entre innovation and extension des connaissances ne peuvent être anticipées et formalisées. En opérant dans un environnement fixe, la réponse est déjà complexe et considérée comme un problème d'interfaces de sous-systèmes et d'intégration des connaissances (Dibiaggio, 2012). Si, tout comme la structure interne du système, l'environnement peut aussi changer, la question de la maîtrise du comportement d'un système est a fortiori plus complexe encore (Alavi & Leidner, 2001). "*What constitutes novelty depends on what knowledge is already in the mind of the problem solver and what help is received from the environment in adding to this knowledge*" (Simon, 1996, p.105). Par exemple, lorsqu'un produit (un système) incorpore de nouveaux éléments (sous-systèmes), jusqu'à quel

¹ Configuration Management – Systems Engineering Handbook: <https://www.nasa.gov/seh/6-5-configuration-management>

point les propriétés du produit demeurent-elles inchangées s'il évolue dans un environnement opérationnel différent de celui habituel ? L'intégration des innovations peut s'avérer risquée du fait de la survenance d'incompatibilités ou d'effets croisés indésirables. Pour éviter cela, les systèmes doivent être correctement identifiés en fonction de leur identité (leur invariance, leur équivalence, leur interchangeabilité), en sorte que toutes les parties prenantes en aient une appréciation commune. Ce problème structurel de gestion des relations du tout avec ses parties (Van der Ven, 1986) relève de la gestion des configurations (Fichman, 2004; Pavlou & El Sawy, 2010) et le choix de systèmes d'information appropriés est stratégique (NASA 2022)². La gestion des configurations est essentielle à la structuration des représentations humaines, notamment si elles doivent permettre de faire des prédictions conditionnelles ayant une validité restreinte, relative à l'état des connaissances et des expériences menées dans leur environnements référentiels, car les implications pour des industries particulièrement sensibles comme celles étudiées ici (aéronautique, informatique/logiciel, santé) peuvent être considérables. De tels processus décisionnels doivent trouver leur traduction et leur paramétrage en langage formel, autrement dit être constructibles, au sens prouvables, décidables, afin de tirer le meilleur parti des solutions informatiques de type Product Lifecycle Management (PLM) et Enterprise Resource Planning (ERP), qui sont à la fois complémentaires et le plus souvent intégrées l'une à l'autre afin de couvrir l'ensemble des besoins des industriels, ainsi que les solutions de type bases de données relationnelles³. Leur application à des activités très évolutives sous l'effet de l'innovation, met en exergue la tension inévitable entre exploitation et exploration de l'identité des systèmes (leur équivalence ou leur invariance ou leur interchangeabilité) qui est à la base de leur reproductibilité. C'est clairement un sujet d'importance en gestion des systèmes de connaissances⁴ (Alavi & Leidner, 2001) et d'information (Gable, 2010; Sidorova et Al., 2008). Pour répondre à notre questionnement, après avoir démontré son importance et mis en lumière les difficultés industrielles (section 2), caractérisé les problématiques d'identités conditionnelles et leur indécidabilité pour les concepteurs (section 3), analysé les formalismes et leur combinaison pour gérer les configurations (section 4), délimité leurs champs des possibles pour gérer des identités conditionnelles (section 5), nous proposons un mouvement stratégique vers une position basée sur des fondements syntaxiques et sémantiques étendus (plus généralisants) que seule une logique non-standard peut permettre (section 6). Notre recherche se veut "[e]mpirically relevant, process-oriented" avec "a commitment to theoretical generality" (Cyert and March, 1963, p. 3) discutant la littérature en vue d'étendre et de redéfinir les théories existantes. Elle est basée sur des années d'investigations empiriques, plus spécialement chez *Airbus Group* et le présent article actualise nos avancées sur la période 2015-20.

Comme cela a été expliqué précédemment, l'identité des systèmes n'est pas une relation naturelle et absolue attendant quelque part d'être découverte et caractérisée une fois pour toute. C'est au contraire une relation artificielle et temporaire, comme l'est une inférence⁵, conditionnellement à l'état des connaissances et des expériences menées dans leurs environnements référentiels, et devrait être mémorisée comme tel dans un langage sémantique approprié. La rationalité standard permet aux concepteurs de gérer des identités (id-entités) relativement à un état référentiel des connaissances immuable (Turing & Girard, 1999; Girard,

² "to document and provide full visibility of the product's present configuration and the status of achievement of its physical and functional requirements, such that everyone working on the project at any time in its life cycle is using correct and accurate information." ISO 10007 standard.

³ Logiciels permettant de créer, d'utiliser et de maintenir des ensembles d'informations connexes.

⁴ Ensemble de méthodes et d'outils qui doivent permettre d'identifier, d'analyser, de stocker et de partager les connaissances des membres d'une organisation (p.114).

⁵ "Inductive or plausible reasoning when the necessary information is not available (deductive reasoning whenever enough information is at hand to permit it)" (Jaynes, 2003 p. x).

2004). Dès lors que cet état change pour refléter l'évolution des entités considérées ou de leur environnement référentiel sous l'effet de l'innovation, ils doivent adopter une rationalité non-standard, basée sur des fondements scientifiques plus étendus (Giacomoni, 2018). Dans cette approche, la compréhension et la conception d'un monde dynamique est plus complexe, ce que P. Senge (2006) a précisément décrit dans sa théorie de 'Learning Organization', et avant lui H. Simon (1996) dans les sciences de 'l'Artificiel', ainsi que E.T. Jaynes' dans son 'extended logic of science' (2003). Pour que la prise de décision conditionnelle, caractéristique des dynamiques industrielles soumises à la pression de l'innovation, puisse trouver dans les applications de type PLM et ERP des formalismes plus adaptés, il faut opérer ce mouvement stratégique vers une nouvelle position scientifique.

2. IMPORTANCE DE LA GESTION DES CONFIGURATIONS POUR LES INDUSTRIES SOUS PRESSION INNOVATIONNELLE

Les entreprises doivent gérer des produits, des installations ou des projets de nature profondément complexe et évolutive (Boldrini, 2012). Ce qui représente des milliers de références de définition et plus encore d'articles en exploitation, avec des versions successives et des documents qui leur sont associés tout au long de leur vie. Pour assurer la fiabilité des définitions, des exploitations, des maintenances opérationnelles, il faut être en mesure de garantir la cohérence et la gestion de la qualité de l'ensemble des données techniques : suivi des événements, des modifications nécessaires, des impacts possibles sur l'ensemble, des non-conformités, des actions, etc. (Molet, 2011). Ces exigences se renforcent dans une économie de la variété où les produits se diversifient et se renouvellent à un rythme accéléré. Le choix de systèmes d'information adapté est donc stratégique.

Pour améliorer leur compétitivité, réduire les cycles et les coûts, accroître la réactivité, maîtriser la complexification de leurs produits et de leurs procédés, les entreprises doivent consentir des efforts et des investissements de grande ampleur pour s'informatiser en reliant, progressivement ou simultanément, tous les segments fonctionnels : R&D, systèmes d'ingénierie, marketing, qualité, achats, etc. Nombre d'entre elles se tournent notamment vers des solutions de type *Product Lifecycle Management* (PLM) qui se couplent à des solutions de type *Enterprise Resource Planning* (ERP), respectivement distribuées par Dassault système ou Siemens ou PTC et aussi par SAP ou Microsoft, en fonction de leurs sphères d'influence. Il s'agit d'outils d'entreprise ou progiciels, pouvant couvrir des besoins relativement variés et permettant ainsi de gérer le cycle de vie des produits, de leur conception à leur mise au rebut (Stark, 2004; Amann, 2002; Batenburg et al., 2005; Pol et al., 2005; Abramovici & Aidi, 2013; Sharma & Patil, 2015; Schweitzer et al. 2019). Ces solutions sont à la fois modulaires et intégrées autour d'un noyau unique⁶ (Mostefai & Batouche, 2005; Dambietz, 2022) pour offrir un environnement architecturé⁷ où il est possible de gérer le processus de développement depuis l'idée initiale jusqu'au produit fini. Ils ont été conçus ainsi pour traiter et partager entre les différents acteurs, toutes les données de l'entreprise représentant les produits et les processus liés à l'innovation et à l'évolution. *"The data to be exchanged between PLM and ERP is identified and these data have to be communicated to the ERP from PLM (...) based on ontologies to resolve syntactic and semantic data conflicts"* (Prashanth & Venkataram, 2017, p.2269). L'innovation peut se définir ainsi: *"development and implementation of new ideas by people who, over time, engage in*

6 Au sens informatique, un noyau unique offre une structure alternative aux systèmes hétérogènes interfaçant des solutions logicielles ayant leur propre noyau et spécifiques à chaque domaine fonctionnel (conception, production, finances, etc.)

7 L'ergonomie d'un interface homme-machine harmonisé unique est préférable (pour la formation, la prise en main, les versions successives, etc.) à des environnements hétérogènes avec interfaces multiples, quitte au besoin à être moins spécifiques sur le plan des fonctionnalités.

transactions with others within an institutional order” (Van der Ven, 1986, p.590). Comme cela a été dit en introduction, la manière de configurer un système en évolution permanente sous la pression de l’innovation est déterminante pour la connaissance d’un programme, d’un projet ou d’un produit, à des fins décisionnelles et d’amélioration de l’efficacité collective. Cela relève de la gestion des configurations et tout particulièrement du problème des relations du tout avec ses parties. Un système – qu’il soit industrialisé ou pas, prévu pour un usage unique ou pour être maintenu tout au long de son cycle de vie – peut se définir comme un groupe d’éléments en interrelation en vue de satisfaire un certain objectif. Ce dernier pouvant être l’expression d’un besoin à satisfaire, à l’instar d’un modèle comportemental répondant à des attentes fonctionnelles ou techniques. Les grandes organisations de l’industrie aéronautique et aérospatiale, de l’ingénierie informatique et logicielle, de la santé, partagent des caractéristiques communes telles qu’un renouvellement rapide de leurs systèmes, en contrepoint du besoin de sériation réclamant un réemploi des systèmes. Observer les acteurs opérer, décrire, concevoir, gérer les identités des systèmes (décompositions, relations, identifications) à l’aide de leurs solutions informatiques de types PLM, ERP et bases de données relationnelles, est une source d’apprentissage pour adapter ces dernières aux instabilités permanentes des activités soumises à la pression de l’innovation. C’est particulièrement instructif lorsque cela touche aux limites ou aux extensions des fondements scientifiques sur lesquels s’appuient ces solutions informatiques pour aider les acteurs à anticiper les comportements et les performances des systèmes d’après les observations et les expériences déjà réalisées dans les environnements référentiels. Nous pensons, comme cela a été mentionné précédemment, que c’est un sujet important pour la recherche en management des connaissances et des systèmes d’information.

