

Manifeste de la Fouille de Procédé

Wil van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, Andrea Burattin, Josep Carmona, Malu Castellanos, Jan Claes, Jonathan Cook, Nicola Costantini, Francisco Curbera, Ernesto Damiani, Massimiliano de Leoni, Pavlos Delias, Boudewijn van Dongen, Marlon Dumas, Schahram Dustdar, Dirk Fahland, Diogo R. Ferreira, Walid Gaaloul, Frank van Geffen, Sukriti Goel, Christian Günther, Antonella Guzzo, Paul Harmon, Arthur ter Hofstede, John Hoogland, Jon Espen Ingvaldsen, Koki Kato, Rudolf Kuhn, Akhil Kumar, Marcello La Rosa, Fabrizio Maggi, Donato Malerba, Ronny Mans, Alberto Manuel, Martin McCreesh, Paola Mello, Jan Mendling, Marco Montali, Hamid Motahari Nezhad, Michael zur Muehlen, Jorge Munoz-Gama, Luigi Pontieri, Joel Ribeiro, Anne Rozinat, Hugo Seguel Pérez, Ricardo Seguel Pérez, Marcos Sepúlveda, Jim Sinur, Pnina Soffer, Minseok Song, Alessandro Sperduti, Giovanni Stilo, Casper Stoel, Keith Swenson, Maurizio Talamo, Wei Tan, Chris Turner, Jan Vanthienen, George Varvaressos, Eric Verbeek, Marc Verdonk, Roberto Vigo, Jianmin Wang, Barbara Weber, Matthias Weidlich, Ton Weijters, Lijie Wen, Michael Westergaard, and Moe Wynn

IEEE Task Force on Process Mining^{***}

<http://www.win.tue.nl/ieetfpm/>

Résumé Les techniques de fouille de procédé sont capables *d'extraire des connaissances à partir des événements de traces d'exécution* communément disponibles dans les systèmes d'information actuels. Ces techniques offrent de nouveaux moyens pour *découvrir, surveiller et améliorer les procédés* dans une variété de domaines d'application. Il y a deux principaux moteurs pour l'intérêt croissant pour la fouille de procédé. D'une part, de plus en plus d'événements sont enregistrés fournissant, par conséquent, des informations détaillées sur l'histoire du procédé. D'autre part, il y a un besoin pour améliorer et soutenir les procédés métier dans des environnements concurrentiels et en évolution rapide. Ce manifeste est créé par le Groupe de travail IEEE sur la fouille de procédé et vise à promouvoir le thème de la fouille de procédé. Par ailleurs, en définissant un ensemble de principes directeurs et énumérant les défis importants, ce manifeste espère servir de *guide pour les développeurs de logiciels, scientifiques, consultants, chefs d'entreprise, et utilisateurs finaux*. L'objectif est d'accroître la maturité de la fouille de procédé comme un nouvel outil pour améliorer la (re)conception, le contrôle et le support des procédés métiers opérationnels.

1 IEEE Task Force sur la Fouille de Procédé

Un *manifeste* est une “déclaration de principes et d'intentions” publique par un groupe de personnes. Ce manifeste a été rédigé par les membres et sympathisants du *groupe de travail (Task Force) IEEE sur la fouille de procédé*. L'objectif de ce groupe de travail est de promouvoir la recherche, le développement, l'éducation, la mise en œuvre, l'évolution et la compréhension de la fouille de procédé.

La fouille de procédé (process mining) est une discipline de recherche relativement jeune qui se trouve entre l'intelligence artificielle et la fouille de données, d'une part, et la modélisation et l'analyse des procédés d'autre part. L'idée derrière la fouille de procédé est de *découvrir, surveiller et améliorer les procédés réels* (c'est à dire, les procédés non supposés) par *l'extraction de connaissances à partir des événements des traces d'exécution* facilement disponibles dans les systèmes

*. La version originale est apparue dans le *BPM 2011 Workshops proceedings, Lecture Notes in Business Information Processing*, Springer-Verlag, 2011

**. Traduit au français par Walid Gaaloul

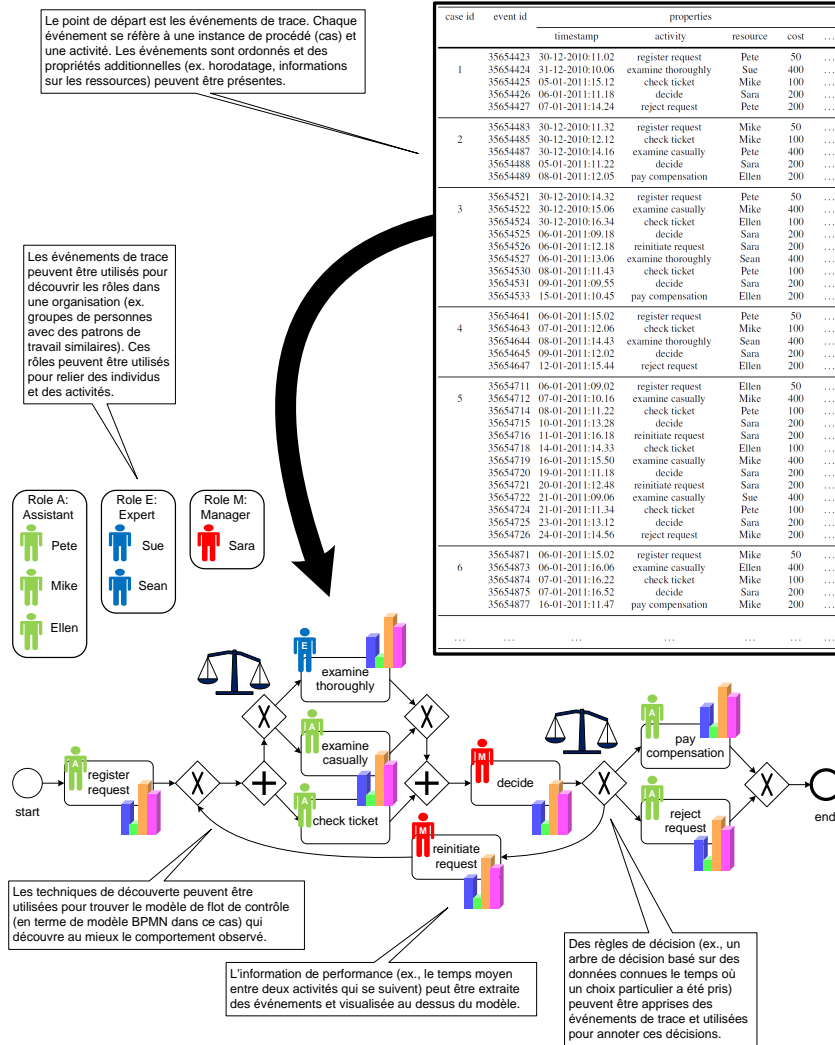


Figure 1. Les techniques de fouille de procédés extraient la connaissance des événements de trace pour découvrir, surveiller et améliorer les procédés.

(d'information) courants (voir Figure 1). La fouille de procédé comprend la découverte (automatisée) de procédés (c'est à dire, l'extraction de modèles de procédé à partir des événements de traces d'exécution), la vérification de la conformité (c'est à dire, la surveillance des écarts en comparant le modèle à ses traces d'exécution), la fouille du réseau social/d'organisation, la construction automatisée de modèles de simulation, l'extension de modèle, la réparation de modèle, la prédiction de cas et les recommandations basées sur l'historique des traces d'exécution.

La fouille de procédé fournit un pont important entre la fouille de données et la modélisation et l'analyse des procédés métiers. Sous les auspices de l'informatique décisionnelle ou l'intelligence métier (Business Intelligence BI) beaucoup de mots en vogue (buzzword) ont été introduits pour faire référence plutôt à des rapports métiers assez simples et des outils de tableaux de bord. La supervision des activités métier (Business Activity Monitoring BAM) fait référence aux technologies permettant la surveillance en temps réel des procédés métier. Le Traitement des événements complexes (Complex Event Processing CEP) se réfère aux techniques nécessaires pour traiter de grandes quantités d'événements, en les utilisant pour suivre, piloter et optimiser l'activité en temps réel. La gestion des performances métier (Corporate Performance Management CPM) est un autre mot à la mode pour mesurer la performance d'un procédé ou une organisation. Des approches de

gestion telles que *l'amélioration continue de procédés* (Continuous Process Improvement CPI), *l'amélioration des procédés métier* (Business Process Improvement BPI), *La qualité totale* (Total Quality Management TQM), et *Six Sigma* font partie également de ces technologies. Ces approches ont en commun la possibilité de voir s'il existe encore des améliorations possibles en mettant les procédés "sous microscope". La fouille de procédé est une technologie habilitante pour le CPM, BPI, TQM, Six Sigma, etc.

Alors que les outils BI et les approches de gestion telles que Six Sigma et TQM ont pour objectif l'amélioration des performances opérationnelles, par exemple, en réduisant le temps et les échecs d'exécution, les organisations mettent aussi l'accent sur la *gouvernance des entreprises*, les *risques*, et la *conformité*. Des législations telles que le Sarbanes-Oxley Act (SOX) et l'Accord de Bâle II illustrent l'accent mis sur les questions de conformité. Les techniques de fouille de procédé offrent un moyen pour vérifier plus rigoureusement la conformité, et assurer la validité et la fiabilité des informations sur les procédés de base d'une organisation.

Au cours des dix dernières années, les traces d'exécution sont devenues facilement disponibles et les techniques de fouille de procédé ont mûri. Par ailleurs, comme nous l'avons mentionné, les tendances de gestion liées à l'amélioration des procédés (par exemple, Six Sigma, TQM, CPI, et CPM) et à la conformité (SOX, BAM, etc.) peuvent bénéficier de la fouille de procédé. Heureusement, les algorithmes de fouille de procédé ont été mis en œuvre dans différents systèmes académiques et commerciales. Aujourd'hui, il y a un groupe actif de chercheurs travaillant sur la fouille de procédé et il est devenu l'un des "sujets d'actualités" de recherche dans la gestion des procédés métiers (Business Process Management BPM). Par ailleurs, il y a un énorme intérêt pour la fouille de procédé. De plus en plus d'éditeurs de logiciels ajoutent des fonctionnalités de fouille de procédé dans leurs outils. On peut citer quelques exemples de logiciels avec des capacités de fouille de procédé : ARIS Process Performance Manager (Software AG), Comprehend (Open Connect), Discovery Analyst (StereoLOGIC), Flow (FoursparK), Futura Reflect (Futura Process Intelligence), Interstage Automated Process Discovery (Fujitsu), OKT Process Mining suite (Exeura), Process Discovery Focus (Iontas/Verint), ProcessAnalyzer (QPR), ProM (TU/e), Rbminer/Dbminer (UPC), et Reflect|one (Pallas Athena). L'intérêt croissant dans l'analyse des traces d'exécution des procédés a motivé la création d'une Task Force sur la fouille de procédé.

