

République Algérienne démocratique et populaire

Ministère de l'enseignement supérieur

Et de la recherche scientifique

Université 20 Août 1955 SKIKDA

Département d'Informatique



Mémoire de fin d'étude en vue de l'obtention du diplôme

Master Académique -Option : Systèmes d'Information

Avancées et Applications (SIAA)

THEME :

**Approche Ontologique pour la Standardisation des
Échanges de Données en Industrie 4.0 : Un Cadre Basé
sur CAEX, PLCopen XML et les Diagrammes de Gantt**

Réalisé par :

- BELLIR Khouloud
- GOUASMIA Asma

Encadré par :

Dr. BOUARROUDJ Samiya

Session : Juin 2024

REMERCIEMENTS

Nous voulons exprimer par ces quelques lignes de remerciements notre gratitude envers tous ceux en qui par leur présence, leur soutien, leur disponibilité et leurs conseils, nous avons eu courage d'accomplir ce travail. Nous commençons par remercier « Dieu » le tout puissant de nous avoir donné la force de commencer et de finir ce travail. Nous tenons à remercier également « Bouarroudj Samiya » qui nous a fait l'honneur d'être notre encadrante. Nous la remercions profondément pour son encouragement continue et aussi d'être toujours là pour nous écouter, nous aider et nous guider à retrouver le bon chemin par sa sagesse et ses précieux conseils, ainsi que son soutien moral et sa preuve de compréhension ce qui nous a donné la force et le courage d'achever ce projet. Nos remerciements les plus sincères s'adressent de même à l'équipe du service instrumentation du complexe CP2K. Nous tenons d'autre part à remercier les respectables membres de jury pour bien vouloir nous accorder de leurs temps précieux pour commenter, discuter et juger notre travail. En fin nous ne pouvons achever ce mémoire sans exprimer notre gratitude à nos parents, nos frères et nos sœurs sont les quels rien ne serait possible, aussi tous les professeurs du département informatique, pour leur dévouement et leur assistance tout au long de nos études universitaires.

Merci à tous.

Dédicace

Je dédie ce mémoire A mes chers parents et mon mari qui m'ont toujours soutenu et encouragé au cours de ces années d'études. Qu'ils trouvent ici le témoignage de ma profonde de reconnaissance.

A ma petite princesse « Mayassine », mes sœurs, mes frères et ceux qui ont partagé avec moi tous les moments d'émotions lors de la réalisation de ce travail, Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours. A ma famille, mes proches et à ceux qui me donnent de la vivacité.

Khouloud

Dédicace

Je dédie ce modeste travail avec le profond remerciement aux êtres les plus chères à mon cœur : À ma chère mère qui jouait le plus grand rôle dans ma réussite avec ses conseils et son aide et de m'avoir donné le courage et la patience pour que ce travail soit achevé, elle est pour moi la source d'amour, que dieu la protège et la garde pour moi. Je dédie aussi ce travail à mon père et mes chères sœurs. À toute ma famille du plus grand au plus petit. À mes chères amies sans exception. À tous mes professeurs qui m'enseignent.

Asma

Glossaire

IoT : Internet of Things

CPS : Cyber Physical Systems

AML: Automation Markup Language

SUC: System Unit Class

RC: Role Class

IC: Interface Class

IE: Internal Element

CAEX: Computer Aided Engineering Exchange

PLC : Programmable Logic Controllers

SFC : Sequential Function Charts

IML : Intermediate Modeling Layer

XML: Extensible Markup Language

OWL: Ontology Web Language

DL: Description Logic

JESS: Java Expert System Shell

RACER: Renamed Abox and Concept Expression Reasoner

SBC : Système à Base de Connaissances

SWRL : Semantic Web Rule Language

Résumé

Afin de remédier à l'hétérogénéité technique, il est essentiel de mener à bien le processus de standardisation technologique, en mettant particulièrement l'accent sur l'utilisation des ontologies et d'AutomationML (AML). Les ontologies jouent un rôle essentiel dans le contexte de la perspective sémantique de l'Internet des objets (IoT). L'AML génère un fichier CAEX et importe un fichier PLCopen XML, formant ainsi un socle pour standardiser les échanges de données techniques au sein des projets de l'industrie 4.0. Pour garantir l'interopérabilité, l'effort de normalisation ontologique repose sur l'utilisation des ontologies standards et englobe un ensemble d'ontologies supérieures spécifiquement conçues pour les domaines de la robotique, de l'automatisation et des CPS.