Pour modéliser les objets, les processus et les connaissances, tout un panel de fonctionnalités est prévu dans les PLM et les ERP (Grieves, 2006; Benbya & Meissonier, 2007; Morshedzadeh I., Ng A.H.C. & Jeusfeld M., 2021), incluant les éléments (matières premières, composants, sous-ensembles, produits finis, etc.) et les évolutions de configuration (par dates ou par rangs d’applicabilité⁸, des mises à jour tenant compte des arborescences de codifications normalisées, des cas d’emploi, variantes, interchangeabilités, traçabilité, etc). Il n’y aurait a priori plus qu’à sélectionner les fonctionnalités nécessaires et suffisantes parmi celles proposées, pour gérer correctement les données techniques, les configurations, les évolutions, d’une activité industrielle donnée. Cet exercice n’a rien d’évident (Srivardhana & Pawlowski, 2007; Molet, 1993). *“PLM systems are not ready yet to implement these functionalities [relative to approved configurations]”* (Lanzillotta, 2015)⁹. Les fonctionnalités sous-tendent des formalismes gestionnaires (pour développement de systèmes, pour productions en série, ou plutôt unitaire, avec ou sans assemblage, etc.) ayant servi de modèle à la conception des progiciels pendant plusieurs années. L’existence de ces fonctionnalités résulte en fait d’une traduction logicielle de pratiques industrielles réelles (observées) ou imaginaires (espérées). Personnaliser l’usage d’outils PLM ou ERP ou bases de données relationnelles ainsi conçus réclame une transcription pertinente de l’activité désirée (notamment via le paramétrage), quitte à adapter certaines pratiques. C’est toute la sensibilité des rapports entre les systèmes d’information, les individus et les organisations (Reix et al, 2011). Dans certains domaines et pour certaines industries, les PLM et les ERP ont démontré leur capacité à s’adapter à différents types d’activités économiques et à améliorer la productivité et la fiabilité des processus de développement (Mermi- nod & Rowe, 2012). Mais qu’en est-il pour des activités industrielles conjuguant gestion inten-

⁸ Le terme d’effectivité est aussi employé pour exprimer la différence entre applicable et appliqué. Effectivité exprime la qualité de l’adéquation entre ce que l’on fait effectivement et ce que l’on voulait faire.

⁹ Configuration and Design Data Management department of Rolls-Royce Plc, Derby (UK). November 2015.

sive de l'innovation (impactant en permanence les données techniques) et sériation (reproduction en série) ?

Les exemples ci-après empruntés à l'industrie spatiale (le premier étant repris et développé en section 5), nous donne un aperçu des enjeux liés à ces questions.

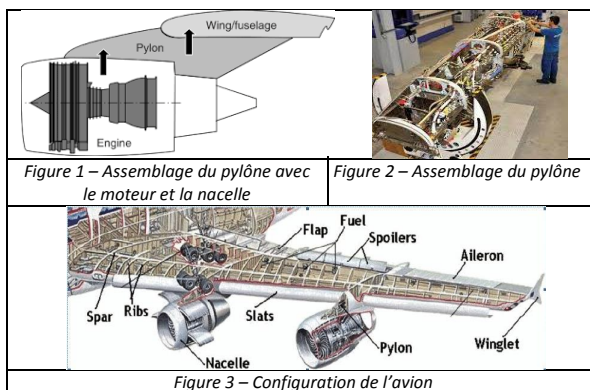
▪ Evolution de l'environnement interne (i.e. de la configuration): installations informatiques de contrôle de vol

En septembre¹⁰ 2008, sur un A330, une configuration non-conforme d'une des trois installations informatiques de contrôle de vol, a provoqué un atterrissage violent, au point de nécessiter le remplacement du train principal. Airbus et l'European Aviation Safety Agency¹¹ ont alerté les opérateurs de maintenance sur le respect des seules combinaisons autorisées par le constructeur: *"To prevent an uncertified configuration that may result in unexpected operation of the aircraft systems owners and operators should adhere to the interchangeability and mixability rules given in Airbus type certificate holder documentation"*.

▪ Evolution de l'environnement externe : non-interchangeabilité des modèles 737 & 737Max

En 2019, le Boeing 737Max a été confronté à des problèmes d'autopiloteage: *"automation-related accidents have come to make up a significant proportion of all air disasters (...) when the system works as designed but the pilot becomes confused and does the wrong thing (...) Boeing has built a plane with a fundamental aeronautical issue that it thought would be resolved by adding a new automated system. But the company convinced airlines and the F.A.A. that the planes were essentially interchangeable with earlier models of 737, and therefore pilots who were already trained in flying older 737s would not need comprehensive additional training on the new system. The F.A.A. agreed with that conclusion (...) That has proved to be a terrible miscalculation."* (Wise, 2019).

Une problématique analogue a été rencontrée dans les usines d'Airbus Group, ainsi par exemple, concernant l'assemblage du pylône¹² avec le moteur et la nacelle¹³ (Cf Figures 1 et 2) qui varie selon la configuration de l'appareil (Cf Figure 3). L'interchangeabilité des versions successives de ces systèmes est donc relative et pas absolue.



Nous aurions pu tourner notre regard vers d'autres industries, par exemple l'industrie pharmaceutique, et nous aurions retrouvé des problématiques analogues : le générique partage avec l'original la même substance pharmacologique active (principe actif), la même forme galénique, la même voie d'administration, le même dosage et les mêmes indications (recherche clinique, autorités sanitaires) que le médicament dont il est la copie. Il est interchangeable avec la préparation originale et est équivalent sur le plan thérapeutique. Mais la preuve est

¹⁰ Source Air Transport Intelligence News, 29/11/08 & Aviation Human Factors Industry News, 12/2008 IV(48):1

¹¹ European Aviation Safety Agency [Safety Information Bulletin 86, (2008), p.2]

¹² ou mât-réacteur, structure rigide fixant le réacteur à la voilure.

¹³ Carter profilé contenant et protégeant le moteur.

complexe et les études cliniques sont établies indirectement, par bioéquivalence. C'est également le cas pour des médicaments originaux lorsqu'il y a modification de leur formulation, changement de processus ou de site de fabrication.

3. CARACTERISATION DES PROBLEMATIQUES D'IDENTITES CONDITIONNELLES INDECIDABLES POUR LES CONCEPTEURS

Les systèmes conçus et produits par l'European Aeronautic Defense and Space Company se destinaient à l'activité spatiale (lanceurs et satellites), aux missions de défense (missiles stratégiques) et à l'aviation. La recherche principale a privilégié une approche longitudinale pour les suivre et les étudier sur une période de plusieurs années de 2002 à 2020. Ces systèmes se caractérisaient par des cycles de réalisation longs, avec des milliers de composants, des technologies de pointe hétérogènes et un renouvellement fréquent de leurs définitions. Une recherche secondaire croisant des sources directes et indirectes s'est consacrée à la santé, à l'informatique/logiciel, aux développements de groupware¹⁴ et de bases de données. Nous avons étudié ces activités sur une période plusieurs mois en privilégiant une approche transversale. Ces systèmes mobilisent des milliers de contributeurs différents, de sous-systèmes hétérogènes interfacés et des millions de nouvelles lignes de codes. Pour l'ensemble de ces activités, le management des évolutions de configuration de ces systèmes complexes implique des considérations sécuritaires, économiques et des impacts médiatiques¹⁵ (Feeney, 1999).

– COMMENTAIRES METHODOLOGIQUES –

L'approche longitudinale était appropriée pour étudier les systèmes complexes ayant des cycles de conception, de production et d'exploitation pluri-mensuels, voire pluri-annuels. Elle l'était également pour évaluer dans des conditions réalistes, l'adéquation de l'implémentation et la performance des solutions informatiques destinées à gérer des milliers de composants, des combinaisons hétérogènes de technologies avancées dont les définitions étaient sans cesse renouvelées. Dans le respect des obligations de confidentialité, il y avait libre accès aux centres de conception et de production, ainsi qu'aux documentations, bases de données et quantités d'informations, séminaires, groupes de travail à différents niveaux de responsabilité. Ceci a été facilité par un statut de chercheur de longue date chez Airbus et encadré sur le plan opérationnel par des accréditations et des engagements contractuels rigoureux.

La seconde recherche additionnelle, basée sur une approche transversale de plusieurs mois, était pertinente pour comparer l'adéquation et la performance des solutions informatiques aux philosophies gestionnaires et aux architectures variées, destinées à gérer les évolutions de configurations dans des contextes tels que ceux déjà signalés (santé, ingénierie informatique et logicielle, développement de groupware et base de données). Nous avons pratiqué la recherche-action et la recherche-intervention. Nous avons suivi l'école de pensée de la 'recherche en problématisation' ou 'problemistic search [for alternatives]' (Cyert & March, 1963) – "*the phenomenon where firm managers pursue explorative activities that contribute to the development of alternatives to existing activities, i.e. innovation, in response to a specific problem*". Autrement dit, nous nous sommes intéressés à l'apprentissage organisationnel et à la génération d'alternatives, déclenchés par la performance des solutions informatiques en dessous du niveau d'aspiration.

Nous avons mixé des sources directes et des sources indirectes. Concernant les sources directes, dans le respect des obligations de confidentialité, il y avait libre accès aux centres d'ingénierie, ainsi qu'aux documentations, bases de données et quantités d'informations, séminaires, groupes de travail à différents niveaux de responsabilité. Ceci a été encadré sur le plan opérationnel par des accréditations et des engagements contractuels rigoureux. Concernant les sources indirectes, nous nous sommes concentrés sur une littérature restreinte, puis nous nous sommes autorisés à réexaminer et à étendre les hypothèses d'origine s'avérant imprécises. Dans cet effort assumé de "problématisation" (Alvesson & Sandberg 2020), nous avons revisité nos propres hypothèses pour les consolider en nous

¹⁴ Ensemble des processus et procédures d'un groupe de travail devant atteindre un objectif particulier, en incluant les logiciels et communications conçus faciliter ce travail.

¹⁵ The National Institute of Standards and Technology estimated at \$60 billion the losses incurred by US manufacturing and commerce due to bugs contained in software (S and T Press USA-n°324-Sept.2002).

appuyant sur une littérature un peu plus étendue à chaque round, en sorte que le problème soit considéré sous de nouveaux angles pour ouvrir de nouveaux horizons interprétatifs ou explicatifs.