La Task Force a été créée en 2009 dans le contexte de la Data Mining Technical Committee (DMTC) de la Computational Intelligence Society (CIS) de l'Institute of Electrical and Electronic Engineers (IEEE). La Task Force actuelle a des membres représentant les *éditeurs de logiciels* (par exemple, Pallas Athena, Software AG, Futura Process Intelligence, HP, IBM, Infosys, Fluxicon, Businesscape, Iontas/Verint, Fujitsu, Fujitsu Laboratories, Business Process Mining, Stereologic), des *cabinets de conseil*, des *utilisateurs finaux* (par exemple, ProcessGold, Business Process Trends, Gartner, Deloitte, Process Sphere, Siav SpA, BPM Chili, BWI Systeme GmbH, Excellentia BPM, Rabobank), et des *instituts de recherche* (par exemple, TU/e, University of Padua, Universitat Politècnica de Catalunya, New Mexico State University, IST - Technical University of Lisbon, University of Calabria, Penn State University, University of Bari, Humboldt-Universität zu Berlin, Queensland University of Technology, Vienna University of Economics and Business, Stevens Institute of Technology, University of Haifa, University of Bologna, Ulsan National Institute of Science and Technology, Cranfield University, K.U. Leuven, Tsinghua University, University of Innsbruck, University of Tartu).

Les objectifs concrets de la Task Force sont de :

- Rendre les utilisateurs finaux, les développeurs, les consultants, les chefs d'entreprise et les chercheurs conscients de l'état de l'art de la fouille de procédé,
- Promouvoir l'utilisation des techniques et des outils de fouille de procédé et stimuler de nouvelles applications,
- Jouer un rôle dans les efforts de normalisation de l'enregistrement des événements d'exécution,
- Organiser des tutoriels, des sessions spéciales, des ateliers, des panels, et
- Publier des articles, des livres, des vidéos et des numéros spéciaux de revues.

Depuis sa création en 2009, il y a eu diverses activités liées aux objectifs ci-dessus. Par exemple, plusieurs workshops et tracks spéciaux ont été (co-)organisés par la Task Force, comme, les workshops Business Process Intelligence (BPI'09, BPI'10 et BPI'11) et des tracks spéciaux dans des

conférences IEEE majeurs (par exemple CIDM'11). La connaissance a été diffusée via des didacticiels (par exemple WCCI'10 et PMPM'09), écoles d'été (ESSCaSS'09, ACPN'10, CICH'10, etc.), vidéos (www.processmining.org) et plusieurs publications, y compris le premier livre sur la fouille de procédé récemment publié par Springer¹. La Task Force a également (co-)organisé le premier Business Process Intelligence Challenge (BPIC'11) : une compétition où les participants devaient extraire des connaissances utiles à partir d'événements de trace importants et complexes. En 2010, la Task Force a également standardisé XES (www.xes-standard.org), un format de trace d'exécution standard qui est extensible et supporté par la bibliothèque OpenXES (www.openxes.org) et par des outils tels que ProM, XESame, Nitro, etc.

Le lecteur est invité à visiter <http://www.win.tue.nl/ieeetfpm/> pour plus d'informations sur les activités de la Task Force.

1. W.M.P. van der Aalst. Process Mining : Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin, 2011. <http://www.processmining.org/book/>

2 Fouille de Procédé : État de l'Art

Les capacités de croissance des systèmes d'information et d'autres systèmes qui dépendent de l'informatique, sont bien caractérisées par la loi de Moore. Gordon Moore, co-fondateur d'Intel, a prédit en 1965 que le nombre de composants dans les circuits intégrés doublerait chaque année. Durant la dernière cinquante d'années, la croissance a en effet été exponentielle, quoique à un rythme légèrement plus lent. Ces avancées ont entraîné une croissance spectaculaire de "l'univers numérique" (c'est à dire, toutes les données stockées et/ou échangées par voie électronique). Par ailleurs, les univers numérique et réel deviennent de plus en plus alignés.

La croissance d'un univers numérique qui est bien alignée avec les procédés dans les organisations, rend possible l'enregistrement et l'analyse des événements. Les événements peuvent varier du retrait d'argent à un guichet automatique, au médecin ajustant une machine à rayons X, au citoyen passant un permis de conduire, au dépôt d'une déclaration fiscale, et à la réception d'un numéro de billet électronique par un voyageur. Le défi consiste à exploiter les événements de trace d'une manière significative, par exemple, de donner un aperçu, d'identifier les goulots d'étranglement, d'anticiper les problèmes, les violations de politique d'archivage, de recommander des contre-mesures, et de rationaliser les procédés. La fouille de procédé vise à faire exactement cela.

Le point de départ pour la fouille de procédé est les *événements de trace d'exécution*. Toutes les techniques de fouille de procédé supposent qu'il est possible d'enregistrer séquentiellement les événements tel que chaque événement se réfère à une activité (par exemple, une étape bien définie dans un procédé) et est lié à un cas particulier (par exemple, une instance de procédés). Les traces d'exécution peuvent stocker des informations supplémentaires sur les événements. En fait, dans la mesure du possible, les techniques de fouille de procédé utilisent des informations supplémentaires telles que la *ressource* (par exemple, une personne ou un dispositif) exécutant ou instanciant l'activité, *l'heure d'exécution* de l'événement, ou des *éléments de données* enregistrés avec l'événement (par exemple, la taille d'une commande).

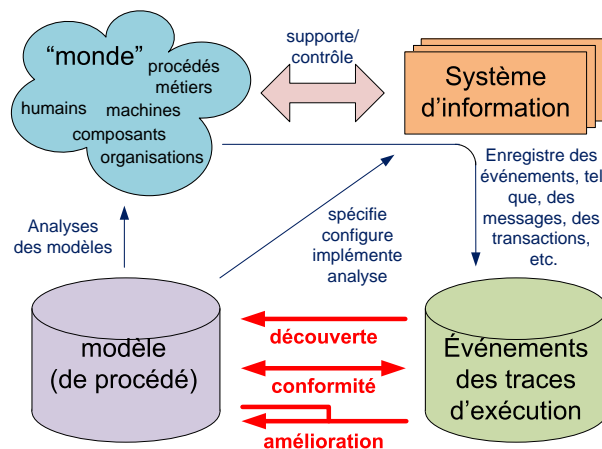


Figure 2. Positionnement des trois principaux types de la fouille de procédé : (a) *découverte*, (b) vérification de la *conformité*, et (c) *amélioration*.

Comme il est illustré dans Figure 2, les événements de trace d'exécution peuvent être utilisés pour effectuer trois types de fouille de procédé. Le premier type de fouille de procédé est la *découverte*. Une technique de découverte prend les événements de traces d'exécution et produit un modèle sans utiliser aucune information a priori. La découverte de procédé est la technique la plus importante des techniques de fouille de procédé. Pour de nombreuses organisations, il est surprenant de voir que les techniques existantes sont en effet capables de découvrir des procédés

réels simplement en se basant sur les événements de trace d'exécutions. Le deuxième type de fouille de procédé est la *conformité*. Ici, un modèle de procédé existant est comparé aux événements des traces d'exécution du même procédé. La vérification de conformité peut être utilisée pour vérifier si la réalité, comme elle est enregistrée dans les traces d'exécution, est conforme au modèle et vice versa. Notez que différents types de modèles peuvent être envisagés : la vérification de conformité peut être appliquée aux modèles procéduraux, aux modèles organisationnels, aux modèles de procédé déclaratifs, aux règles/politiques métiers, aux lois, etc. Le troisième type de fouille de procédé est l'*amélioration*. Ici, l'idée est d'étendre ou d'améliorer un modèle de procédé existant en utilisant des informations sur le procédé réel enregistré dans certains événements de trace d'exécution. Alors que la vérification de conformité mesure l'alignement entre le modèle et la réalité, ce troisième type de fouille de procédé vise à changer ou étendre le modèle a priori. Par exemple, en utilisant le temps dans les événements de trace d'exécution, on peut étendre le modèle pour montrer les goulots d'étranglement, les niveaux de service, les temps de passage, et les fréquences.

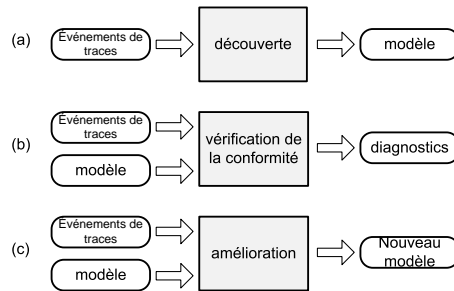


Figure 3. Les trois types de base de la fouille de procédé expliqués en terme d'entrée et de sortie : (a) découverte, (b) vérification de la conformité, et (c) amélioration.

Figure 3 décrit les trois types de fouille de procédé en termes d'entrée et de sortie. Les techniques de découverte prennent les événements des traces d'exécution et produisent un modèle. Le modèle découvert est typiquement un modèle de procédé (par exemple, un réseau de Petri, BPMN, EPC, ou un diagramme d'activité UML), toutefois, le modèle peut également décrire d'autres perspectives (par exemple, un réseau social). Les techniques de vérification de conformité ont besoin des événements de traces d'exécution et du modèle en entrée. La sortie se compose d'informations de diagnostic montrant les différences et les similitudes entre le modèle et les traces d'exécution. Les techniques d'amélioration du modèle (réparation ou extension) ont aussi besoin des événements des traces d'exécution et du modèle en entrée. La sortie est un modèle amélioré ou étendu.

La fouille de procédés peut couvrir différentes perspectives. La *perspective de flux de contrôle* se concentre sur le flot de contrôle, c'est à dire, l'ordre des activités. L'objectif de la fouille de procédé dans cette perspective est de trouver une bonne caractérisation de tous les chemins possibles. Le résultat est généralement exprimé en termes de réseau de Petri ou une autre méthode de modélisation de procédé (par exemple, EPC, BPMN, ou des diagrammes d'activité UML). La *perspective organisationnelle* axée sur les renseignements concernant les ressources cachées dans les traces d'exécution, c'est à dire qui sont les acteurs (par exemple, les personnes, les systèmes, les rôles, ou les départements) impliqués et comment sont-ils liés. L'objectif est de structurer l'organisation soit en classant les personnes en termes de rôles et d'unités organisationnelles ou pour illustrer le réseau social. La *perspective d'instance* se concentre sur les propriétés des instances. Évidemment, une instance peut être caractérisée par son chemin dans le procédé ou par les acteurs qui l'ont exécutée. Toutefois, les instance peuvent aussi être caractérisées par les valeurs des éléments de données correspondantes. Par exemple, si une instance représente une commande de réapprovisionnement, il peut être intéressant de connaître le fournisseur ou le nombre

de produits commandés. La *perspective temporelle* est concernée par le timing et la fréquence des événements. Lorsque les événements ont un temps d'exécution, il est possible de découvrir les goulots d'étranglement, mesurer les niveaux de service, surveiller l'utilisation des ressources, et prédire le temps de traitement restant des instances en cours.