Cette étude vise à élaborer une ontologie OWL expressive à partir des données contenues dans les documents CAEX, PLCopen XML et les diagrammes de Gantt. Des algorithmes détaillés ont été proposés pour effectuer la correspondance entre ces catégories de documents et les ontologies OWL. L'ontologie supérieure de l'industrie 4.0 AML a été étendue pour incorporer des concepts supplémentaires spécifiques à la chaudière industrielle.

Mot clés

IoT, Industrie 4.0, AutomationML, CAEX, PLCopenXML, Ontologies, Diagramme de Gantt.

Abstract

In order to overcome technical heterogeneity, it is essential to complete the process of technological standardization, with particular emphasis on the use of ontologies and AutomationML (AML). Ontologies play an essential role in the semantic perspective of the Internet of Things (IoT). AML generates a CAEX file and imports a PLCopen XML file, forming a basis for standardizing the exchange of technical data within Industry 4.0 projects. To guarantee interoperability, the ontological standardization effort is based on the use of standard ontologies and encompasses a set of superior ontologies specifically designed for the robotics, automation and CPS domains.

This study aims to build an expressive OWL ontology from the data contained in CAEX, PLCopen XML and Gantt chart documents. Detailed algorithms have been proposed to map these document categories to OWL ontologies. The superior Industry 4.0 ontology AML has been extended to incorporate additional concepts specific to the industrial boiler.

Keywords

IoT, Industry 4.0, AutomationML, CAEX, PLCopenXML, Ontologies, Gantt chart.

Table des matières

| | |
|------------------------------------------------------------------------|----|
| Introduction générale | 1 |
| 1 Contexte de la recherche..... | 1 |
| 2 Problématique | 2 |
| 3 Contribution..... | 2 |
| 4 Organisation du document | 3 |
| Chapitre 01 : L'Industrie 4.0 | 5 |
| 1 Introduction..... | 5 |
| 2 Définition et Objectifs | 5 |
| 3 Principes de conception de l'industrie 4.0..... | 7 |
| 4 Internet des objets | 8 |
| 5 Systèmes cyber-physiques | 9 |
| 6 L'impact positif de l'industrie 4.0 sur la production | 9 |
| 7 Conclusion | 10 |
| Chapitre 02 : Ontologies Supérieures dans l'Industrie 4.0 | 11 |
| 1 Introduction..... | 11 |
| 2 Web Sémantique | 11 |
| 3 Notion d'Ontologie..... | 13 |
| 3.1 Composants d'une ontologie | 13 |
| 3.2 Rôle d'ontologies | 14 |
| 3.3 Langages d'ontologies..... | 14 |
| 3.4 Types d'ontologie | 15 |
| 3.3.1 Ontologie de domaine | 15 |
| 3.3.2 Ontologie de tache | 15 |
| 3.3.3 Ontologies d'application | 15 |
| 3.3.4 Ontologies supérieures..... | 16 |
| 3.3.5 Exemples d'ontologies supérieures..... | 16 |
| 3.3.5.1 Ontologie AML | 16 |
| 3.3.5.2 Ontologie sémantique de la fabrication (MASON)..... | 17 |
| 3.3.5.3 Langage de description des services de la machine (MSDL) | 19 |
| 3.3.5.4 RAMI..... | 20 |
| 4 Conclusion | 22 |
| Chapitre 3 : AutomationML et La Chaudière Industrielle | 23 |
| Partie 1 :AutomationML | 23 |
| 1 Introduction..... | 23 |