Ce qui rend difficile la gestion des configurations est l'incapacité à prédire comment un système complexe se comportera quand de plus en plus d'éléments et de sous-systèmes évoluent et interagissent. La difficulté tient à la complexité de l'étude combinatoire de toutes les combinaisons possibles dans des temps acceptables, mais aussi à l'impossibilité de décrire l'infinité des effets que les variations possibles de l'environnement référentiel sont susceptibles de produire. L'environnement référentiel, aussi appelé contexte ou situation (Lautman, 2006), permet de décrire et de comprendre un phénomène ou un comportement apparent. Cette connaissance est donc fondamentale et doit être mémorisée. C'est un principe de base pour toute expérimentation que de connaître l'environnement référentiel, car s'il venait à se modifier (ambiance, température, pression, etc.), les systèmes pourraient avoir des comportements différents. En contrepoint, des comportements identiques ne résultent pas nécessairement de configurations similaires. Des systèmes différents peuvent présenter des comportements analogues parce que les environnements en sont les générateurs. Par exemple, la pesée de deux objets permet de comparer leurs poids apparents et de conclure à leur égalité si l'équilibre de la pesée est observé. Il est ainsi possible d'anticiper leurs comportements respectifs (actions ou variations) dans un environnement référentiel comme l'air. Quand la pesée est faite dans un environnement référentiel différent – l'eau – les poids apparents des objets diffèrent, la poussée d'Archimède se rapportant au poids du volume de fluide déplacé (l'eau vs l'air). Un environnement référentiel différent – l'eau – peut ainsi générer des pesées équilibrées (comportements analogues des objets : flottaison) alors même que ces derniers diffèrent par leur volume. Le changement d'environnement référentiel exige de concevoir une relation variationnelle plus générale entre poids et volume, étendue à tout fluide, pour rendre compte des phénomènes et prédire les comportements apparents des objets. Ce raisonnement par extension, connu depuis Archimède (Beauzamy, 2012, p.119) est à l'oeuvre derrière la gestion des configurations et plus généralement derrière toute inférence scientifique (Jaynes, 2003) basée sur une information incomplète (jusqu'à ce qu'une relation plus étendue soit établie et ainsi de suite). Il faut donc considérer les transformations des systèmes, des sous-systèmes et des environnements ainsi que l'ordre de leurs occurrences, car des irréversibilités peuvent se manifester au cours de ces transformations.

Toute référence à des systèmes, sous-systèmes et environnements pour représenter des entités immatérielles (compétences, connaissances, etc.) et matérielles (composants, composés, documents, etc.) exige que leur identité soit définie et commune pour l'ensemble des parties prenantes (concepteurs, producteurs, clients, etc.). L'identité d'un système ou d'un environnement est définissable comme la qualité d'être (reconnaisable comme) le même qu'un autre (id-entité)¹⁶ et cela permet de reproduire autant de copies que nécessaires (Winter & Szulanski, 2001). Tout le monde s'accordera sur le fait que l'identification des systèmes ou sous-systèmes est cruciale pour pouvoir les reproduire et les remplacer dans des environnements référentiels. Par conséquent, les identifier et les reconnaître (dans les stocks, les bases de données, les bibliothèques, etc.) est crucial également, d'autant plus si des changements surviennent et que les générations successives de systèmes coexistent. La gestion des configurations exige donc des stratégies pour définir (sous-systèmes, systèmes, environnements), nommer (sémantique et syntaxe) et structurer (relations de dépendance). La définition des systèmes est attachée à leurs propriétés, lesquelles reflètent des relations de

¹⁶ ou interchangeabilité dans l'industrie – "The ability of one product, process or service to be used in place of another to fulfill the same requirements", according to the North Atlantic Treaty Organization (NATO) and the International Organization for Standardization (IOS).

dépendance internes avec les sous-systèmes et rendent compte de comportements caractéristiques dans des environnements référentiels. Par exemple, la propriété 'poids' traduit une relation de dépendance 'force d'attraction d'une masse sur une autre' dans 'l'air, à l'équateur et au niveau de la mer'. De ce fait, les relations de dépendance entre sous-systèmes et systèmes peuvent être structurés à travers des configurations rendant compte de comportements attendus (variation, action, mouvement, travail utile, etc.) dans des environnements référentiels. Les identifications (numéros de séries ou de fabricant, dénominations, etc.) sont attribuées aux sous-systèmes et systèmes conformément aux normes applicables (sémantique et syntaxe) afin de les rendre discernables. Il faut préciser que les identifications représentent non-nécessairement un objet (système ou sous-système) unique mais plus généralement la classe de tous les objets (systèmes ou sous-systèmes) de même identité et présentant de facto les mêmes propriétés (Cartwright, 1971, 1987; Watts, 2015; Quigley, 2019).

Ce n'est pas chose simple comme le montre les exemples suivants : dans l'ingénierie logicielle, l'*Unicode* est un protocole de conception destiné à absorber la pression évolutionnaire produite par la prolifération des ensembles de caractères nationaux (André & Hudrisier, 2002; David, 2011). Ce protocole de conception est une uniformisation, à l'instar d'un Localisateur Uniforme de Ressource (URL) identique partout dans le monde, utilisé par les navigateurs pour localiser une ressource (fichier, document, image, etc.) sur internet en spécifiant une adresse unique (deux URLs identiques relient la même ressource, mais peuvent pointer vers une qui n'existe plus ou a été déplacée). Dans le système des noms de domaines sur internet, l'URL se compose de différentes séquences alphanumériques :

//site address/path/filename. Un URL absolu indique un chemin complet et est libre de toute relation. Pour accéder à une ressource sur un site internet, il faut naviguer à travers une structure (directory). Il est adapté pour un référencement par exemple ou pour créer des liens entre des sites web distincts. Mais la mise à niveau des sites, pour maintenance ou transfert, s'accommode mieux d'un URL relatif qui n'indique que le chemin relatif à la racine du site et permet d'éviter les problèmes de mise à jour des liens et des pages. Une standardisation procède par changement de définitions, d'identifications et de structures: "*The development and implementation of concepts, doctrines, procedures and designs in order to achieve and maintain the compatibility, interchangeability or commonality that are necessary to attain the required level of interoperability, or to optimize the use of resources, in the fields of operations, material and administration*", (OTAN – Section inter-Organisations). Un autre exemple concerne la performance des systèmes d'information distribués à grande échelle en environnement sous pression (tel qu'un site web soumis à une charge dynamique et à un volume de requêtes extrêmement élevé). Identifier la cause principale d'une anomalie revient à chercher tous les invariants *incomplets* (Ge & Al., 2013, p. 1054) ou *reliés* (Jiang & Al., 2006, pp.825) dans un réseau de relations entre des paires de mesures d'intensité de débit (monitoring data). De telles relations sont des équations supposées exprimer fidèlement la réaction du système, d'après son architecture, sa capacité matériel (hardware), des applications logicielles, etc.

Il est stratégique de pouvoir statuer sur les identités et les relations de dépendance lorsqu'elles évoluent. L'issue doit donc être décidable dans les temps impartis. Si l'exploration des scénarii possibles est de nature combinatoire et qu'il est impossible de tous les passer en revue (problème dit NP-complet), il est alors crucial de conditionner la validité de la décision aux seuls scénarii étudiés et d'enregistrer cette information pour éviter toute mésinterprétation. Les solutions informatiques doivent permettre de formaliser et d'accélérer ce processus décisionnel. A défaut, elles sont impuissantes à gérer les prédictions conditionnelles et

l'indécidabilité. Ce qui est presque toujours le cas. La décidabilité est en effet un processus constructible, au sens prouvable, débouchant sur une réponse 'oui' ou 'non' universellement valide. En réalité, il n'existe pas de réponse absolue constructible attendant d'être établie ou prouvée une fois pour toutes : *"It [the undecidability] signifies only the incompleteness of our own information"* (Jaynes, 2003). Cette approche va dans le sens de la position d'H. Simon's dans les sciences de l'artificiel (Simon, 1996). Sa thèse est que certains phénomènes sont 'artificiels' en un sens très spécifique : ils sont tels qu'ils sont seulement parce que le système est façonné par des objectifs ou des intentions, en fonction de l'environnement auquel il se destine (p.xi). *"The natural sciences are concerned with how things are. Ordinary systems of logic - the standard propositional and predicate calculi, say -- serve these sciences well. Since the concern of standard logic is with declarative statements, it is well suited for assertions about the world and for inferences from those assertions. Design, on the other hand, is concerned with how things ought to be, with devising artifacts to attain goals. We might question whether the forms of reasoning that are appropriate to natural science are suitable also for design. One might well suppose that introduction of the verb "should", may require additional rules of inference, or modification of the rules already imbedded in declarative logic."* (p.114-115). Cette nouvelle position stratégique peut changer notre regard sur la manière dont les solutions informatiques de type PLM ou ERP ou bases de données relationnelles peuvent traiter une gestion des configurations (à validité) 'conditionnelle', et plus généralement sur la manière de formaliser un processus de décision conditionnel en statistiques ou en recherche opérationnelle, dès lors que le raisonnement pas inférence est à l'œuvre (Jaynes, 2003).

4. ANALYSE DES FORMALISMES DE GESTION DES CONFIGURATIONS ET LEUR COMBINAISON

Les formalismes dont il est question sont des modélisations de l'architecture dynamique des systèmes¹⁷. De nombreuses architectures ont été adoptées et même combinées (sous certaines conditions) dans les PLM, les ERP et les bases de données relationnelles, afin d'améliorer les performances industrielles par une bonne gestion des changements et des invariances des systèmes complexes (Mantripragada & Whitney, 1999; Giacomoni & Sardas, 2014). Les architectures sont le fruit de choix stratégiques actionnables via des définitions, des identifications et des structures qui évoluent à différents rythmes. Il ne s'agit pas du tout du concept de *multi-vues* consistant à représenter un objet technique exploité dans différents contextes de conception ou de configuration (Rehman & Yan, 2007). Le *multi-vues* permet d'associer différents points de vue (structurel, technologique, géométrique, fonctionnel, comportemental ou contextuel) aux acteurs métiers impliqués dans le cycle de vie (Bernard, 1996). Ce concept a fait l'objet de travaux de recherche dans les domaines du PLM et du développement des produits (Bronsvort & Noort, 2004 ; Noël, 2006 ; Bouikni et al., 2008, IBM & Dassault Systèmes, 2008). Il ressort de ces travaux que la construction, l'évolution de chaque vue, et le contrôle de la cohérence dans le modèle produit, font partie des enjeux toujours en vigueur (Bernard & Perry, 2003), lesquels nous ramènent aux problématiques de gestion des configurations (complexité combinatoire, indécidabilité, algèbre relationnel, syntaxe et sémantique des identités, etc.). Un des formalismes référentiels de gestion des configurations est basé sur le principe d'applicabilité. La distinction systématique des versions pour chaque train de modifications fiabilise leur traçabilité mais entraîne une prolifération des versions et une spécificité des productions qui en deviennent complexes et coûteuses. Un autre

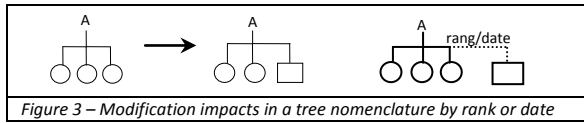
¹⁷ On englobe sous cette désignation l'architecture système (Le Moigne, 1977)

formalisme référentiel est, a contrario, basé sur le principe de standardisation. Ce formalisme demande une étude combinatoire irréalisable pour en satisfaire toutes les conditions. Une troisième alternative consiste à reconcevoir l'architecture des systèmes (leur décomposition) – en termes de modularité, d'environnement intégré, d'intégration continue, etc. – en sorte de préserver l'invariance des sous-systèmes (Raymond, 1998). Mais ce formalisme ne fait que déplacer la gestion de la complexité combinatoire des configurations à l'échelle des interfaces et des sous-systèmes (Bellagio & Milligan, 2005; Djezzar, 2003; Rahmani & Thomson, 2011; ElMaraghy, 2019). A l'instar des systèmes aux architectures décentralisées multi-couches qui séparent les couches stables (hardware) des couches évolutives (software), voire fortement évolutives (communication) (Rousseau, 2005). Les formalismes proposés par les PLM, les ERP et les bases de données relationnelles, sont aussi combinés afin d'offrir des complémentarités possibles, mais cela ne résout pas davantage la problématique de la gestion des configurations car leur fondement commun est une identité des systèmes de nature universelle. Ce fondement n'est pas tenable dans un monde qui change perpétuellement. Un nouveau fondement est incontournable : une identité des systèmes de nature conditionnelle. Cela ouvre un nouvel espace stratégique sur l'échiquier des offres.