Il y a quelques idées courantes fausses liées à la fouille de procédé. Certains vendeurs, analystes et chercheurs limitent la portée de la fouille de procédé à une technique spéciale de fouille de données pour la découverte de procédé qui ne peut être utilisée que pour une analyse hors ligne. Ce *n'est pas* le cas, par conséquent, nous insistons sur les trois caractéristiques suivantes :

- *La fouille de procédé n'est pas limitée à la découverte de flot de contrôle.* La découverte des modèles de procédé à partir des événements des traces d'exécution alimente l'imagination des praticiens et des universitaires. Par conséquent, la découverte de flot de contrôle est souvent considérée comme la partie la plus excitante de la fouille de procédé. Cependant, la fouille de procédé n'est pas limitée à la découverte de flot de contrôle. D'une part, la découverte est simplement l'une des trois formes de base de la fouille de procédé (la découverte, la conformité et l'amélioration). D'autre part, la portée n'est pas limitée au flot de contrôle; les perspectives d'organisation, d'instance et de temps jouent également un rôle important.
- *La fouille de procédé n'est pas seulement un type spécifique de fouille de données.* La fouille de procédé peut être vue comme le "lien invisible" entre la fouille de données et le modèle-dirigé BPM traditionnel. La plupart des techniques de fouille de données ne sont pas du tout centrées sur les procédés. Les modèles de procédé incluant potentiellement la concurrence sont incomparables à des simples structures de fouille de données tels que les arbres de décision et les règles d'association. Par conséquent, des types complètement nouveaux de représentations et d'algorithmes sont nécessaires.
- *La fouille de procédé n'est pas limitée à une analyse hors ligne.* Les techniques de fouille de procédé extraient des connaissances à partir des événements de traces historiques. Bien que des données "post-mortem" sont utilisées, les résultats peuvent être appliqués à des cas en cours. Par exemple, le délai d'achèvement d'une commande client partiellement traitée peut être prédit en utilisant un modèle de procédé découvert.

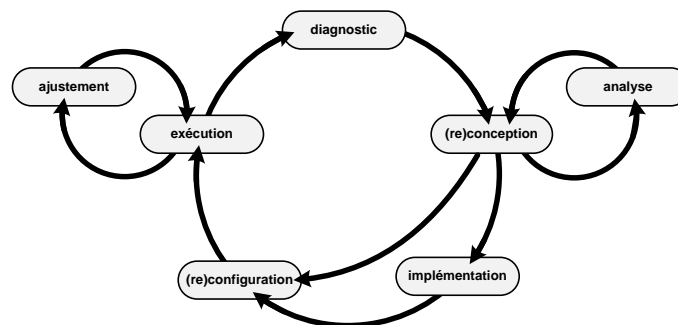


Figure 4. Le cycle de vie BPM identifiant les différentes phases d'un procédé métier et de son(s) système(s) d'information correspondant(s); la fouille de procédé joue un rôle potentiel dans toutes les phases (sauf pour la phase d'implémentation).

Pour positionner la fouille de procédé, nous utilisons le cycle de vie du Business Process Management (BPM) illustré dans Figure 4. Le cycle de vie BPM montre sept phases d'un procédé métier et de ses système(s) d'information correspondant(s). Dans la *phase de (re)conception*, un nouveau modèle de procédé est créé ou un modèle de procédé existant est adapté. Dans la *phase d'analyse*, un modèle candidat et ses alternatives sont analysés. Après la phase de (re)conception, le modèle est implémenté (*phase d'implémentation*), ou un système existant est (re)configuré (*phase de (re)configuration*). Dans la phase d'implémentation, le modèle conçu est exécuté. Pendant la phase d'implémentation le procédés est surveillé. Par ailleurs, des petits ajustements peuvent être

effectués sans la restructuration du procédé (*phase d'ajustement*). Dans la *phase de diagnostic*, le procédé adopté est analysé et la sortie de cette phase peut déclencher une nouvelle phase de (re)conception. La fouille de procédé est un outil précieux pour la plupart des phases montrées dans Figure 4. De toute évidence, la phase de diagnostic peut bénéficier de la fouille de procédé. Cependant, la fouille de procédé n'est pas limitée à la phase de diagnostic. Par exemple, dans la phase d'exécution, les techniques de fouille de procédé peuvent être utilisées pour le *support opérationnel*. Des prédictions et des recommandations fondées sur des modèles appris en utilisant les traces d'exécution peuvent être utilisées pour influencer les cas en cours d'exécution. Des formes similaires d'aide à la décision peuvent être utilisées pour ajuster les procédés et pour guider le processus de (re)configuration.

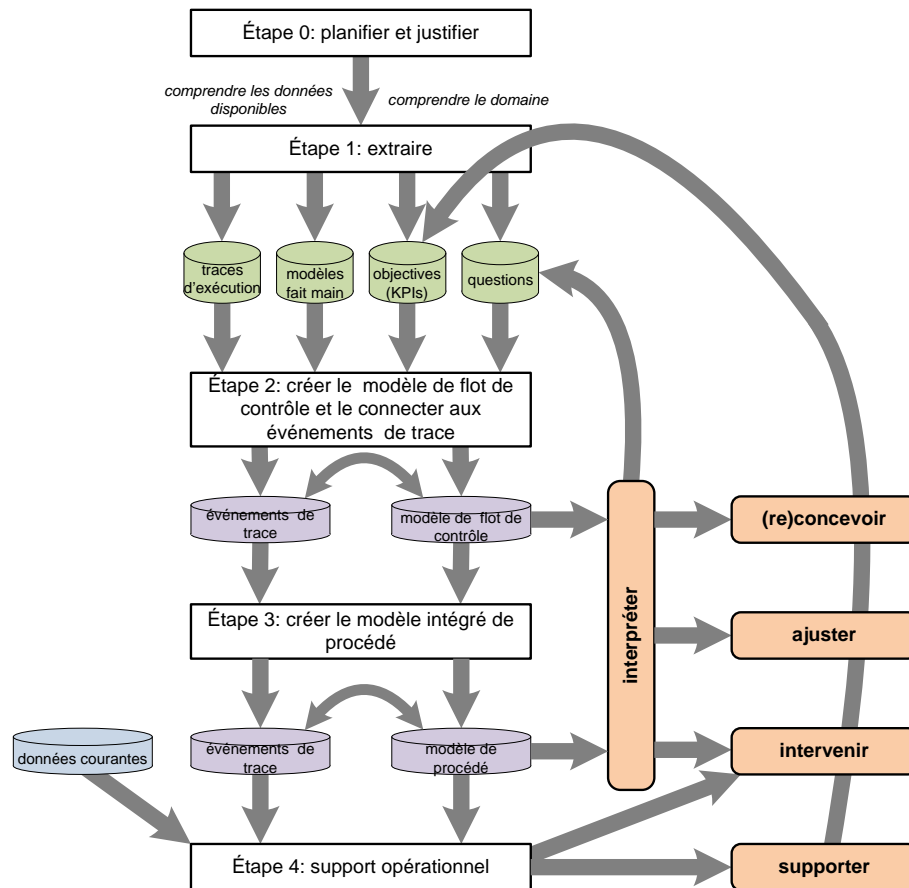


Figure 5. Le modèle du cycle de vie L^* décrivant un projet de fouille de procédé se composant de cinq étapes : planifier et justifier (phase 0), extraire (phase 1), créer un modèle de flot de contrôle et le connecter aux événements des traces d'exécution (étape 2), créer un modèle de procédé intégré (étape 3), et fournir un support opérationnel (phase 4).

Tandis que Figure 4 montre l'ensemble du cycle de vie de BPM, Figure 5 se concentre sur les activités concrètes et les artefacts de fouille de procédé. Figure 5 décrit les phases possibles dans un projet de fouille de procédé. Tout projet de fouille de procédé commence par une planification et une justification de cette planification (étape 0). Après le lancement du projet, des événements des traces d'exécution, des modèles, des objectifs et des questions doivent être extraits des systèmes, des experts du domaine et de gestion (étape 1). Cela nécessite une bonne compréhension des données disponibles ("Qu'est ce qui peut être utilisé pour l'analyse?") et une

bonne compréhension du domaine (“Quelles sont les questions importantes?”) et des résultats dans les artefacts montrés dans Figure 5 (c’est à dire, les traces d’exécution, les modèles faits à la main, les objectifs et les questions). À l’étape 2 le modèle de flot de contrôle est construit et relié aux traces d’exécution. Ici, des techniques automatisées de découverte de procédé peuvent être utilisées. Le modèle de procédé découvert peut déjà fournir des réponses à certaines questions et déclencher des actions de reconception et d’adaptation. Par ailleurs, les événements des traces d’exécution peuvent être filtrés ou adaptés en utilisant le modèle (par exemple, la suppression des activités rares ou des instances aberrantes, et l’ajout d’événements manquants). Parfois, des efforts importants sont nécessaires pour corréliser les événements appartenant à la même instance de procédé. Les autres événements sont liés à des entités du modèle de procédé. Lorsque le procédé est relativement structuré, le modèle de flot de contrôle peut être étendu avec d’autres perspectives (par exemple, les données, le temps et les ressources) pendant l’étape 3. La relation entre les événements des traces d’exécution et le modèle créé à l’étape 2 est utilisée pour étendre le modèle (par exemple, l’horodatage des événements associés est utilisé pour estimer les temps d’attente pour les activités). Ceci peut être utilisé pour répondre aux questions supplémentaires et peut déclencher des actions supplémentaires. Finalement, les modèles construits à l’étape 3 peuvent être utilisés pour le support opérationnel (étape 4). Les connaissances extraites des données des traces d’exécution sont combinées avec des informations sur les instances en cours d’exécution. Ceci peut être utilisé pour intervenir, prédire, et recommander. Les étapes 3 et 4 ne peuvent être atteintes que si le procédé est suffisamment stable et structuré.

Actuellement, il existe des techniques et des outils qui peuvent supporter toutes les étapes montrés dans Figure 5. Cependant, la fouille de procédé est un paradigme relativement nouveau et la plupart des outils actuellement disponibles sont encore assez immatures. Par ailleurs, les utilisateurs potentiels ne sont pas souvent conscients du potentiel et des limites de la fouille de procédé. Par conséquent, ce manifeste catalogue certains principes directeurs (cf. Section 4) et défis (cf. Section 4) pour les utilisateurs de techniques de fouille de procédé, ainsi que les chercheurs et les développeurs qui sont intéressés à faire progresser l’état de l’art.

3 Principes Directeurs

Comme avec toute nouvelle technologie, il y a des erreurs évidentes qui peuvent être faites lors de l’application de la fouille de procédé dans les conditions de la vie réelle. Par conséquent, nous énumérons six *principes directeurs* pour empêcher les utilisateurs/analystes de faire de telles erreurs.