| | | |
|-----------|----------------------------------------------------------------|-----------|
| 2 | Architecture générale d'AML..... | 23 |
| 3 | La structure CAEX | 25 |
| 4 | Relations entre objets CAEX..... | 27 |
| 5 | Stockage des descriptions logiques dans AutomationML | 29 |
| 6 | Modèles de séquence et de comportement dans AutomationML | 30 |
| 6.1 | Les graphiques de Gantt | 31 |
| 6.2 | Les graphiques de PERT | 32 |
| 6.3 | Les diagrammes d'impulsion | 33 |
| 6.4 | Diagramme d'état..... | 34 |
| 6.5 | Les graphiques de fonctions séquentielles | 35 |
| 7 | Conclusion | 35 |
| | Partie 2 : La chaudière Industrielle..... | 36 |
| 1 | Introduction..... | 36 |
| 2 | Présentation du Complexe Pétrochimique CP2K..... | 36 |
| 2.1 | Implantation de l'usine..... | 36 |
| 2.2 | Description de l'unité CP2K SKIKDA | 37 |
| 2.3 | Présentation de l'unité utilité..... | 37 |
| 2.3.1 | Section d'eau..... | 37 |
| 2.3.1.1 | Prise de l'eau | 37 |
| 2.3.1.2 | Distillation de l'eau de Mer | 38 |
| 2.3.1.3 | Déminéralisation de l'eau..... | 38 |
| 2.3.1.4 | Traitement des effluents | 38 |
| 2.3.2 | Section d'énergie..... | 38 |
| 2.3.2.1 | Génération de vapeur (La Chaudière) | 38 |
| 2.3.2.2 | Les principaux composants de la chaudière | 39 |
| 2.3.2.2.1 | Les équipements | 39 |
| 2.3.2.2.2 | Les instruments..... | 41 |
| 3 | Indicateur de température | 42 |
| 4 | Séquence de mise en service chaudière | 43 |
| 4.1 | Préventilation | 43 |
| 5 | Le schéma de principe de fonctionnement de l'unité Eau | 46 |
| 6 | Conclusion | 46 |
| | Chapitre 4 : Conception | 47 |
| 1 | Introduction..... | 47 |
| 1.1 | Objectif du travail | 47 |
| 1.2 | Travaux existants | 48 |

| | | |
|---------|--------------------------------------------------------------------------------------------------------------|-----------|
| 2 | Architecture du système de création semi-automatique d'ontologie à partir des documents CAEX et PLCopen..... | 49 |
| 2.1 | Phase de développement de l'ontologie de domaine « Boiler.OWL »..... | 51 |
| 2.1.1 | Construction de la hiérarchie des concepts et les relations binaires..... | 51 |
| 2.1.1.1 | Construction du dictionnaire des concepts..... | 53 |
| 2.1.1.2 | Construction de la table des relations binaires..... | 57 |
| 2.1.1.3 | Construction de la table des instances des concepts..... | 59 |
| 2.1.1.4 | Représentation sémantique des activités d diagramme de GANTT avec le langage SWRL | 61 |
| 2.1.2 | Phase de description AML Statique des composants matériels..... | 63 |
| 2.1.2.1 | Création d'un document AML de la chaudière | 63 |
| 2.1.3 | Phase de description du comportement dynamique | 70 |
| 2.1.3.1 | Modélisation de diagrammes de Gantt..... | 70 |
| 2.2 | Le mapping de GANTT vers le format IML..... | 72 |
| 2.2.1 | Présentation des éléments du système IML | 73 |
| 3 | Additional Data..... | 74 |
| 3.1.1 | Propriétés et relations de l'élément | 74 |
| 3.2 | Le mapping de l'IML vers le format XML de PLCopen | 76 |
| 3.3 | Mapping automatique CAEX-OWL | 77 |
| 4 | Conclusion | 79 |
| | Chapitre 05 : Implémentation..... | 80 |
| 1 | Introduction..... | 80 |
| 2 | Programmation XML..... | 80 |
| 2.1 | Les parseurs XML..... | 80 |
| 2.2 | Sax..... | 80 |
| 2.3 | API DOM | 81 |
| 3 | Langage de programmation java | 81 |
| 4 | Environnement de développement | 81 |
| 4.1 | Protégé..... | 81 |
| 4.2 | JCreator | 82 |
| 4.3 | L'API JENA | 83 |
| 5 | AML | 83 |
| 6 | Editeur SWRL | 83 |
| 7 | Quelques interfaces de notre simulateur..... | 84 |
| 8 | Conclusion | 88 |
| | Conclusion et perspectives..... | 89 |
| | Bibliographie | |