4.1 Faire l'appoint des modifications pour l'exemplaire à produire et principe d'applicabilité

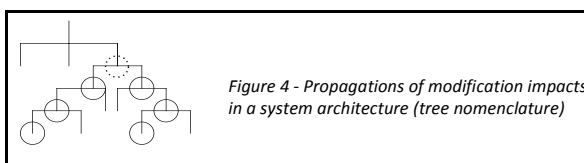
Supposons, à un instant initial, que la configuration de l'objet (définition, identification, structure) soit partagée par l'ensemble des acteurs et tienne lieu de référentiel industriel d'origine. Cette définition évolue sous l'effet de l'innovation (amélioration des performances ou de la qualité, etc.). Ces modifications sont instruites et classées en fonction des impacts générés. Une première manière de gérer les états successifs de la définition est d'en mesurer les écarts par rapport à l'original en recensant les modifications survenues. Dans cette logique, l'objet technique est défini avec sa définition d'origine, augmentée des modifications applicables pour des exemplaires donnés. C'est ainsi que peuvent se coordonner les différents intervenants (Giacomoni & Sardas, 2011). L'appoint des modifications doit être fait à compter d'une date, d'un contrat client, ou d'un rang donné (numéro d'exemplaire de l'objet) généralement négocié(e) en fonction des stocks existants pour limiter les obsolescences. L'exemple ci-après illustre bien ces pratiques : *“Réglage moteur, modification d'une porte ou d'un câble électrique, chaque semaine, (...) une cinquantaine de nouvelles modifications. Minutieuses et rigoureuses, ces opérations d'analyse d'impact sur la documentation applicable, de cas d'emploi, de simulation de résultat et de synthèse s'exécutent très lentement en comparaison à d'autres industries. Normal lorsque l'on sait que chaque modification (...) équivaut à un changement sur le seuil de tir de Kourou et surtout sur le lanceur. Et vu que ce dernier est estimé à 130 millions d'euros, mieux vaut ne pas se tromper. Avec, fin février 2007, 25 vols commerciaux réussis pour seulement 4 échecs, Ariane 5 vise l'excellence.”* (Fodor, 2008). Cette philosophie gestionnaire est essentiellement sécuritaire et convient uniquement pour des productions à l'unité ou en très petites séries. Impossible de reproduire en série dans de telles conditions. Du reste, cette pratique s'accommode mal des contraintes de dépannage. Un élément ou un sous-système ne peut être considéré séparément du système. Pour spécialiser les productions, accroître la réactivité, assurer le réassort, reproduire en série les différents éléments, la condition préalable est donc une définition gérable de manière autonome de chaque élément et sous-système par rapport à l'ensemble du système. Il faut dans cette optique que la définition de chaque élément ou sous-système précise quelles modifications lui sont appliquées et qu'un ensemble d'éléments ou sous-systèmes modifiés incorpore exactement les modifications prévues à compter d'un rang (d'une date, etc.) donné

(Cf Figure 3). C'est une manière de gérer les configurations dans des environnements référentiels d'après le principe d'applicabilité.



4.2 Renommer ce qui diffère dans la composition reproduite et principe d'interchangeabilité

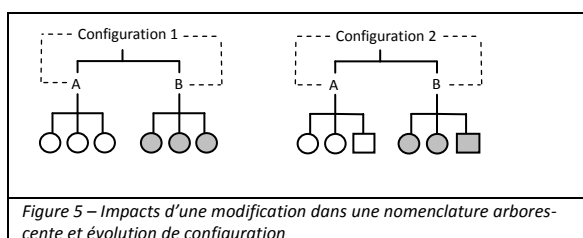
Par rapport au formalisme précédent, où chaque objet était identifié dans sa globalité, une identification unique (codification normalisée alphanumérique) est cette fois attribuée à chaque élément et sous-système. Elle est commune avec sa définition (ensemble des dossiers de définition, données techniques, plans, modèles CAO, etc.) et suffisante pour les reproduire. Toute modification se répercute physiquement par un marquage des objets, ainsi que sur les documentations associées et partagées par l'ensemble des acteurs. Il est évident que la distinction systématique des configurations successivement générées est inapplicable si les scénarii se complexifient. Une façon de parer à une propagation des changements d'identification dans une architecture de système est le principe d'interchangeabilité. Qu'il s'agisse d'éléments, de sous-systèmes ou de systèmes, gérer leurs évolutions d'après ce principe, implique de les différencier en leur attribuant de nouvelles identifications, à moins qu'ils ne soient considérés comme interchangeables, auquel cas ils conservent leur identification. La population des éléments, sous-systèmes et systèmes s'enrichit donc de nouveaux arrivants nés de l'amélioration technologique (performance ou qualité) de leurs aînés sous l'effet de l'innovation. Les relations entre systèmes, sous-systèmes modifiés et éléments modifiés sont restructurées en rendant compte des comportements attendus dans les environnements référentiels (sachant qu'ils sont susceptibles eux-mêmes d'évoluer). Des comportements similaires reflètent des configurations interchangeables et des identifications inchangées. La conquête de l'interchangeabilité (identité et indiscernabilité) est un moyen d'absorber les effets de l'innovation (Saga & Zmud, 1996) en développant de nouveaux cas d'emploi et ainsi de standardiser une production en évitant une distinction systématique des générations successives d'éléments et d'ensembles (sous-systèmes ou systèmes) qui segmente les séries. Mais c'est un moyen radical - sans mémoire de composition (par rapport au principe d'une gestion par rang ou par date consistant à faire l'appoint des modifications affectant un système). Pour connaître l'antériorité d'un élément modifié, il faut pouvoir cheminer en sens inverse et retrouver dans sa composition, l'élément constitutif impacté qui a répercuté l'impact à son tour, ainsi de suite à chaque niveau. Mais une interchangeabilité efface toute trace des différences de composition ou d'architecture. Du reste, une interchangeabilité totale (non-prématurément statuée) exige une exploration combinatoire de l'analyse des impacts contraignant à étudier plus de configurations qu'il n'en faut produire (Cf Figure 4). Le formalisme basé sur le principe d'interchangeabilité ne prévoit pas de définition conditionnelle dans les PLM, les ERP ou les bases de données (bien que la conditionnalité soit couramment admise et gérée hors cadre).



4.3 Combiner les formalismes, partitionner les systèmes et principe de conditionnalité

Les formalismes que nous avons décrits, basés sur les principes d'applicabilité ou d'interchangeabilité, sont praticables lorsque les activités ont un rythme innovatif faible (définitions stabilisées, peu de modifications) et/ou lorsque les activités sont peu sériées (nombre d'exemplaires réduit). En somme, des activités n'exigeant pas d'analyse combinatoire pour gérer les configurations. Ces formalismes ne peuvent pas se combiner dans les PLM, les ERP et les bases de données relationnelles sans la plus grande précaution. Par définition en effet, le principe d'interchangeabilité est construit sur l'indifférence d'architecture (de composition) contrairement au principe d'applicabilité par rang/date. Les philosophies gestionnaires sont contradictoires et leur combinaison est porteuse de risques de mésinterprétation et d'erreur, nonobstant le surcroît de travail qu'une double gestion des configurations implique.

Les PLM, les ERP et les bases de données relationnelles les plus sophistiquées proposent de combiner les formalismes de manière consistante (non-contradictoire) en contrôlant le partitionnement des définitions (modularité) et l'identité conditionnelle des systèmes (conditionnalité). La modularité permet de partitionner la définition d'un système en fonction de l'intérêt stratégique que procure une architecture distribuée sur différents niveaux (métadonnées, modules, sous-ensembles ou nomenclatures ou sous-ensembles basés sur les matériaux). Les stratégies reposent sur les facteurs de complexité tels que la taille des systèmes, l'hétérogénéité technologique, le processus d'assemblage, le rythme de production et le rythme de renouvellement des systèmes sous l'effet de l'innovation (Boothroyd & Dewhurst, 2010; Raymond, 1998; Giacomoni & Sardas, 2014). Les conditions d'identité des systèmes sont gérées à travers un partitionnement stratégique – le tout, ses parties (sous-systèmes), leurs relations – et une syntaxe pour les identifier et les discerner dans leur environnement référentiel. Ces dernières sont normalisées (les références fabricants par exemple) généralement complétées de codes ou de codiciles (d'interchangeabilité ou de modèles) qui ne sont pas marqués physiquement sur les objets et n'existent que dans les mémoires informatiques (PLM, ERP, bases de données relationnelles). Les relations entre le tout et ses parties sont gérées à travers les évolutions de configurations [*baselines, photos, models*] assorties d'applicabilités conditionnelles [*classes, context-driven settings, restricted/full interchangeability*], afin de limiter la validité des interchangeabilités en fonction des environnements d'emploi [context-driven content] et donc des exemplaires concernés dans la chaîne d'assemblage (Cf Figure 5). Une interchangeabilité restreinte¹⁸ est ainsi classée en fonction de propriétés techniques caractéristiques [*Form Fit Function classes* basées sur des considérations physiques, fonctionnelles, de performance ou des spécifications] devant être satisfaites pour qu'un système d'une classe donnée puisse être substituable par un autre de la même classe (dite d'équivalence).