3.1 PD1 : Les Traces d’Exécution Doivent Être Traitées comme des Citoyens de Première Classe

Le point de départ pour toute activité de fouille de procédé est les événements enregistrés des traces d’exécution. Nous nous référons aux collections d’événements en tant que *traces d’exécution*, cependant, cela ne signifie pas que les événements doivent être stockés dans des fichiers de traces d’exécution dédiés. Les événements peuvent être stockés dans des tables de bases de données, des messages, des archives de courrier, les traces d’exécution de transactions, et d’autres sources de données. Plus important que le format de stockage, c’est la *qualité* des traces d’exécution. La qualité d’un résultat d’une fouille de procédé dépend fortement de l’entrée. Par conséquent, les événements des traces d’exécution doivent être traités comme des citoyens de première classe dans les systèmes d’information supportant les procédés à analyser. Malheureusement, les événements des traces d’exécution sont souvent simplement un “sous-produit” utilisé pour le débogage ou le profilage. Par exemple, les appareils médicaux de Philips Healthcare enregistrent tout simplement des événements parce que les développeurs de logiciels ont inséré des “print statements” dans le code. Bien qu’il existe quelques directives informelles pour l’ajout de ces déclarations dans le code, une approche plus systématique est nécessaire pour améliorer la qualité des événements des traces d’exécution. Les événements de traces doivent être considérés comme des citoyens de première classe (plutôt que des citoyens de seconde classe).

Il existe plusieurs critères pour juger de la qualité des événements de trace. Les événements doivent être *fiables*, c'est à dire, on doit être sûr de supposer que les événements enregistrés se sont réellement passés et que les attributs d'événements sont corrects. Les événements de traces d'exécution doivent être *complets*, c'est à dire, étant donné une portée particulière, aucun événement ne peut être manquant. Tout événement enregistré doit avoir une *sémantique* bien définie. Par ailleurs, les événements de traces doivent être *sûrs*, dans le sens que les préoccupations de confidentialité et de sécurité sont abordées lors de l'enregistrement des événements. Par exemple, les acteurs doivent être conscients de la nature des événements qui sont enregistrés et la façon avec laquelle ils sont utilisés.

Table 1 définit cinq niveaux de maturité des événements des traces d'exécution allant de l'excellente qualité (★★★★★) à la plus mauvaise qualité (★). Par exemple, les événements des traces d'exécution de Philips Healthcare appartiennent au niveau ★★★, c'est à dire, les événements sont enregistrés automatiquement et le comportement enregistré correspond à la réalité, mais aucune approche systématique est utilisée pour assigner une sémantique aux événements et assurer une couverture à un niveau particulier. Les techniques de fouille de procédé peuvent être appliquées à des traces d'exécution appartenant aux niveaux ★★★★★, ★★★★ et ★★★. En principe, il est également possible d'appliquer la fouille de procédé en utilisant des événements des traces d'exécution appartenant aux niveaux ** ou *. Cependant, l'analyse de ces traces est généralement problématique et les résultats ne sont pas fiables. En fait, il n'est pas judicieux ou fructueux d'appliquer la fouille de procédé à des traces d'exécution appartenant au niveau *.

Afin de bénéficier de la fouille de procédé, les organisations devraient viser des événements de traces d'exécution au niveau de qualité le plus élevé possible.

3.2 PD2 : L'Extraction des Traces d'Exécution Devrait Être Guidée par des Questions

Figure 5 montre que les activités de fouille de procédé doivent être conduites par des questions. Sans questions concrètes, il est très difficile d'extraire des traces d'exécution significatives. Considérons, par exemple, des milliers de tables dans la base d'un système ERP comme SAP. Sans questions concrètes, il est impossible de sélectionner les tables pertinentes pour l'extraction de données des traces d'exécution.

Un modèle de procédé tels que celui montré dans Figure 1 décrit le cycle de vie des instances (c'est à dire, les instances de procédé) d'un type particulier. Ainsi, avant d'appliquer une technique de fouille de procédé on doit choisir le type d'instance à analyser. Ce choix devrait être guidé par les questions auxquelles on doit avoir une réponse et cela peut être non négligeable. Considérons, par exemple, la manipulation des commandes de clients. Chaque commande client peut être constituée de lignes de commandes multiples comme le client peut commander des produits multiples en une seule commande. Une commande peut entraîner des livraisons multiples. Une livraison peut se référer à l'ordre de lignes des commandes multiples. Par conséquent, il existe une relation plusieurs-à-plusieurs entre les commandes et les livraisons et une relation un-à-plusieurs entre les commandes et les lignes de commande. Étant donné une base de données avec des événements de traces liés à des commandes, lignes de commandes, et livraisons, il y a des modèles de procédé différents qui peuvent être découverts. On peut extraire des données dans le but de décrire le cycle de vie des commandes individuelles. Cependant, il est également possible d'extraire des données dans le but de découvrir le cycle de vie des lignes de commandes individuelles ou le cycle de vie des livraisons individuelles.

3.3 PD3 : La Concurrence, le Choix et d'Autres Structures de base du Flot de contrôle Devraient Être Supportés

Une pléthore de langages de modélisation de procédé existe (par exemple, BPMN, EPC, les réseaux de Petri, BPEL, et des diagrammes d'activité UML). Certains de ces langages fournissent de nombreux éléments de modélisation (par exemple, BPMN offre plus de 50 différents éléments graphiques) alors que d'autres sont très basiques (par exemple, les réseaux de Petri sont composés

Table 1. Niveaux de maturité des événements des traces d'exécution.

Niveau	Caractérisation
★★★★★	<p>Plus haut niveau : les événements des traces d'exécution sont d'excellente qualité (à savoir, fiables et complets) et les événements sont bien définis. Les événements sont enregistrés dans un système automatique, systématique, de manière fiable et sûre. Les considérations de confidentialité et de sécurité sont abordées d'une manière adéquate. Par ailleurs, les événements enregistrés (et tous leurs attributs) ont une sémantique claire. Cela implique l'existence d'une ou plusieurs ontologies. Les événements et leurs attributs pointent vers cette ontologie.</p> <p><i>Exemple</i> : les traces d'exécution sémantiquement annotées des systèmes BPM.</p>
★★★★	<p>Les événements sont enregistrés automatiquement et de manière systématique et fiable, c'est à dire, les traces d'exécution sont fiables et complets. Contrairement aux systèmes appartenant au niveau de ★★★, des notions telles que par exemple les instances (cas) et les activités de procédé sont supportées de manière explicite.</p> <p><i>Exemple</i> : les événements des traces d'exécution des systèmes traditionnels BPM/Workflow.</p>
★★★	<p>Les événements sont enregistrés automatiquement, mais aucune approche systématique est suivie pour enregistrer les événements. Cependant, contrairement aux traces d'exécution au niveau ★★, il y a un certain niveau de garantie que les événements enregistrés correspondent à la réalité (par exemple, les événements de traces d'exécution sont fiables, mais pas nécessairement complets). Considérons, par exemple, les événements enregistrés par un système ERP. Bien que les événements doivent être extraits d'une variété de tableaux, l'information peut être supposée correcte (par exemple, il est évident de supposer qu'un paiement enregistré par l'ERP existe réellement et vice versa).</p> <p><i>Exemples</i> : les tableaux dans les systèmes ERP, les événements des traces d'exécution des systèmes CRM, les traces d'exécution de transactions des systèmes de messagerie, les événements des traces d'exécution des systèmes high-tech, etc.</p>
★★	<p>Les événements sont enregistrés automatiquement, c'est à dire, en tant que sous-produit de certains systèmes d'information. La couverture varie, c'est à dire, aucune approche systématique est suivie pour décider quels événements sont enregistrés. Par ailleurs, il est possible de contourner le système d'information. Par conséquent, des événements peuvent être manquants ou non enregistrés correctement.</p> <p><i>Exemples</i> : les événements des traces d'exécution des systèmes de gestion de produits et de documents, des traces d'exécution d'erreurs des systèmes embarqués, des feuilles de techniciens de maintenance, etc.</p>
★	<p>Niveau le plus bas : les événements des traces d'exécution sont de mauvaise qualité. Les événements enregistrés peuvent ne pas correspondre à la réalité et les événements peuvent être manquants. Les événements des traces d'exécution pour lesquels les événements sont enregistrés à la main ont généralement de telles caractéristiques.</p> <p><i>Exemples</i> : les traces laissées dans les documents papier acheminés à travers l'organisation ("note jaune"), des dossiers médicaux sur papiers, etc.</p>

seulement de trois éléments différents : des places, des transitions et des arcs). La description de flot de contrôle est l'épine dorsale de tout modèle de procédé. Les structures de base de workflow (aussi connues comme *patterns*) supportées par tous les langages classiques sont séquence, routage parallèle (AND-splits/joins), choix (XOR-splits/joins), et boucles. Évidemment, ces patrons devraient être supportés par les techniques de fouille de procédés. Cependant, certaines techniques

ne sont pas en mesure de découvrir la concurrence et supportent seulement les chaînes de Markov/systèmes de transition.

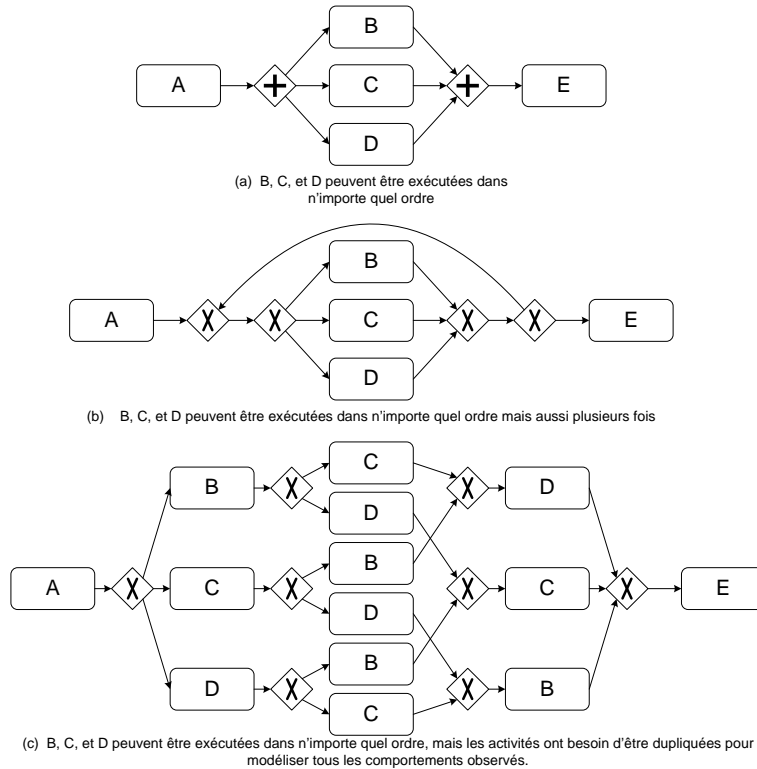


Figure 6. Un exemple illustrant des problèmes quand la concurrence (c'est-à-dire AND-splits/joins) ne peut pas être exprimée directement. Dans l'exemple seulement trois activités (B , C , et D) sont en concurrence. Imaginez les modèles de procédé résultant quand il y a 10 activités en concurrence ($2^{10} = 1024$ états et $10! = 3,628,800$ séquences d'exécution possibles).

Figure 6 montre l'effet de l'utilisation de techniques de fouille de procédé incapables de découvrir la concurrence (pas de AND-split/joins).

Considérons la trace d'exécution suivante $L = \{\langle A, B, C, D, E \rangle, \langle A, B, D, C, E \rangle, \langle A, C, B, D, E \rangle, \langle A, C, D, B, E \rangle, \langle A, D, B, C, E \rangle, \langle A, D, C, B, E \rangle\}$. L contient des cas qui commencent par A et se terminent par E . Les activités B , C et D apparaissent dans n'importe quel ordre entre A et E . Le modèle BPMN dans Figure 6(a) montre une représentation compacte du procédé sous-jacent en utilisant les opérateurs AND. Supposons que la technique de fouille de procédé ne prend pas en charge les opérateurs AND. Dans ce cas, les deux autres modèles BPMN dans Figure 6 sont des candidats évidents. Le modèle BPMN dans Figure 6(b) est compact, mais permet beaucoup trop de comportements (par exemple, des cas tels que $\langle A, B, B, B, E \rangle$ sont possibles selon le modèle, mais ne sont pas probables selon les événements des traces d'exécution). Le modèle BPMN dans Figure 6(c) tient en compte toutes les instances de L , mais encode toutes les séquences explicitement, il n'est pas donc une représentation compacte des traces d'exécution. L'exemple montre que pour des modèles réels ayant des dizaines d'activités potentiellement concurrentes les modèles résultants sont sévèrement sous ajustés (c'est à dire, permettant beaucoup trop de comportements) et/ou extrêmement complexes si la concurrence n'est pas supportée.

Comme il est illustré par Figure 6, il est important de supporter au moins les patrons de base de workflow. Outre les patrons de base mentionnés, il est également souhaitable de supporter OR-

splits/joins, parce que ceux-ci fournissent une représentation compacte de décisions inclusives et de synchronisations partielles.

3.4 PD4 : Les Événements Doivent Être Liés aux éléments du modèle

Comme il est indiqué dans Section 2, c'est une erreur de considérer que la fouille de procédé est limitée à la découverte de flot de contrôle. Figure 1 montre que le modèle de procédé découvert peut couvrir différents points de vue (perspective organisationnel, perspective temporelle, perspective des données, etc.).

Par ailleurs, la découverte est juste l'un des trois types de fouille de procédé représentés dans Figure 3. Les deux autres types de fouille de procédé (vérification de la conformité et l'amélioration) dépendent fortement de la relation entre les éléments du modèle et les événements dans les traces d'exécution. Cette relation peut être utilisée pour "rejouer" les événements des traces d'exécution sur le modèle. "Rejouer" peut être utilisé pour révéler les écarts entre les événements des traces d'exécution et un modèle, par exemple, certains événements dans les traces d'exécution ne sont pas possibles selon le modèle. Les techniques de vérification de la conformité quantifient et diagnostiquent de tels écarts. Le temps dans les événements des traces d'exécution peut être utilisé pour analyser le comportement temporel pendant la relecture. Les différences de temps entre les activités liées causalement peut être utilisées pour ajouter les temps d'attente prévus dans le modèle. Ces exemples montrent que la relation entre les événements dans les traces d'exécution et les éléments dans le modèle servent de point de départ pour les différents types d'analyse.

Dans certains cas il peut être non négligeable d'établir une telle relation. Par exemple, un événement peut se référer à deux activités différentes, ou il n'est pas clair à quelle activité il se réfère. De telles ambiguïtés doivent être enlevées afin d'interpréter les résultats de la fouille de procédé traiter correctement. Outre le problème de mise en relation des événements aux activités, il y a le problème de mise en relation des événements aux instances de procédé. Ceci est communément appelé la *corrélation d'événements*.

3.5 PD5 : Les Modèles Devraient Être Traités comme des Abstractions Intentionnelle de la Réalité.

Les modèles dérivés des traces d'exécution offrent *des vues sur la réalité*. Une telle vision devrait fournir une abstraction résolue du comportement capturé dans les traces d'exécution. Étant donné des traces d'exécution, il peut y avoir plusieurs vues qui sont utiles. Par ailleurs, les différents intervenants peuvent exiger des vues différentes. En fait, les modèles découverts à partir des traces d'exécution doivent être considérés comme des "cartes" (comme des cartes géographiques). Ce principe directeur fournit des renseignements importants, dont deux sont décrites dans la suite.

Tout d'abord, il est important de noter qu'il n'y a rien de tel qu'une "carte" pour une zone géographique particulière. Selon l'usage prévu il y a différentes cartes : cartes routières, cartes de randonnée, cartes cyclistes, etc. Toutes ces cartes montrent une vue sur la même réalité et il serait absurde de supposer qu'il y aurait une "carte parfaite". La même chose vaut pour les modèles de procédé : le modèle devrait insister sur les choses pertinentes pour un type particulier d'utilisateur. Les modèles découverts peuvent se concentrer sur des perspectives différentes (flot de contrôle, flot de données, temps, ressources, coûts, etc.) et les montrer à différents niveaux de granularité et de précision, par exemple, un gestionnaire peut vouloir voir un modèle de procédé informel grossier en se concentrant sur les coûts tandis qu'un analyste de procédé peut vouloir voir un modèle de procédé détaillé portant sur les écarts du flot normal. À noter également que les différentes parties prenantes peuvent vouloir voir un procédé à différents niveaux : au niveau stratégique (les décisions à ce niveau ont des effets à long terme et sont fondées sur des traces d'exécution agrégées sur une plus longue période), au niveau tactique (les décisions à ce niveau ont des effets à moyen terme et sont principalement basées sur des données récentes), et niveau opérationnel (les décisions à ce niveau ont des effets immédiats et sont fondées sur les traces d'exécution liées à des instances en cours d'exécution).

Deuxièmement, il est utile d'adopter les idées de la cartographie quand il s'agit de produire des cartes compréhensibles. Par exemple, les cartes routières abstraient les routes et les villes moins importantes. Des choses moins importantes sont soit laissées de côté soit dynamiquement regroupées dans des formes agrégées (par exemple, les rues et les faubourgs sont fusionnés dans les villes). Les cartographes n'éliminent pas seulement les détails inutiles, mais aussi utilisent des couleurs pour mettre en évidence des caractéristiques importantes. Par ailleurs, les éléments graphiques ont une taille particulière pour indiquer leur signification (par exemple, la taille des lignes et des points peuvent varier). Les cartes géographiques ont également une interprétation claire de l'axe des x et l'axe des y , à savoir, la présentation d'une carte n'est pas arbitraire, autant que les coordonnées des éléments qui ont une signification. Tout cela est en fort contraste avec les modèles de procédé courants, qui n'utilisent pas typiquement la couleur, la taille et les fonctions de localisation pour rendre les modèles plus compréhensibles. Cependant, les idées de la cartographie peut facilement être intégrés dans la construction des cartes des procédés découvertes. Par exemple, la taille d'une activité peut être utilisée afin de refléter sa fréquence ou de toutes autre propriété indiquant sa signification (par exemple, les coûts ou de l'utilisation des ressources). La largeur d'un arc peut refléter l'importance de la dépendance causale correspondante, et la coloration des arcs peuvent être utilisées pour mettre en évidence les goulots d'étranglement.

Les observations ci-dessus montrent qu'il est important de sélectionner une bonne représentation et de l'ajuster pour le public visé. Ceci est important pour la visualisation des résultats aux utilisateurs finaux et pour guider les algorithmes de découverte vers les modèles appropriés (voir aussi Défi D5).

3.6 PD6 : La Fouille de Procédé Devrait Être un Processus Continu

La fouille de procédé peut aider à fournir des "cartes" significatives qui sont directement reliées à des traces d'exécution. Les données des traces d'exécution et les données actuelles peuvent être projetées sur de tels modèles. Par ailleurs, les procédés peuvent changer alors qu'ils sont en cours d'analyse. Étant donné la nature dynamique des procédés, il n'est pas conseillé de voir la fouille de procédé comme une activité ponctuelle. L'objectif ne doit pas être de créer un modèle fixe, mais de redonner vie à des modèles de procédé afin que les utilisateurs et les analystes sont encouragés à les regarder sur une base quotidienne.

On peut Comparer cela à l'utilisation de mashups utilisant le géo-marquage. Il y a des milliers de mashups utilisant Google Maps (par exemple, les applications projetant des informations sur les conditions de circulation, de l'immobilier, des restaurants fast-food, ou les horaires de films sur une carte choisie). Les gens peuvent parfaitement zoomer et dézoomer en utilisant ces cartes et interagir avec eux (par exemple, les embouteillages sont projetées sur la carte et l'utilisateur peut sélectionner un problème particulier pour voir les détails).

Il devrait également être possible d'effectuer une fouille de données basée sur des événements temps réel. En utilisant la "métaphore carte", nous pouvons penser à des événements ayant des coordonnées GPS qui peuvent être projetées sur des cartes en temps réel. Analogie aux systèmes de navigation automobile, les outils de fouille de procédé peuvent aider les utilisateurs finaux (a) en naviguant à travers les procédés, (b) en projetant des informations dynamiques sur des cartes de procédé (par exemple, en montrant les "embouteillages" dans les procédés métiers), et (c) en fournissant des prédictions concernant les instances en exécution (par exemple, l'estimation de la "heure d'arrivée" d'une instance qui est retardée). Ces exemples démontrent qu'il est dommage de ne pas utiliser les modèles de procédé plus activement. Par conséquent, la fouille de procédé devrait être considérée comme un processus continu fournissant des informations à une action conformément aux différents échelles de temps (minutes, heures, jours, semaines et mois).

4 Défis

La fouille de procédé est un outil important pour les organisations modernes qui ont besoin de gérer des procédés opérationnels non triviaux. D'un côté, il y a une croissance incroyable de traces

d'exécution. D'autre part, les procédés et les informations doivent être parfaitement alignés afin de répondre aux exigences liées à la conformité, l'efficacité et au service clientèle. Malgré l'applicabilité de la fouille de procédé il y a encore des défis importants qui doivent être abordés ; ceci illustre le fait que la fouille de procédé est une discipline émergente. Dans la suite, nous énumérons certains de ces défis. Cette liste n'est pas destinée à être complète et, au fil du temps, de nouveaux défis peuvent émerger ou des défis existants pourraient disparaître en raison des progrès dans la fouille de procédé.