5. LIMITES OPERATIONNELLES ET THEORIQUES DES FORMALISMES EN USAGE POUR GERER LES IDENTITES CONDITIONNELLES

Dans des contextes industriels de hautes technologies sous pression innovante, les formalismes de gestion des configurations existants ne permettent pas un traitement entièrement automatisable des interchangeabilités conditionnelles et requièrent une intervention décisionnelle humaine (Brown, 2006; Bouikni et al. 2008; Hwang et al; 2009; Giacomoni & Sar-

¹⁸ Par exemple, à proximité des moteurs, seuls les produits résistants aux hautes températures sont utilisables (Champion Aerospace LLC, Service Bulletin S.B. CH53536-1-74-001, Interchangeability and Intermixability of Parts, December 19th 2008)

das, 2014). *“The parametric definition of common validity is critical. It makes it easier to ground the validity of an object in the value of some of its attributes (...). We are aware that this operation mode [a specific implementation of a change management strategy] is not suitable to all businesses but it does provide a satisfactory base for most implementations (...) In terms of product structure management, many systems seem to provide equivalent basic functions. Still, it is important to verify if these functions are available in the delivered basic solution or if the latter is more like a toolbox, which requires the customer to build their own applications of product structure management. (...) It is possible to define product configurations and manage their validity. (...) “the process” must be manually connected with each subset structure and each related document in the product structure. In some cases [complex products with many variants], the relations with the subsets used become confusing, which makes it harder to use the system (...)”* (CIM Data, 2002). Plus récemment: *“Huge progress was done by software [Configuration Management] tools in the last two decades. Unfortunately, PLM systems are trailing behind in their capability to track changes, manage revisions and support configuration management principles. Especially when it comes to a current state of modern product development. A product complexity combined with the complexity of communication in manufacturing created an unprecedented level of data management challenges PLM vendors will continue to face in the coming decade.”* (Shilovitsky, 2020)¹⁹. Se posent donc deux questions, celle de la coexistence de l’homme et de l’outil informatique, notamment de savoir quand l’intervention humaine peut être remplacée ou augmentée (Lanzolla et al., 2018) et celle de savoir comment repenser les fondements théoriques à l’origine des limitations fonctionnelles des formalismes proposés par les outils informatiques existants et ouvrir la possibilité de gérer des identités conditionnelles de systèmes. Ces questions concernent les solutions informatiques les plus avancés: *“human innovation management will be expected to work side by side with AI and machine learning algorithms in identifying and selecting opportunities as well as investigating what could be the organization’s next competitive advantage (...) However, our knowledge of AI’s limitations in the context of innovation is still quite sparse.”* (Haefner et al., 2021, p. 2).

Tant que cet obstacle n’est pas levé, toutes les potentialités des PLM, des ERP et des bases de données relationnelles ne peuvent être mises à profit. Ceci est bien confirmé par Rolls-Royce (*Configuration and Design Data Management Department – Trent 7000 & Airbus Projects*, Derby UK, en nov. 2015) et par Airbus (*Production – Logistics and Transportation*, Toulouse France, Juin – Déc. 2020). Et quand bien même cet obstacle à l’automatisation de la gestion des configurations serait-il levé, qu’un autre obstacle se dresserait encore : le discernement physique des objets. Pour un opérateur donné, seule l’identification peut être parlante ou bien silencieuse sur la composition/l’architecture de l’objet. Or cette composition/architecture n’est pas indifférente aux configurations/architecture de montage.

Il sera donc toujours impossible de calculer de manière automatisée le regroupement et la planification des besoins, ou de procéder à la recherche automatisée de l’ensemble des éléments sérialisés devant être remis à jour (retrofits), et de profiter pleinement des économies d’échelles. La survivance de différentes générations d’objets techniques se côtoyant dans les bases de données, les ateliers ou les parcs installés réclamera de les différencier (pour les éléments modifiés) et de les regrouper (pour les éléments invariants).

Les PLM, les ERP et les bases de données relationnelles ne pourront que présenter des solutions incomplètes ou indécidables aux acteurs désignés, à des fins d’analyses et de décisions, avec un recours à une main d’œuvre coûteuse mobilisée à l’examen des risques d’erreur d’assemblage ou de localisation. Sans être nécessairement préjudiciables à la qualité des maté-

¹⁹ The bloodbath of debates about FFF [From, Fit, Function] and revisions. OpenBOM. 29/02.

riels, les inspections déboucheront dans les mêmes proportions sur des fiches d'anomalies, des dérogations et parfois même de délicates substitutions d'éléments. Il sera toujours aussi délicat de créer une campagne de mise à niveau d'un parc complet de matériels quelles que soient leurs versions respectives, de localiser les éléments ainsi que leur mouvement entre les installations et les magasins, ou de gérer la disponibilité des éléments. Si ces outils informatiques sont indispensables à la gestion des relations entre des systèmes évolutifs et leurs environnements référentiels tout au long de leur cycle de vie, ils ne sont pas satisfaisants du fait de leur incomplétude. Même l'ingénierie des systèmes, définie comme *"a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods"* (IncoSE, 2022)²⁰ est impuissante, notamment en ce qui concerne les *"levels of complexity, change, uncertainty and variety"*. Il en va de même de l'A.I., en incluant les Machine Learning, Deep (Reinforcement) Learning, etc., car *l'univers des possibles* ne peut pas être connu à l'avance (Giacomoni, 2022).

Pour illustrer les limitations des formalismes de gestion des configurations de façon concrète et formelle, reprenons l'exemple issu de l'aéronautique présenté en section 2 : pour rappel, sur un Airbus A330, une configuration non-conforme d'une des trois installations informatiques de contrôle de vol a provoqué un atterrissage violent, au point de nécessiter le remplacement du train principal. Considérons le système S partitionné en sous-systèmes – parmi lesquels trois installations informatiques de contrôle de vol (A, B and C) – la configuration initiale ayant été validée dans des environnements référentiels. Les sous-systèmes étant susceptibles d'évoluer au fil du temps sous l'effet de l'innovation, considérons que le sous-système A doit incorporer un élément modifié. Si l'identification est inchangée, le principe d'applicabilité sera inopérant notamment en cas d'opération de maintenance. Il sera impossible de savoir si le sous-système de remplacement incorpore bien l'élément modifié ou pas. Renommer le sous-système après chaque modification d'élément serait impraticable car les concepteurs devraient alors renommer aussi le système pour conserver la trace de la modification opérée. Et ainsi de suite pour toute modification survenant à un niveau intermédiaire. En vertu du principe d'interchangeabilité, l'identification du sous-système restant inchangée, il sera impossible de filtrer par date ou par rang les relations de composition interne entre le sous-système, l'élément original et l'élément modifié. Ceci a déjà été expliqué. Pour appliquer le changement (à une série spécifique ou à partir d'une date donnée), il convient d'activer la relation avec l'élément modifié et désactiver celle avec l'élément d'origine. A ce stade, deux configurations interchangeables C1 et C2 sont validées pour le système S, celle d'origine et celle incorporant l'élément modifié. Considérons à présent que ce scénario se rejoue pour un autre sous-système B. Un tel scénario générerait une nouvelle configuration C3 validée pour le système S et couplant ensemble le sous-système modifié A et sous-système modifié B. La configuration C3 est interchangeable avec la configuration C2 (et par conséquent avec la configuration C1 également). L'interchangeabilité est ainsi valable entre les configurations C1, C2 et C3 successivement générée. Une configuration C4 n'a cependant pas été validée, celle couplant le sous-système A d'origine et le sous-système B modifié. L'exemple de l'aéronautique illustre les conséquences qu'entraînent un usage inadéquat (configuration non-validée). Pour garantir une interchangeabilité totale dans le temps, toutes les combinaisons devraient être validées. Mais cette combinatoire n'est pas praticable pour les raisons déjà exposées. Seule une interchangeabilité conditionnelle est possible mais les identifications ne délivreront alors qu'une information incomplète. C'est la raison pour laquelle Airbus et l'European Aviation Safety Agency

²⁰ International Council on Systems Engineering

ont alerté les opérateurs de maintenance sur le respect des seules combinaisons autorisées par le constructeur.

L'exemple déroule des scénarii dans un environnement référentiel stable. Si ce dernier change aussi, comme nous l'avons expliqué en section 3, une relation plus générale doit être élaborée (à l'instar du poids volumique ou densité), en sorte que l'interchangeabilité soit de nouveau décidable tout en considérant les deux environnements possibles (celui d'origine et le nouveau). L'interchangeabilité est alors étendue à une nouvelle propriété (de même que la propriété poids volumique étend la propriété poids) afin de rendre compte des comportements observables indifféremment dans l'un ou l'autre environnement (par exemple dans tous fluides). L'exemple de la formation des pilotes (section 2) initialement prévue pour l'environnement du Boeing 737 à un environnement étendu au Boeing 737 Max (équipé d'un nouveau système automatique) en donne une illustration.

Il faut donc considérer les configurations C1, C2 et C3 comme interchangeables de manière conditionnelle dans leurs environnements référentiels (l'interchangeabilité n'est valide qu'en regard de propriétés spécifiques).

Il pourrait être tentant de penser que si les formalismes de gestion des configurations s'accommodent mal des identités conditionnelles, les obstacles sont purement opérationnels et non théoriques. Mais il suffit de ne pas perdre de vue que toute construction scientifique (incluant l'arithmétique, en d'autres termes ce qui peut se faire avec les codes binaires 0 et 1) ne peut être à la fois consistante (non-contradictoire) et complète (basée sur un état référentiel des connaissances), qu'il en va de même de la syntaxe et de la sémantique qui lui sont associées, pour comprendre que l'inaccessibilité des identités conditionnelles n'est pas seulement pratique mais aussi théorique.

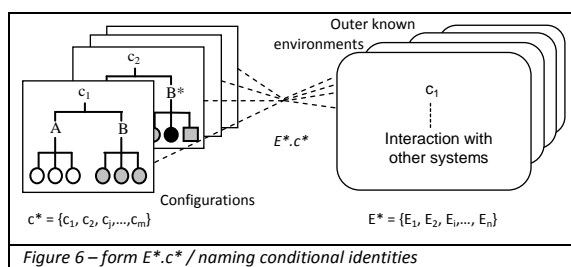
6. PROPOSITION D'UNE SYNTAXE ET D'UNE SEMANTIQUE ALTERNATIVES BASEES SUR UNE LOGIQUE NON-STANDARD

En somme, les relations d'identité sont une façon de formaliser, dans des environnements référentiels, la similarité de comportement de systèmes présentant des configurations différentes. Les propriétés des systèmes servent à caractériser les configurations relationnelles (relations de dépendance entre systèmes et sous-systèmes) pour rendre compte des comportements apparents dans des environnements référentiels, afin de statuer sur les invariances (dans le temps) ou les similarités (dans l'espace). L'identité des systèmes est nécessairement conditionnelle à un état des connaissances et devrait pouvoir être identifiable via une stratégie de dénomination appropriée, restituant toute l'information pertinente pour que les conditions de validité soient toujours réunies. Ces dernières ne sont généralement mémorisables via les syntaxes et les sémantiques des normes de codification internationales non-prévues à cet effet, quels que soient les mondes référentiels, logiciel, chimie, biologie ou génétique. A titre d'exemple, les différentes règles d'identification des substances décrites en 2017 dans le document d'orientation²¹ destiné à aider l'industrie, les états-membres et l'European Chemicals Agency souligne l'incomplétude de l'identification: *"A substance is completely identified by its chemical composition i.e. the chemical identity and the content of each constituent in the substance (...) for certain substances [Substances of Unknown or Variable composition, Complex reaction products or Biological Materials] it is not feasible or not adequate (...). In those cases, other or additional substance identification information is required (p.28) (...) It is therefore considered that petroleum substances are manufactured to meet the performance specifications rather than specifications about the composition"* (p.39).

²¹ Guidance on identification and naming of substances under REACH and CLP, ECHA, version 1.3, Feb. 2014.

En 2022, une annexe du document d'orientation étendait les recommandations aux "nanofformes" [substance naturelle ou artificielle à l'échelle du milliardième de mètre] : "complete list of use (...) characterization parameters (...) chemical composition (...) After the identification (...) all co-registrants must discuss and agree on the registration strategy, and decide on (...) single nanofforms or via sets of similar nanofforms, or a combination of these two" (p.23, p.39). Les identifiants complémentaires peuvent être de différents types selon les domaines : origines ou espèces en biologie, propriétés, etc.).

Les langages scientifiques proposent une grande variété de stratégies syntaxiques et sémantiques possibles pour formaliser les identités via des identifiants. "In computer languages we encounter such statements as 'J=J+1' which everybody seems to understand" (Jaynes, 2003, p.17). E.F. Codd également a abordé cette question dans un article seminal de 1970 sur les bases de données relationnelles: "One important effect that the view adopted toward data has on the language used to retrieve it is in the naming of data elements and sets (...) The simple form - R.d - will often be adequate (...) while it is not the purpose of this paper" (Codd, 1970, p.380-381). Le domaine d est un ensemble dénombrable de valeurs, caractérisable par un nom, tandis que R est un sous-ensemble du produit cartésien d'une liste de domaines caractérisable par un nom. D'après ce langage scientifique, nous pourrions considérer c comme un ensemble dénombrable de configurations possibles et E comme un sous-ensemble dénombrable d'environnements référentiels. La forme simplifiée $E.c$ serait bien adaptée pour nommer des identités conditionnelles. Sous réserve de bien considérer E comme des classes d'équivalence d'environnements référentiels interchangeables et c comme des classes d'équivalence de configurations (éléments, sous-ensembles, systèmes) interchangeables. A défaut de quoi, la forme $E.c$ serait tenue d'évoluer en cas de changements affectant un environnement ou une configuration, contrevenant alors au principe même d'une identité. La forme simplifiée $E*.c*$ en lieu et place de la forme $E.c$ pour désigner le conditionnement mutuel de ces classes d'équivalence de configurations interchangeables $c*$ et d'environnements référentiels interchangeables $E*$, présente l'avantage d'écarter toute ambiguïté (Cf Figure 6).



Cette forme régénérative de dénomination permet aux concepteurs de gérer des identités "limitées" reflétant la rationalité "limitée" qui est à l'oeuvre (Simon, 1996) et permettant d'anticiper l'identité du comportement des systèmes tenus pour interchangeables dans des environnements référentiels, tout en sachant bien que cette identité sera reconsidérée dans de futurs environnements, pour l'heure encore inconnus. La forme de dénomination $E*.c*$ – permet en outre, à l'ensemble des acteurs, une formalisation explicative de l'incomplétude et de l'indécidabilité immanente à tout état des connaissances. Une extension des connaissances est donc incontournable pour reconcevoir des classes d'équivalence conditionnelles et les identifier sous la forme $E*.c*$, au gré des évolutions de configurations et/ou des environnements référentiels. Ce formalisme clarifie aussi comment l'intégration de nouvelles connaissances et l'innovation sont reliées (Aleksander & Morton, 2014). Il n'y a rien d'étonnant à ce que des environnements industriels instables conduisent une fois de plus à l'émergence d'une stratégie de conception non-standard (Gavetti & Rivkin, 2007) pour que les

organisations travaillent de manière performante (Woodward, 1965; Lawrence & Lorsch, 1967).

Il est temps à présent d'ouvrir la discussion stratégique sur l'extension des fondements scientifiques des solutions informatiques et sur les nouveaux avantages compétitifs qu'ils seraient en mesure d'apporter (Schwenck, 1988; Esterhuizen et al., 2012; Popadiuk & Choo, 2006).

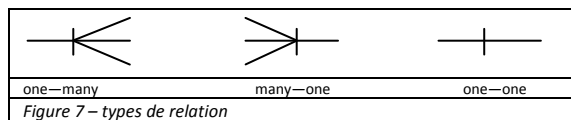
Quand les évolutions surviennent, les concepteurs doivent redéfinir les relations de dépendance et d'identité entre les sous-systèmes, les systèmes et les environnements référentiels. A dessein, ils doivent explorer la combinatoire des scénarii possibles et caractériser à l'aide de propriétés, les comportements à attendre des systèmes. Ils raisonnent donc en référence à un état initial des connaissances pour décrire un système, la manière dont il interagit et se comporte dans un environnement référentiel à l'aide de propriétés existantes. Ils raisonnent aussi en référence à un état étendu des connaissances pour décrire le système, la manière dont il se comporte et interagit dans des environnements différents, à l'aide de propriétés étendues (à l'instar des propriétés 'poids' dans l'air et 'poids volumiques' dans tous fluides). De ce fait, l'état des connaissances initial est englobé dans l'état des connaissances étendu. Une telle transformation est indescriptible en logique standard (Cori & Lascar, 1993; Turing & Girard, 1999; Girard, 2004). La logique est devenue la science des structures générales se tenant entre un sujet pensant et un objet à connaître et peut se définir comme la recherche des conditions formelles d'une vérité (Klein, 2013, p.21). Les concepteurs peuvent définir les objets de manière consistante (non-contradictoire) à l'aide de propriétés caractérisant les relations de dépendance et d'identité, en rendant compte formellement de la manière les objets se comportent et interagissent dans un environnement référentiel. Pour ce faire, ils se réfèrent nécessairement à un tel environnement pour statuer sur des valeurs de vérité ('vrai' ou 'faux', '1' ou '0'), à propos des phénomènes et des comportements observés. Par exemple, il est formellement correct d'un point de vue logique, de dire "*si A est vrai et que B inclus dans A est vrai, alors B est vrai*". En commençant pas "*si ...*", cette proposition est bien conditionnelle. Pour statuer sur une valeur de vérité à propos de A ("*A est vrai*") ou à propos de B ("*B inclus dans A est vrai*"), les concepteurs doivent se référer à un environnement référentiel dans lequel A et B ont un sens, comme l'a décrit A. Tarsky avec sa notion de "*truth-in-a-mode*" (1969; 1983). De cette façon, ils tiennent un tel environnement référentiel pour universel, jusqu'à le définir comme tel, un espace contenant tout, dénommé "*univers*", dans lequel la logique opère. Dans l'exemple de la section 3, l'universalité de la propriété "*poids*" impliquée dans la relation de dépendance "*force d'attraction qu'une masse exerce sur une autre*", est décidable en référence "*à l'air ambiant, à l'équateur et au niveau de la mer*" par exemple. L'expression fait d'ailleurs explicitement référence à une "*constante universelle*" ($G=6,67408 \cdot 10^{-11} \text{ m}^3 \cdot \text{kg}^{-1} \cdot \text{s}^{-2}$ soit $9,81 \text{ m} \cdot \text{s}^{-2}$ à la surface de la Terre selon le comité de données pour la science et la technologie). En référence à d'autres situations possibles (sous l'eau ou au-dessus du niveau de la mer, aux pôles, etc.), la valeur de vérité (booléenne) est indécidable (information incomplète) tant que le fluide considéré n'est pas déterminé (section 3). Il est impossible par exemple de dire si le comportement apparent des corps s'explique par une identité de poids, sans savoir si l'expérience est faite dans l'air ou dans l'eau. En outre, la constante universelle varie ($9,78 \text{ m} \cdot \text{s}^{-2}$ aux pôles et de $9,83 \text{ m} \cdot \text{s}^{-2}$ à l'équateur).

Une propriété tenue initialement pour universelle devient une propriété situationnelle à l'issue d'un processus d'extension des connaissances et de l'univers référentiel. A l'instar de la propriété "*poids*", tenue pour universelle en raisonnant dans l'air, en référence à l'état des connaissances initial et qui devient situationnelle en raisonnant dans tous fluides, en référence à

l'état des connaissances étendu. Une nouvelle propriété "poids volumique" est dès lors tenue pour universelle et ainsi de suite.

La logique standard n'est pas en mesure de définir l'universalité de manière consistante (non-contradictoire) au cours d'un processus d'extension. En effet, la logique procède conformément à une axiomatique (Tarsky, 1983). Les axiomes sont posés sans démonstration, tandis que les théorèmes résultent de démonstrations (une fois les implications ou les équivalences établies, etc.). Les propositions (axiomes ou théorèmes) ont une valeur de vérité (vrai ou faux). Aussi longtemps que la logique fonctionnera à partir d'axiomes, les valeurs de vérité seront sujets à débat. L'absence d'axiomes conduirait à une régression infinie, chaque proposition devant être définie à partir de proposition plus primitive encore, etc. Et il n'y a pas de logique-mère pouvant engendrer toutes les autres, avec leurs axiomatiques et leurs extensions. Aucun formalisme²² mathématique constructible, ne semble être en mesure d'aider les concepteurs à définir des propriétés étendues à plusieurs univers référentiels et impliquées dans des métarelations de dépendance et d'identité. Ces métarelations préservent du reste les relations de dépendance et d'identité qui préexistaient au sein de chaque univers.

Il faut bien admettre que le raisonnement par extension est à l'oeuvre quand plusieurs univers coexistent (nous avons choisi de le qualifier de *multivers*). Toute (méta)relation construite comme un produit Cartésien²³ [R x R x R ...] procède de la sorte, puisqu'il y a réplique des univers R (nombres réels ou complexes ...). Dans la mémoire des concepteurs, où se construisent les métarelations, le processus est connu sous le terme de d'abstraction (Fine, 2002; Tennant, 2004) ou d'intuition (Calabrese & Costa, 2015; Kahneman & Smith, 2002). Il aide à comprendre comment, dans la mémoire d'un ordinateur ou dans un cloud, le *multivers* pourrait être formalisé et comment des métarelations de type *one-many* pourraient être générées, sachant que les fonctions ne peuvent rendre compte que de relations de type *one-one* ou *many-one* (Figure 7).



Les bases de données relationnelles se heurtent à cette difficulté – appelée *point of ambiguity* – depuis de nombreuses années (Codd, 1970, pp.382-384).

Pour les systèmes virtuels qui n'existent que dans la mémoire d'un ordinateur ou dans un cloud, même si le processus d'extension n'est pas entièrement formalisable, il est néanmoins possible de gérer des représentations codées formalisant des relations d'identités conditionnelles (Cf Dassault système, Siemens, PTC, SAP, Microsoft, etc.). En revanche, pour les systèmes matériels, seules les dénominations (physiquement marquées) permettent aux décideurs de discerner ceux qui sont interchangeables et appartiennent de ce fait à la même classe d'équivalence.

Bien qu'abstraite et apparentée à celle d'E. Codd (*R.d.*) pour les éléments de données et les ensembles, la forme $E^*.c^*$ est cependant relative à des classes d'équivalence de configurations et d'environnements référentiels (i.e. interchangeabilité conditionnelle). Cette syntaxe et sa sémantique sont semble-t-il, la seule réponse possible pour gérer de manière constructible les identités conditionnelles de systèmes en constante évolution. Elles constituent une voie explo-

²² Par exemple, la théorie des ensembles (Jech, 1978) est un formalisme axiomatique très général, représentant tout système à partir d'ensembles définis par des propriétés distinctives non-universelles. L'ensemble de tous les ensembles n'étant pas lui-même définissable comme un ensemble, mais comme une collection définie par d'une propriété universelle.