4.1 D1 : Trouver, Fusionner, et Nettoyer les traces d'exécution

Il faut encore des efforts considérables pour extraire des traces d'exécution adaptées à la fouille de données. Typiquement, plusieurs obstacles doivent être surmontés :

- Les données des traces d'exécution peuvent être *distribuées* sur une variété de sources. Ces données doivent être fusionnées. Ceci tend à être problématique quand des identifiants différents sont utilisés dans les différentes sources de données. Par exemple, un système utilise le nom et la date de naissance pour identifier une personne, tandis qu'un autre système utilise le numéro de sécurité sociale de la personne.
- Les traces d'exécutions sont souvent "centrée sur l'objet" plutôt que "centrée sur le procédés". Par exemple, des produits individuels, des palettes et des conteneurs peuvent avoir des étiquettes RFID et les événements enregistrés se réfèrent à ces balises. Toutefois, pour surveiller une commande client particulière, de tels événements centrés sur l'objet doivent être fusionnés et prétraités.
- Les traces d'exécution peuvent être *incomplètes*. Un problème commun est que les événements ne pointent pas explicitement sur des instances de procédé. Souvent, il est possible d'inférer cette information, mais cela peut nécessiter des efforts considérables. Par ailleurs, l'information sur le temps peut être manquante pour certains événements. On aurait besoin d'induire cette information sur le temps afin de continuer à l'utiliser pour des informations de synchronisation disponibles.
- Les traces d'exécution peuvent contenir *des données aberrantes*, c'est à dire, un comportement exceptionnel appelé aussi *bruit*. Comment définir des valeurs aberrantes? Comment détecter de telles valeurs aberrantes? On doit répondre à ces questions pour nettoyer les traces d'exécution.
- Des traces d'exécution peuvent contenir des événements à *différents niveaux de granularité*. Dans les traces d'exécution du système d'information d'un hôpital les événements peuvent se référer à des tests sanguins simples ou des procédures chirurgicales complexes. Aussi les temps d'exécution peuvent avoir différents niveaux de granularité allant de la précision de millisecondes (28-9-2011 : h11m28s32ms342) au jour (28-9-2011).
- Des événements se produisent dans un *contexte* particulier (météo, charge intensif de travail, jours de la semaine, etc.). Ce contexte peut expliquer certains phénomènes, par exemple, le temps de réponse est plus long que d'habitude en raison de travaux en cours ou les jours fériés. Pour l'analyse, il est souhaitable d'intégrer ce contexte. Cela implique la fusion des traces d'exécution avec des données contextuelles. Ici, la "malédiction de la dimensionnalité" frappe l'analyse qui devient insoluble par l'ajout de trop nombreuses variables.

De meilleurs outils et les méthodologies sont nécessaires pour résoudre les problèmes ci-dessus. Par ailleurs, comme indiqué précédemment, les organisations ont besoin de traiter les traces d'exécution comme des citoyens de première classe, plutôt que des sous-produits. L'objectif est d'obtenir des traces d'exécution ***** (voir Table 1). Ici, les leçons apprises dans le contexte de datawarehousing sont utiles pour assurer des traces d'exécution de haute qualité. Par exemple, de simples vérifications lors de la saisie des données peuvent aider à réduire la proportion de données incorrectes de manière significative.

4.2 D2 : Traiter des Traces d'Exécution Ayant Diverses Caractéristiques

Les traces d'exécution peuvent avoir des caractéristiques très différentes. Certaines traces d'exécution peuvent être extrêmement larges rendant difficile leur traitement alors que d'autres

traces d'exécution sont si petites que pas assez de données est disponible pour faire des conclusions fiables.

Dans certains domaines, des quantités ahurissantes d'événements sont enregistrées. Par conséquent, des efforts supplémentaires sont nécessaires pour améliorer les performances et l'évolutivité. Par exemple, ASML suit en permanence tous ses scanners à plaques. Ces scanners à plaques sont utilisés par diverses organisations (par exemple, Samsung et Texas Instruments) pour produire des puces (env. 70% des puces sont produites à l'aide des scanners à plaques d'ASML). Les outils existants ont des difficultés à traiter les pétaoctets de données recueillies dans ces domaines. Outre le nombre d'événements enregistrés, il y a d'autres caractéristiques telles que le nombre moyen d'événements par cas, la similitude entre les cas, le nombre d'événements uniques, et le nombre de chemins uniques. Considérons la traces d'exécution *L1* avec les caractéristiques suivantes : 1000 instance, 10 événements par cas en moyenne, et peu de variations (par exemple, plusieurs instance suivent les mêmes chemins ou des chemins très similaires). La trace d'exécution *L2* contient seulement 100 instances, mais il y a en moyenne 100 événements par instance et toutes les instances suivent un chemin unique. De toute évidence, *L2* est beaucoup plus difficile à analyser que *L1*, même si les deux traces d'exécution ont des tailles similaires (environ 10 000 événements).

Comme les traces d'exécution contiennent un comportement exemple, ils ne devraient pas être considérées comme complets. Les techniques de fouille de procédé doivent traiter l'incomplétude en utilisant une "hypothèse du monde ouvert" : le fait que quelque chose ne s'est pas produit ne signifie pas qu'il ne peut pas arriver. Cela rend difficile le traitement des petites traces d'exécution avec beaucoup de variabilité.

Comme mentionné précédemment, certaines traces d'exécution contiennent des événements à un niveau d'abstraction très bas. Ces traces d'exécution ont tendance à être extrêmement large et les événements individuels de bas niveau sont d'intérêt minimes pour les intervenants. Par conséquent, on aimerait agréger les événements de faible niveau en événements de haut niveau. Par exemple, en analysant les procédés de diagnostic et de traitement d'un groupe particulier de patients, on peut ne pas être intéressé par les tests individuels enregistrés dans le système d'information du laboratoire de l'hôpital.

Actuellement, les organisations ont besoin d'utiliser une approche par "tests et erreurs" pour voir si des traces d'exécution sont adaptées à la fouille de procédé. Par conséquent, les outils devraient permettre un test rapide de la faisabilité étant donné un ensemble de données particulier. Un tel test devrait indiquer des problèmes de performances potentiels et détecter des traces d'exécution qui sont loin d'être complètes ou trop détaillées.

4.3 D3 : Création de Banc de Tests Représentatifs

La fouille de procédé est une technologie émergente. Cela explique pourquoi des bons banc de tests manquent toujours. Par exemple, des douzaines de techniques de découverte de procédé sont disponibles et différents fournisseurs offrent des produits variés, mais il n'y a pas de consensus sur la qualité de ces techniques. Bien qu'il existe d'énormes différences dans les fonctionnalités et performances, il est difficile de comparer les différentes techniques et outils. Par conséquent, de bons banc de tests composés d'ensembles de données exemples et des critères de qualité représentatifs doivent être développés.

Pour les techniques classiques de fouille de données, de nombreux bons bancs de tests sont disponibles. Ces bancs de tests ont des outils de simulation proposés par des chercheurs et des fournisseurs pour améliorer les performances de leurs techniques. Dans le cas de la fouille de procédé ceci est plus difficile. Par exemple, le modèle relationnel introduit par Codd en 1969 est simple et largement supporté. En conséquence, peu d'efforts est consentit pour convertir les données d'une base à une autre et il n'y a pas de problèmes d'interprétation. Pour les procédés un tel modèle simple est manquant. Les standards proposés pour la modélisation des procédés sont beaucoup plus compliqués et peu de fournisseurs supportent exactement le même ensemble de concepts. Les procédés sont tout simplement plus complexes que des données tabulaires.

Néanmoins, il est important de créer des bancs de tests représentatifs pour la fouille de procédé. Certains travaux préliminaires sont déjà disponibles. Par exemple, il existe diverses métriques

pour mesurer la qualité des résultats de la fouille de procédé (adéquation, simplicité, précision et généralisation). Par ailleurs, plusieurs traces d'exécution sont disponibles publiquement (cf. www.processmining.org). On peut voir par exemple les traces d'exécution utilisées pour le premier Business Process Intelligence Challenge (BPIC'11) organisé par le groupe de travail (cf. doi :10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54)).

D'une part, il devrait y avoir des bancs de tests basés sur des ensembles de données réelles. D'autre part, il y a la nécessité de créer des ensembles de données synthétiques capturant des caractéristiques particulières. Ces données synthétiques aident à développer des techniques de fouille de procédé qui sont adaptés pour des traces d'exécution incomplètes, des traces d'exécution contenant du bruit, ou des populations spécifiques de procédé.

Outre la création de banc de tests représentatifs, il doit également y avoir un plus grand consensus sur les critères utilisés pour juger de la qualité des résultats de fouille de procédés (voir aussi Défi D4). Par ailleurs, la *validation croisée* des techniques de fouille de données peut être adaptée pour évaluer le résultat. Considérons par exemple une vérification en k couches. On peut diviser les traces d'exécution en k parties. Les $k - 1$ parties peuvent être utilisées pour apprendre un modèle de procédé, et les techniques de vérification de conformité peuvent être utilisées pour évaluer le résultat à l'égard de la partie restante. Ceci peut être répété k fois, fournissant ainsi un aperçu sur la qualité du modèle.

4.4 D4 : Traiter la Dérive de Concept

Le terme *dérive de concept* renvoie à la situation dans laquelle le procédé se transforme pendant son analyse. Par exemple, dans le début de la trace d'exécution deux activités peuvent être en concurrence alors que plus tard dans la trace d'exécution ces activités deviennent séquentielles. Les procédés peuvent changer en raison des changements périodiques/saisonniers (par exemple, "en Décembre il y a plus de demandes" ou "le vendredi après-midi il y a moins d'employés disponibles") ou en raison de l'évolution des conditions (par exemple, "le marché devient plus concurrentiel"). De tels changements impactent sur les procédés et il est essentiel de les détecter et de les analyser. La dérive de concept dans un procédé peut être découverte par fractionnement de la traces d'exécution en petites traces et l'analyse des "empreintes" de traces plus petites. Cette analyse de "second ordre" nécessite beaucoup plus de traces d'exécution. Néanmoins, peu de procédés sont en état constant et la compréhension de la dérive de concept est d'une importance primordiale pour la gestion des procédés. Par conséquent, une recherche et des outils de support supplémentaires sont nécessaires pour analyser adéquatement la dérive de concept.

4.5 D5 : Amélioration du Biais de Représentation Utilisé dans la Découverte de Procédé

Une technique de découverte de procédé produit un modèle en utilisant un langage particulier (par exemple, BPMN ou les réseaux de Petri). Cependant, il est important de séparer la visualisation du résultat de la représentation utilisée pendant la découverte réelle de procédé. Le choix d'un langage cible englobe souvent plusieurs hypothèses implicites. Il limite l'espace de recherche, les procédés qui ne peuvent pas être représentés par le langage cible ne peuvent pas être découverts. Ce soi-disant "biais de représentation" utilisé pendant la découverte de procédé doit être un choix conscient et ne doit pas être (seulement) influencé par la représentation graphique préférée.