²³ A Cartesian product is defined from many universes U_1, U_2, \dots, U_n as the set of distributions of possible values (n-tuples) respectively selected one by one in each universe. U_1, U_2, \dots, U_n can be copies of a same universe U (a proof that many universes have to coexist).

ratoire sérieuse mais pas encore exploitable en l'état, car les normes internationales imposent leurs propres syntaxe et sémantique. Le problème de l'interchangeabilité n'est donc toujours pas résolu. Des raisons à la fois opérationnelles et théoriques sont donc bien à l'origine des difficultés rencontrées par les PLM, les ERP et les bases de données relationnelles pour gérer informatiquement les interchangeabilités conditionnelles, sans intervention humaine pour décision ou arbitrage. C'est un résultat important qui ne pouvait pas être attendu de la littérature. Il soulève la question de la coexistence de l'homme et de l'outil informatique, notamment de savoir quand l'intervention humaine peut être remplacée ou augmentée. Le sujet fera l'objet de recherches à venir.

7. CONCLUSION

Les prédictions concernant les comportements apparents des systèmes sont conditionnelles à un état des connaissances et des expériences réalisées. La gestion des configurations, du tout en relation avec ses parties, devrait être logiquement formalisées et labellisées comme tel. Mais les normes internationales ne sont pas prévues à cet effet. Des progrès devraient être accomplis pour lever ce point de blocage. Dans l'intervalle, la prudence est de rigueur en ce qui concerne l'identité des systèmes (la décision demeurant essentiellement humaine). Ce challenge a toujours été au coeur de l'histoire de l'industrie (Roser, 2016).

L'environnement référentiel délivre toute l'information nécessaire sur le comportement apparent des systèmes qui se formalise à travers des relations de dépendance ou d'identité et des propriétés correspondantes (universelles ou spécifiques pour caractériser respectivement l'environnement référentiel ou les systèmes). La logique standard est opérante si l'environnement référentiel est fixe. Mais s'il change, elle n'est plus en mesure de rendre compte du processus d'extension des relations de dépendances et d'identité, ni des propriétés correspondantes, qui exigent de comparer les comportements des systèmes dans des environnements référentiels multiples. L'extension de la logique standard est nécessaire pour pouvoir définir les notions d'universalité et d'identité conditionnellement à un état des connaissances référentiel et les gérer informatiquement au sein de PLM, d'ERP ou de bases de données relationnelles (et plus largement en recourant à l'IA, machine learning, etc.). Ce pas de côté vers une nouvelle position scientifique fait échos aux travaux d'E. T. Jaynes ("extended logic of science", 2003) et d'H. Simon ("sciences of the artificial" (1996). "*As we succeed in broadening and deepening our knowledge, theoretical and empirical, about computers, we discover that in large part their behavior is governed by simple general laws, that what appeared as complexity in the computer program was to a considerable extent complexity of the environment to which the program was seeking to adapt its behavior. This relation of program to environment opened up an exceedingly important role for computer simulation as a tool for achieving a deeper understanding of human behavior.*" (p.21).

D'un point de vue opérationnel, la syntaxe et la sémantique proposées ($E^*.c^*$) permettraient de gérer des classes d'équivalence conditionnelles de configurations et d'environnements référentiels, si elles ne se heurtaient aux normes internationales de codification. Cette recherche pose un éclairage nouveau sur la question de la coexistence de l'homme et de l'outil informatique, des fondements scientifiques d'une décision humaine augmentée. Le sujet fera l'objet de recherches à venir.

8. BIBLIOGRAPHIE

- Abramovici, M., & Aidi, Y. (2013). Next Generation Product Lifecycle Management (PLM). In: Fathi M. (eds) *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*, Springer, Berlin, Heidelberg. DOI:10.1007/978-3-642-34471-8_12.
- Abramovici, M. (2007). Future Trends in Product Lifecycle Management, in *The Future of Product Development*, Springer Berlin Heidelberg. DOI:10.1007/978-3-540-69820-3_64.
- Alavi, M., & Leidner, D.E. (2001). Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues, *MIS Quarterly*, 25(1), 107-136. DOI:10.2307/3250961.
- Aleksander, I., & Morton, H. (2014). Aristotle's Laptop: The Discovery of our Informational Mind, *International Journal of Machine Consciousness*, 06(1), World Scientific Publishing Company, 49-53. DOI:10.1142/S1793843014400083.
- Alvesson, M., & Sandberg, J. (2020). The Problematizing Review: A Counterpoint to Elsbach and Van Knippenberg's Argument for Integrative Reviews, *Journal of Management Studies*, 57(6), 1290-1304. 10.1111/joms.12582.
- Amann, K. (2002). Product lifecycle management: empowering the future of business, CIM Data, Inc.
- Andersson, M., Lindgren, R., Henfridsson, O., (2008), Architectural knowledge in inter-organizational IT innovation, *Journal of Strategic Information Systems*, 17(1), 19-38. DOI: 10.1016/j.jsis.2008.01.002
- André, J., & Hudrisier, H. (2002). Unicode, écriture du monde ?, *Revue Document Numérique* (special issue), 6(3-4), Hermès Lavoisier. ISBN : 2746205947.
- Batenburg, R., Helms, R., & Versendaal, J. (2005). The maturity of product lifecycle management in Dutch organizations: A strategic alignment perspective, *PLM'05: International conference on product life cycle management*, Lyon, France.
- Beauzamy, B. (2012). *Archimedes' Modern Works*, SCM (Ed.), Real Life Mathematics (Coll.), Broché, Paris.
- Bellagio, D. E., & Milligan T. J. (2005). *Software Configuration Management Strategies and IBM Rational Clearcase: A Practical Introduction*, IBM Press, USA. ISBN13 : 978-0321200198.
- Benbya, H., & Meissonier, R. (2007). La contribution des Systèmes de Gestion des Connaissances au développement de nouveaux produits : étude de cas d'une entreprise du secteur de l'industrie aéronautique, *Systèmes d'Information et Management*, 12(1). ISSN : 12260-4984.
- Bernard, A. (1996), *Contribution à la modélisation des produits et à l'intégration des technologies productrices dans un environnement multi-acteurs*, HDR, Université Nancy I.
- Boothroyd, G., Dewhurst, P., & W. A. Knight. (2010). *Product Design for Manufacture and Assembly Design for Assembly*, (3rd Ed.), (Manufacturing Engineering and Materials Processing) Hardcover. DOI: 10.1201/9781420089288.
- Bouikni, N., Rivest L., & Desrochers, A. (2008). A Multiple Views Management System for Concurrent Engineering and PLM, *Concurrent Engineering: Research and Applications*, 16(1). DOI: 10.1177/1063293X07084641.
- Bronsvoort, W.F., & Noort, A. (2004). Multiple-view feature modeling for integral product development, *Computer-Aided Design*, 36(10), 929-946. DOI: 10.1016/j.cad.2003.09.008.
- Brown, J. (2006). Managing Product Relationships: Enabling Iteration and Innovation in Design, *Business Value Research Series*, June, AberdeenGroup.
- Calabrese, A., & Costa, R. (2015). Strategic thinking and business innovation: Abduction as cognitive element of leaders' strategizing, *Journal of Engineering and Technology Management*, 38(October-December), 24-36. DOI: 10.1016/j.jengtecman.2015.06.001.
- Cartwright, R. (1971). Identity and substitutivity, in *Identity and individuation*, (1971), M. K., Munitz (ed.), New York University Press, pp. 119-133, reprinted in Cartwright R. (1987), *Philosophical essays*, pp.135-147. DOI: 10.2307/2025325.
- CIM Data, (2002). Etude de l'offre SAP pour le cPDM: my SAP PLM, Pilotage des nouveaux produits et processus grâce à la gestion du cycle de vie des produits., SAP.AG, Germany.
- Codd, E.F. (2001). A Relational Model of Data for Large Shared Data Banks. In: Broy, M., Denert, E. (eds) *Pioneers and Their Contributions to Software Engineering*. Springer, Berlin, Heidelberg. DOI : 10.1007/978-3-642-48354-7_4.
- Cori, R. & Lascar, D. (1993). *Logique mathématique II. Fonctions récursives, théorème de Gödel, théorie des ensembles, théorie des modèles*, 1st ed., Ed. Masson, Coll. Axiomes. ISBN13 : 978-

2100054534.

- Cyert, R., & March, J. G. (1963). *A Behavioral Theory of the Firm*. Prentice Hall. Englewood Cliffs NJ. DOI: 10.7202/1021288ar.
- Dambietz, F. M. (2022). *Performance simulation of modular product architectures by model-based configuration*, Springer Vieweg, Hamburg, Germany. DOI: 10.1007/978-3-662-64233-7.
- David, P. (2011). Multilingual Web-LT: Meta-data interoperability between Web CMS, Localization tools and Language Technologies at the W3C, World Wide Web Consortium (W3C), University of Limerick, Dec.
- Dibiaggio, L. (2012). The Governance of Knowledge integration, in L. Dibiaggio P-X. Meschi P-X. (Ed.), *Management in the Knowledge Economy - New Managerial Models for Success?*, Pearson, France. ISBN13 : 978-2744076169
- Djezzar, L. (2003). *Gestion de configuration*, Dunod, Paris. ISBN13 : 978-2-10-006914-9.
- ElMaraghy, W.H. (2019). Complexity in Manufacturing. In S. Chatti, L. Laperrière, G. Reinhart & T. Tolio (eds), *CIRP Encyclopedia of Production Engineering*. Springer, Berlin. Heidelberg. DOI: 10.1007/978-3-662-53120-4_6670.
- Esterhuizen, D., Schutte, C.S.L., & du Toit, A.S.A. (2012). Knowledge creation processes as critical enablers for innovation, *International Journal of Information Management*, 32(4), August, 354-364, 10.1016/j.ijinfomgt.2011.11.013.
- Fichman, R. G. (2004). Going beyond the dominant paradigm for IT innovation diffusion research: Emerging concepts and methods, *Journal of the Association for Information Systems*, 5(8), 314-355. DOI:10.17705/1jais.00054.
- Fine, K. (2002). *The Limits of Abstraction*, Oxford University Press, UK. ISBN13 : 978-0199533633.
- Feeney, M. (1999). *Digital Culture: Maximizing the Nation's Investment: a Synthesis of JISCO/NPO Studies on the Preservation of Electronic Materials*, London, National Preservation Office, UK.
- Fodor, X. (2008). Organiser les échanges de données techniques : ARIANESPACE ne travaille pas dans la précipitation, Dossier : Collaboration: Optimisez votre travail, *iTechnologie*, (2).
- Gable, G. (2010). Strategic information systems research: an archival analysis, *Journal of Strategic Information Systems*, 19(1), 3-16, DOI: 10.1016/j.jsis.2010.02.003.
- Gavetti, G., & Rivkin J. W. (2007). On the origin of strategy: Action and cognition over time, *Organization Science*, 18(3), May-June. DOI: 10.1287/orsc.1070.0282.
- Ge, Y., Jiang, G., & Ge, Y. (2013, 7-10 Dec.). Efficient Invariant Search for Distributed Information Systems, IEEE 13th International Conference on Data Mining, Dallas, USA. DOI: 10.1109/ICDM.2013.133.
- Giacomoni, G. (2022). Towards a general framework for (early-stage of) innovation shaped with A.I. to create and transform market offerings, *European Management Review*. DOI: 10.1111/emre.12492.
- Giacomoni, G., & Sardas, J.-C. (2014). Why innovation requires new scientific foundations for manageable identities of systems (Part II – Chap.4), in D. T'Eni & F. Rowe (Eds). *R&D Strategy and Operations - Innovation and IT in an International Context*, Palgrave MacMillan (Publisher). DOI: 10.1057/9781137336132_5.
- Giacomoni, G., & Sardas, J.-C. (2011). Pilotage des productions d'objets complexes dans l'industrie de l'espace : innovation intensive et sériation, *Revue Française de Gestion Industrielle*, 30(1). DOI : 10.53102/2011.30.01.807.
- Girard, J. Y. (2004). Linear logic in computer science, *London Mathematical Society Lecture Notes Series*, No 316, Cambridge University Press, USA. DOI: 10.1017/CBO9780511550850.
- Grieves, M. (2006). *Product Lifecycle Management: Driving the Next Generation of Lean Management.*, New York, NY: McGraw Hill, USA. ISBN13: 978-0071452304.
- Haefner, N., Wincent, J., Parida, V. & Gassmann, O. (2021). Artificial intelligence and innovation management: A review, framework, and research agenda. *Technological Forecasting and Social Change Elsevier*, 162, 120392. <https://doi.org/10.1016/j.techfore.2020.120392>
- Hwang, J., Mun, D. & Han, S. (2009). Representation and Propagation of Engineering Change Information in Collaborative Product Development using a Neutral Reference Model, *Concurrent Engineering: Research and Applications*, 17(2). DOI:10.1177/1063293X09105339.
- IBM, Dassault Systèmes. (2008). IBM and Dassault Systèmes: Business Process Accelerators for Systems Engineering – Integrated Product Development from needs identification through to final