Considérons par l'exemple de Figure 6 : selon que le langage cible permet la concurrence ou pas on peut avoir un effet sur la visualisation du modèle découvert et la classe des modèles considérés par l'algorithme. Si le biais de représentation ne permet pas de concurrence (Figure 6(a) n'est pas possible) et ne permet pas d'activités multiples ayant le même nom (Figure 6(c) n'est pas possible), alors seulement des modèles problématiques telles que celui indiqué dans Figure 6(b) sont possibles. Cet exemple montre qu'une sélection plus rigoureuse et raffinée du biais de représentation est nécessaire.

4.6 D6 : Équilibre entre les Critères de Qualité tels que la Justesse, la Simplicité, la Précision, et la Généralisation

Les traces d'exécution sont souvent loin d'être complètes, c'est à dire, seulement un comportement exemple est donné. Les modèles de procédé permettent généralement un nombre exponentiel ou même infini de traces différentes (en cas de boucles). Par ailleurs, quelques traces peuvent avoir une probabilité beaucoup plus faible que les autres. Par conséquent, il est irréaliste de supposer que toute trace possible est présente dans les événements de trace d'exécution. Pour illustrer qu'il est impossible d'avoir comme acquis des traces d'exécution complètes, considérons un procédé composé de 10 activités qui peuvent être exécutées en parallèle et d'une trace d'exécution correspondante qui contient des informations sur 10.000 instances. Le nombre total des entrelacements possibles dans le modèle avec 10 activités concurrentes est de $10! = 3.628.800$. Par conséquent, il est impossible que chaque entrelacement soit présent dans la trace d'exécution car il y a moins de (10.000) instance que des traces possibles (3628800). Même s'il y a des millions d'instances dans la trace d'exécution, il est extrêmement peu probable que toutes les variations possibles soient présentes. Une complication supplémentaire est que certaines alternatives sont moins fréquentes que d'autres. Cela peut être considéré comme du "bruit". Il est impossible de construire un modèle raisonnable pour de tels comportements comportant du bruit. Le modèle découvert a besoin de faire abstraction de cela, il est préférable d'étudier le comportement à basse fréquence en utilisant la vérification de conformité.

Le bruit et l'incomplétude font de la découverte des procédés un problème difficile. En fait, il y a quatre dimensions de qualité concurrentes : (a) justesse, (b) simplicité, (c) précision, et (d) généralisation. Un modèle avec une bonne *justesse* permet la plupart des comportements vu dans la trace d'exécution. Un modèle a une justesse parfaite, si toutes les instances de traces peuvent être rejouées par le modèle du début à la fin. Le plus simple modèle qui peut expliquer le comportement vu dans la trace d'exécution est le meilleur modèle. Ce principe est connu comme le rasoir d'Occam. La justesse et la simplicité ne sont pas suffisantes pour juger de la qualité d'un modèle de procédé découvert. Par exemple, il est très facile de construire un réseau de Petri extrêmement simple ("modèle de fleur") qui est capable de rejouer toutes les instances dans une trace d'exécution (mais aussi toute autre trace d'exécution se référant au même ensemble d'activités). De même, il est souhaitable d'avoir un modèle qui ne permet que le comportement exact vu dans les événements de la trace d'exécution. Rappelons que la trace d'exécution contient des comportements exemples et que de nombreuses instances possibles peuvent ne pas avoir été encore vu. Un modèle est *précis* s'il ne permet pas "trop" de comportement. De toute évidence, le "modèle de fleur" manque de précision. Un modèle qui n'est pas précis est "sous-ajusté". Le problème de sous-ajustement est que le modèle sur-généralise le comportement exemple dans la trace d'exécution (c'est à dire, le modèle permet des comportements très différents de ce qu'il a constaté dans sa trace d'exécution). Un modèle devrait généraliser et ne pas restreindre le comportement simplement à des instances vues dans la trace d'exécution. Un modèle qui ne se généralise pas est "surdimensionné". Le surdimensionnement est le problème qu'un modèle très spécifique est généré alors qu'il est évident que la trace d'exécution ne détient qu'un comportement exemple (c'est à dire, le modèle explique l'échantillon particulier de la trace d'exécution, mais un échantillon futur du même procédé peut produire un modèle de procédé complètement différent).

L'équilibre entre justesse, simplicité, précision et généralisation est difficile. C'est la raison pour laquelle la plupart des techniques les plus puissantes de découverte de procédé fournissent divers paramètres. L'amélioration des algorithmes doit être développée afin de mieux équilibrer les quatre dimensions de qualité en compétition. Par ailleurs, tous les paramètres utilisés doivent être compréhensibles par les utilisateurs finaux.

4.7 D7 : Fouille Inter-Organisationnelle

Traditionnellement, la fouille de procédé est appliquée au sein d'une seule organisation. Cependant, comme l'approche service, l'intégration de la chaîne logistique, et le cloud computing

deviennent plus répandus, il existe des scénarios où les traces d'exécution d'organisations multiples sont disponibles pour analyse. En principe, il y a deux cadres pour une fouille de *procédés inter-organisationnels*.

Tout d'abord, nous pouvons envisager un cadre collaborative où des organisations différentes travaillent ensemble pour gérer des instances de procédé. On peut considérer un tel procédé inter-organisationnel comme un "puzzle", c'est à dire l'ensemble du procédé est coupé en parties et réparti sur les organisations qui ont besoin de coopérer pour terminer avec succès l'exécution des instances de procédé. Analyser les traces d'exécution au sein de l'une de ces organisations est insuffisant. Pour découvrir de bout en bout les procédés, les traces d'exécution des différentes organisations ont besoin d'être fusionnées. C'est une tâche non triviale vu que les événements doivent être corrélés à travers les frontières organisationnelles.

Deuxièmement, nous pouvons également envisager le cadre, où différentes organisations exécutent essentiellement le même procédé, tout en partageant des expériences, des connaissances, ou une infrastructure commune. Considérons par exemple Salesforce.com. Les procédés de vente de nombreuses organisations sont gérés et supportés par Salesforce. D'une part, ces organisations partagent une infrastructure (procédés, bases de données, etc.). D'autre part, ils ne sont pas obligés de suivre un modèle de procédé strict comme le système peut être configuré pour supporter des variantes du même procédé. Comme autre exemple, considérons le procédé de base exécuté dans toute municipalité (par exemple, la délivrance des permis de construction). Bien que toutes les municipalités dans un pays ont besoin de supporter le même ensemble de base de procédé, il peut y avoir aussi des différences. Évidemment, il est intéressant d'analyser ces variations entre les différentes organisations. Ces organisations peuvent apprendre les unes des autres et des fournisseurs de services peuvent améliorer leurs services et offrir des services à valeur ajoutée basés sur les résultats de la fouille de procédé inter-organisationnels.

Des nouvelles techniques d'analyse doivent être développées pour les deux types de fouille de procédés inter-organisationnels. Ces techniques devraient aussi envisager la vie privée et les questions de sécurité. Les organisations ne veulent pas partager les informations pour des raisons concurrentielles ou en raison d'un manque de confiance. Par conséquent, il est important de développer des techniques de fouille de procédé préservant la vie privée.

4.8 D8 : Fournir un Support Opérationnel

Initialement, l'objectif de la fouille de procédé a été l'analyse des données des traces d'exécution. Aujourd'hui, cependant, de nombreuses sources de données sont mises à jour en (presque) temps réel et la puissance de calcul disponible est suffisante pour analyser les événements quand ils se produisent. Par conséquent, la fouille de procédé ne devrait pas se limiter à l'analyse hors ligne et peut aussi être utilisée pour le support opérationnel en ligne. Trois activités de support opérationnel peuvent être identifiées : *détecter*, *prédire*, et *recommander*. Le moment où une instance s'écarte de la procédure prédéfinie peut être détecté et le système peut générer une alerte. Souvent on aimerait générer de telles notifications immédiatement (pour être encore en mesure d'influencer les choses) et non pas de façon off-line. Les traces d'exécution peuvent être utilisées pour construire des modèles prédictifs. Ceux-ci peuvent être utilisés pour guider l'exécution d'instances de procédé. Par exemple, il est possible de prédire le temps de traitement restant d'une instance. Basé sur de telles prédictions, on peut aussi construire des systèmes de recommandation qui proposent des actions particulières pour réduire les coûts ou raccourcir le temps d'exécution. Appliquer des techniques de fouille de procédé dans un tel environnement en ligne pose des défis supplémentaires en terme de puissance de calcul et de qualité des données.

4.9 D9 : Combiner la Fouille de Procédé avec d'Autres Types d'Analyse

La opérations de gestion, et en particulier les opérations de recherche, représente une branche des sciences de gestion s'appuyant lourdement sur la modélisation. Ici une variété de modèles mathématiques allant de la programmation linéaire et la planification de projet à des modèles de files d'attente, les chaînes de Markov et la simulation sont utilisés. La fouille de données peut

être définie comme “l’analyse d’ensembles (souvent grands) de données pour trouver des relations insoupçonnées et résumer les données de façon novatrice à la fois compréhensible et utile pour le propriétaire des données”. Une grande variété de techniques a été développée : la classification (par exemple, l’apprentissage par arbre de décision), la régression, le clustering (par exemple, k-means clustering) et la découverte de patrons (par exemple, l’apprentissage de règles d’association).

Les deux domaines (gestion des opérations et fouille de données) fournissent des techniques d’analyse précieuses. Le défi est de combiner les techniques dans ces domaines à la fouille de procédé. Considérons par exemple la simulation. Les techniques de fouille de procédé peuvent être utilisées pour découvrir un modèle de simulation basé sur des traces d’exécution. Par la suite, le modèle de simulation peut être utilisé pour fournir un support opérationnel. En raison du lien étroit entre les traces d’exécution et le modèle, le modèle peut être utilisé pour rejouer l’historique d’exécution et on peut commencer des simulations de l’état actuel offrant ainsi un “bouton d’avance rapide” vers l’avenir en se basant sur les données actuelles.

De même, il est souhaitable de combiner la fouille de procédé avec *l’analyse visuelle*. L’analyse visuelle combine une analyse automatisée avec des visualisations interactives pour une meilleure compréhension des ensembles de données volumineux et complexes. L’analyse visuelle exploite les étonnantes capacités de l’être humain à voir des patrons dans des données non structurées. En combinant des techniques de fouille de procédé avec l’analyse visuelle interactive, il est possible d’extraire plus de connaissances des traces d’exécution.