product validation.

- Jaynes, E.T. (2003). *Probability Theory: The Logic of Science*, Cambridge University Press, USA. ISBN13 : 978-0521592710.
- Jech, T. (1978). *Set theory*, Academic Press, Switzerland. ISBN13 : 978-0123819505.
- Jiang, G., Chen, H., & Yoshehira, K. (2006). Modeling and tracking of transaction flow dynamics for fault detection in complex systems, *IEEE Transactions on Dependable and Secure Computing*, 3(4), 312–326. DOI: 10.1109/TDSC.2006.52.
- Kahneman, D., & Smith, V. (2002, 17 December). Foundations of Behavioral and Experimental Economics, *Advanced information on the Prize in Economic Sciences 2002*, The Royal Swedish Academy of Sciences, Swedish.
- Klein, E. (2013). *D'où viennent les idées (scientifiques) – Modélisation des imaginaires*, Editions Manucius, Paris. ISBN : 2845781571.
- Lanzolla, G., Lorenz, A., Miron-Spektor, E. Schilling, M. Solinas G. & Tucci C. (2018). Digital transformation: What is new if anything? *Academy of Management Discoveries*, vol. 4(3), 378–387. <https://doi.org/10.5465/amd.2018.0103>.
- Lautman, A. (2006). *Les Mathématiques, les Idées et le Réel Physique*, Vrin, Librairie Philosophique J., Paris.
- Lawrence, P., & Lorsch, J. (1967). *Adapter les structures de l'entreprise - Intégration ou différenciation*, French translation, Editions d'organisation, Paris. ISBN13 : 978-2-7081-1091-5.
- Mantripragada, R., & Whitney, D.E. (1999). “Modeling and Controlling Variation Propagation in Mechanical Assemblies using State Transition Models”, *IEEE Transaction on Robotics and Automation*, 15(1). DOI: 10.1109/70.744608.
- Merminod, V., & Rowe, F. (2012). How does PLM technology support knowledge transfer and translation in new product development? Transparency and boundary spanners in an international context, *Information and organization*, 22(4), 295-322, Elsevier, DOI: 10.1016/j.infoandorg.2012.07.002.
- Morshedzadeh, I., Ng, H.C.A., & Jeusfeld, M. (2021). Managing manufacturing data and information in product lifecycle management systems considering changes and revisions, *International Journal of Product Lifecycle Management*, 13(3), 244-264, Inderscience Publishers. DOI: 10.1504/IJPLM.2021.118041.
- Mostefai, S., & Batouche, M. (2005). Data integration in Product Lifecycle Management: an ontology-based approach, Geneve: Inderscience Enterprises Limited, PLM'05: International conference on product life cycle management, Lyon, France. ISBN : 0-907776-18-3.
- Noël, F. (2006). A dynamic multi-view product model to share product behaviors among designers: how process model adds semantic to the behavior paradigm, *International Journal of Product Lifecycle Management*, 1(4), Paris. DOI: 10.1504/ijplm.2006.011056.
- Pavlou, P. A., & El Sawy, O. A. (2010). The ‘Third Hand’: IT-enabled competitive advantage in turbulence through improvisational capabilities, *Information Systems Research*, 21(3), 443-471. DOI: 10.1287/isre.1100.0280.
- Prashanth, B.N., & Venkataram, R. (2017). Development of Modular Integration Framework between PLM and ERP Systems, *Materials Today: Proceedings*, 4(2), Part A, 2269-2278. DOI: 10.1016/j.matpr.2017.02.075.
- Pol, G., Jared, G., Merlo, C. & Legardeur, J. (2005). From PDM systems to integrated project management systems: a case study, *PLM'05: International conference on product life cycle management*, Lyon, France.
- Popadiuk, S., Choo, C.W. (2006). Innovation and knowledge creation: How are these concepts related?, *International Journal of Information Management*, 26(4), August, 302-312. DOI: 10.1016/j.ijinfomgt.2006.03.011.
- Quigley, J. M. (2019). *Configuration Management*, Second Edition, Routledge, 2nd Ed. DOI: 10.1201/9780429318337.
- Rahmani, K., & Thomson, V. J. (2011). Managing subsystem interfaces of complex products, *International Journal of Product Lifecycle Management*, 5(1), Paris. DOI: 10.1504/IJPLM.2011.038103.
- Raymond, E.S. (1998). *The Cathedral and the Bazaar*, O'Reilly, UK. ISBN13: 978-1607962281.
- Rehman, F.U. & Yan, X.-T. (2007). Supporting early design decision making using design context

knowledge, *Journal of Design Research*, 6(1-2). DOI: 10.1504/JDR.2007.015568

- Reix, R., Kalika, M., Fallery, B., & Rowe, F. (2011). *Systèmes d'information et management des organisations*, 6th ed., Vuibert, Paris. ISBN13 : 9782311012354.
- Roser, C. (2016). *Faster, Better, Cheaper in the History of Manufacturing: From the Stone Age to Lean Manufacturing and Beyond*, 1st ed. Productivity Press, UK. ISBN : 9781498756303.
- Rousseau, F. (2005). *Conception des systèmes logiciel/matériel: du partitionnement logiciel/matériel au prototypage sur plateformes reconfigurables*, HDR thesis, Université Joseph Fourier, Grenoble I.
- Schweitzer, F.M., Handrich M. & Heidenreich S. (2019). Digital transformation in the new product development process: The role of IT-Enabled PLM systems for relational, structural, and NPD performance. *International Journal of Innovation Management*, 23(7), 1950067 (2019). DOI: 10.1142/S1363919619500671.
- Senge, P.M. (2006). *The Fifth Discipline: The Art and Practice of the Learning Organization*, Random House Business (Eds), Broché, 2nd Ed, Doubleday/Currency, USA. ISBN13: 978-1905211203.
- Sharma, G.T., & Patil, R.K. (2015). Reviewing Product Lifecycle Management Through Definition, Structure, Analysis, Strategies, Implementation and Application (DSASIA), *International Journal of Engineering Research & Technology*, 3(17), NCERAME Conference Proceedings, Special Issue, India.
- Simon, H. A. (1996). *The science of the artificial*, Cambridge: MIT Press. ISBN13: 9780262193740.
- Sodirova, A., Evangelopoulos N., Valacich J.S. & Ramakrishnan T. (2008). Uncovering the Intellectual Core of the Information Systems Discipline, *MIS Quarterly*, 32(3), 467-482, DOI: 10.2307/25148852.
- Srivardhana, T., & Pawlowski, S.D. (2007). ERP systems as an enabler of sustained business process innovation: A knowledge-based view, *Journal of Strategic Information Systems*, 16(1), 51-69. DOI: 10.1016/j.jsis.2007.01.003Get.
- Stark, J. (2004). Product Lifecycle Management - 21st century Paradigm for Product Realization, *Decision Engineering Series*, Springer Verlag, Berlin. DOI: 10.1007/978-0-85729-546-0.
- Schwenck, C. R. (1988). The cognitive perspective on strategic decision making, *Journal of Management Studies*, 25(1), January, Bloomington, U.S.A, 41-55.
- Tarski, A. (1969). Truth and Proof, *Scientific American*, 220, 63-77. DOI: 10.1038/SCIENTIFICAMERICAN0669-63.
- Tarski, A. (1983). *Logic, Semantics, Metamathematics*, Corcoran J. (ed.), Hackett, Woodger H.J. (1st ed. & transl.), Oxford University Press. ISBN: 0-915144-76-X (pbk.)
- Tennant, N. (2004). A general theory of abstraction operators, *The Philosophical Quarterly*, 54(214), 105-133. DOI: 10.1111/j.0031-8094.2004.00344.x
- Turing, A. & Girard, J. Y. (1999). *La machine de Turing*, Editions du seuil, Paris. ISBN : 2020369281.
- Van der Ven, A.H. (1986). Central problems in the management of innovation, *Management Science*, 32(5), 590-607. DOI: 10.1287/mnsc.32.5.590
- Watts, F.B. (2015). *Configuration Management for Senior Managers, Essential Product Configuration and Lifecycle Management for Manufacturing*, Lavoisier, Paris. ISBN: 978-0-12-802382-2.
- Wiggins, D. (2001). *Sameness and Substance Renewed*, Cambridge University Press, USA. ISBN13: 978-0521454117.
- Winter, S. G., & Szulanski, G. (2001). Replication as strategy, *Organization Science*, 12(6), USA, 730-743. DOI: 10.1287/orsc.12.6.730.10084.
- Wise, J. (2019, March 14). The Boeing 737 Max and the Problems Autopilot Can't Solve, *The New-York Times*, USA.
- Woodward, J. (1965). *Industrial organization. Theory and practice*, Oxford University Press, USA. 1