4.10 D10 : Améliorer l’ergonomie pour les non-experts

Un des objectifs de la fouille de procédé consiste à créer “des modèles de procédé vivants”, c’est à dire, des modèles de procédé qui sont utilisés sur une base quotidienne plutôt que des modèles statiques qui finissent dans une archive. De nouvelles traces d’exécution peuvent être utilisées pour découvrir les comportements émergents. Le lien entre les traces d’exécution et les modèles de procédé permet la projection de l’état actuel et les activités récentes sur des modèles mis à jour. Ainsi, les utilisateurs finaux peuvent interagir avec les résultats de fouille de procédé sur une base quotidienne. Ces interactions sont d’une très grande valeur, mais exigent aussi des interfaces utilisateur intuitives. Le défi est de cacher les algorithmes sophistiqués de fouille de procédés derrière des interfaces conviviales qui fixent automatiquement les paramètres et proposent des types d’analyse appropriés.

4.11 D11 : Améliorer la compréhension pour les non-experts

Même s’il est facile de générer des résultats de fouille de procédés, cela ne signifie pas que les résultats sont réellement utiles. L’utilisateur peut avoir des problèmes de compréhension de l’entrée ou de la sortie et il peut être tenté d’en déduire des conclusions erronées. Pour éviter ces problèmes, les résultats devraient être présentés en utilisant une représentation appropriée (voir aussi PD5). Par ailleurs, la fiabilité des résultats doit toujours être clairement indiquée. Il peut y avoir trop peu de données pour justifier des conclusions particulières. En fait, les techniques existantes de découverte de procédé n’avertissent pas d’un niveau bas de justesse ou de sur-dimensionnement. Elles montrent toujours un modèle, même quand il est clair qu’il y a trop peu de données pour justifier des conclusions.

5 Épilogue

La Task Force IEEE sur la fouille de procédé vise à (a) promouvoir l’application de la fouille de procédé, (b) guider les développeurs de logiciels, consultants, chefs d’entreprise, et utilisateurs finaux lors de l’utilisation des techniques introduites dans l’état de l’art, et (c) stimuler la recherche sur la fouille de procédé. Ce manifeste énonce les grands principes et les intentions de la Task Force. Après avoir introduit la fouille de procédé, le manifeste énonce quelques principes directeurs (Section 3) et défis (Section 4). Les principes directeurs peuvent être utilisés afin d’éviter des erreurs

évidentes. La liste des défis est destinée à orienter les efforts de recherche et de développement. Les deux visent à accroître le niveau de maturité de la fouille de procédé.

Pour conclure, nous introduisons quelques mots sur la terminologie. Les termes suivants sont utilisés dans le domaine de fouille de procédé : la fouille de workflow, la fouille de procédé (métier), la découverte automatisée de procédé (métier), et l'intelligence de procédé (métier). Différentes organisations semblent utiliser des termes différents pour des concepts qui se chevauchent. Par exemple, Gartner fait la promotion du terme “Automated Business Process Discovery” (ABPD) et Software AG utilise “Process Intelligence” pour se référer à leur plate-forme de contrôle. Le terme “workflow mining” semble moins adapté comme la création de modèles de workflow n’est qu’une des nombreuses applications possibles de la fouille de procédé. De même, l’ajout du terme “métier” réduit la portée à certaines applications de fouille de procédé. Il y a de nombreuses applications de la fouille de procédé (par exemple, l’analyse de certains systèmes de haute technologie ou l’analyse de sites web) où cet ajout semble inapproprié. Bien que la découverte de procédé représente une partie importante du spectre de la fouille de procédé, elle est seulement un des nombreux cas d’utilisation. La vérification de conformité, la prédiction, la fouille organisationnelle, l’analyse des réseaux sociaux, etc. sont d’autres cas d’utilisation qui s’étendent au-delà de la découverte de procédés.

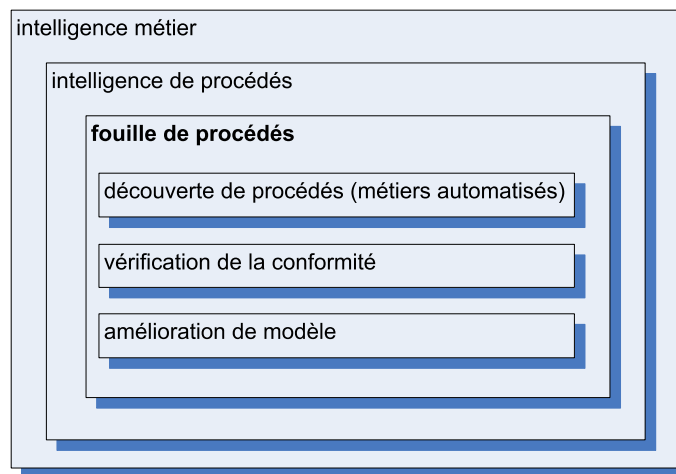


Figure 7. Relier les différents termes.

Figure 7 relate certains des termes mentionnés. Toutes les technologies et les méthodes qui visent à fournir des informations exploitables et qui peuvent être utilisées pour appuyer la prise de décision peuvent être placées sous l’égide de l’intelligence métier (Business Intelligence (BI)). L’intelligence de procédés ((Business) Process Intelligence) peut être vue comme la combinaison de BI et BPM, c’est à dire, les techniques de BI sont utilisées pour analyser et améliorer les procédés et leur gestion. La fouille de procédé peut être vue comme une concrétisation de l’intelligence de procédé prenant en compte les traces d’exécution comme point de départ. La découverte de procédés ((Automated business) process discovery) est juste l’un des trois types de base de fouille de procédé. Figure 7 peut être un peu trompeuse dans la mesure où la plupart des outils BI ne fournissent pas la fonctionnalité de fouille de procédé, comme elle est décrite dans ce document. Le terme BI est souvent et commodément biaisé en faveur d’un outil ou d’une méthode particulière couvrant seulement une petite partie du plus large spectre de BI.

Il peut y avoir des raisons commerciales pour l’utilisation de termes alternatifs. Certains fournisseurs peuvent aussi vouloir souligner un aspect particulier (par exemple, la découverte ou l’intelligence). Toutefois, pour éviter toute confusion, il est préférable d’utiliser le terme “fouille de procédé” pour la discipline couverte par le présent manifeste.

Glossaire

- **Activité** : une étape bien définie dans le procédé. Les événements peuvent se référer au début, terminaison, annulation, etc. d'une activité pour une instance spécifique de procédé.
- **Amélioration de modèle** : l'un des trois types de base de la fouille de procédé. Un modèle de procédé est étendu ou amélioré en utilisant des informations extraites des traces d'exécution. Par exemple, les goulots d'étranglement peuvent être identifiés en rejouant une trace d'exécution sur un modèle de procédé, tout en examinant l'horodatage.
- **Biais de représentation** : le langage cible sélectionné pour représenter et modéliser les résultats de la fouille de procédé.
- **Business Intelligence (BI)** : une large collection d'outils et de méthodes qui utilisent des données pour appuyer la prise de décision.
- **Business Process Intelligence** : voir **Intelligence de procédé**.
- **Business Process Management (BPM)** : la discipline qui associe des connaissances issues des systèmes d'information et des connaissances issues des sciences de gestion et qui applique les deux aux procédés opérationnels.
- **Découverte automatique de procédé métiers** : voir **découverte de procédé**.
- **Découverte de procédé** : l'un des trois types de base de la fouille de procédé. Basé sur des traces d'exécution, un modèle de procédé est découvert. Par exemple, α algorithm est capable de découvrir un réseau de Petri en identifiant les patrons de procédé dans des collections d'événements.
- **Dérive de concept** : le phénomène où le procédé change souvent au fil du temps. Le procédé observé peut progressivement (ou brutalement) changer en raison des changements saisonniers ou de la concurrence accrue, ce qui complique l'analyse.
- **Événement** : une action enregistrée dans une trace d'exécution, par exemple, le début, la terminaison ou l'annulation d'une activité pour une instance de procédé particulier.
- **Fouille de données** : l'analyse d'ensembles (souvent grands) de données pour trouver des relations inattendues et pour résumer les données de façon à apporter de nouveaux éclairages.
- **Fouille de procédé** : techniques, outils et méthodes pour découvrir, surveiller et améliorer les procédés réels (c'est à dire, des procédés non explicitement définis) par extraction de connaissances à partir des traces d'exécution couramment disponibles dans les systèmes (d'information) actuels.
- **Fouille de procédé inter-organisationnels** : l'application de techniques de fouille de procédé pour les traces d'exécution provenant de différentes organisations.
- **Généralisation** : une métrique pour déterminer à quelle mesure le modèle permet des comportements non observés. Un modèle "surdimensionné" n'est pas en mesure de généraliser suffisamment.
- **Instance** : voir **Instance de procédés**.
- **Instance de procédé** : l'entité gérée par le procédé qui est analysé. Les événements font référence aux instances de procédé. Des exemples d'instances de procédé sont les commandes des clients, les réclamations d'assurance, les demandes de prêt, etc.
- **Intelligence de procédé** : une branche de l'Intelligence Métier se concentrant sur le Business Process Management.
- **Justesse** : une métrique déterminant dans quelle mesure un modèle donné permet le comportement vu dans la trace d'exécution. Un modèle a une justesse parfaite, si toutes les instances des traces d'exécution peuvent être rejouées par le modèle du début à la fin.
- **MXML** : un format basé sur XML pour échanger des traces d'exécution. XES remplace MXML comme nouvel format indépendant des outils de fouille de procédé.
- **Précision** : une mesure déterminant si le modèle interdit un comportement très différent du comportement vu dans la trace d'exécution. Un modèle avec une faible précision est "sous-dimensionné".
- **Support opérationnel** : une analyse en ligne des traces d'exécution dans le but de surveiller et d'influencer les instances de procédés en cours d'exécution. Trois activités de support opérationnel peuvent être identifiées : détecter (générer une alerte si le comportement observé

s'écarter du comportement modélisé), prévoir (prédire le comportement futur basé sur le comportement passé, par exemple, prédire le temps de traitement restant), et recommander (suggérer des actions appropriées pour réaliser un objectif particulier, par exemple, pour minimiser les coûts).

- **Trace d'exécution** : collection d'événements utilisés comme entrée pour la fouille de procédé. Les événements n'ont pas besoin d'être stockés dans un fichier séparé (par exemple, les événements peuvent être dispersés sur des tables de bases de données différentes).
- **Simplicité** : une mesure opérationnelle du rasoir d'Occam, c'est à dire, le modèle le plus simple qui peut expliquer le comportement vu dans la trace d'exécution, est le meilleur modèle. La simplicité peut être quantifiée de diverses façons, par exemple, le nombre de nœuds et d'arcs dans le modèle.
- **Vérification de Conformité** : analyser si la réalité, comme elle est enregistrée dans une trace d'exécution, est conforme au modèle et vice versa. L'objectif est de détecter les anomalies et de mesurer leur gravité. La vérification de conformité est l'une des trois types de base de la fouille de procédés.
- **XES** : est un standard basé sur XML pour les traces d'exécution. La norme a été adoptée par la Task Force IEEE sur la fouille de procédé comme format d'échange par défaut pour les événements de traces d'exécution (cf. www.xes-standard.org).