Liste de figures

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1: Paramètres horizontaux et verticaux des scénarios I4.0 | 6 |
| Figure 1.2: Principes de conception de I4.0 | 7 |
| Figure 1.3: Périmètre de l'industrie 4.0 | 8 |
| Figure 2.1: Vision du web sémantique | 12 |
| Figure 2.2: Architecture du web sémantique. | 13 |
| Figure 2.3: Expressivité des langues. | 14 |
| Figure 2.4: Une ontologie sous forme de graphe. | 15 |
| Figure 2.5: Hiérarchie des classes d'entités dans l'ontologie MASON | 18 |
| Figure 2.6: Hiérarchie des classes d'opérations dans l'ontologie MASON | 19 |
| Figure 2.7: Classes principales de MSDL | 20 |
| Figure 2.8: Fragment de la hiérarchie des classes BFO aligné avec les classes portant les étiquettes des vingt-cinq premiers termes fournis par l'IOF | 21 |
| Figure 2.9: Classes et propriétés principales de l'ontologie RAMI4.0 . | 21 |
| Figure 3.1: L'architecture générale de l'AML . | 24 |
| Figure 3.2: Le modèle CAEX | 25 |
| Figure 3.3: L'éditeur AutomationML. | 26 |
| Figure 3.4: Description Topologique de l'Architecture de l'AutomationML. | 27 |
| Figure 3.5: Relation parent-enfant dans une bibliothèque de classes. | 27 |
| Figure 3.6: Exemple de relation d'héritage entre classes. | 28 |
| Figure 3.7: Exemple de relation de classe-instance. | 28 |
| Figure 3.8: Relation entre deux instances. | 28 |
| Figure 3.9: Les relations entre les objets CAEX. | 29 |
| Figure 3.10: Types de descriptions logiques dans AutomationML. | 31 |
| Figure 3.11: Exemple d'un diagramme de Gantt. | 32 |
| Figure 3.12: Exemple de graphique PERT. | 33 |
| Figure 3.13: Exemple de diagramme d'impulsion avec conventions de dénomination. | 34 |

| | |
|-------------------------------------------------------------------------------------------------------------------------|----|
| Figure 3.14: exemple de diagramme d'état | 34 |
| Figure 3.15: Exemple de SFC dans la CEI 61131-3. | 35 |
| Figure 4.1: Architecture générale. | 50 |
| Figure 4.2: hiérarchie AML des concepts et des relations | 52 |
| Figure 4.3: Hiérarchie des classes. | 56 |
| Figure 4.4: Définition des propriétés. | 56 |
| Figure 4.4: Définition des instances | 57 |
| Figure 4.6: Editeur de dialogue multi-ligne de SWRL | 61 |
| Figure 4. 7: les règles SWRL de l'ontologie. | 64 |
| Figure 4.8: La définition de la bibliothèque my RoleClassLib avec la description XML. | 65 |
| Figure 4.9: Un exemple d'unité de système « Réservoir d'eau » et « Pompe d'eau » avec une représentation XML. | 65 |
| Figure 4.10: Un exemple d'unité de système « Economiseur » et « les ballons d'eau » avec une représentation XML. | 66 |
| Figure 4.11: Présentation de la hiérarchie de la chaudière en XML (exemple 1). | 67 |
| Figure 4.12: Présentation de la hiérarchie de la chaudière en XML (exemple 2). | 68 |
| Figure 4.13: Présentation de la hiérarchie de la chaudière en XML (exemple 3). | 69 |
| Figure 4.14: Diagramme de GANTT de préventilation. | 71 |
| Figure 4.15: Diagramme de GANTT d'Allumage du premier brûleur. | 72 |
| Figure 4.16: Le mapping de GANTT vers le format IML. | 76 |
| Figure 4.17: Le mapping de GANTT vers le format IML. | 77 |
| Figure 5.1: Présentation de la fenêtre Protégé. | 82 |
| Figure 5.2: Présentation de la fenêtre JCreator. | 82 |
| Figure 5.3: Présentation de la fenêtre AutomationML | 83 |
| Figure 5.4: Présentation de l'éditeur SWRL | 84 |
| Figure 5.5: Interface de l'application. | 85 |
| Figure 5.6: Interface Principale de la Modélisation Statique | 86 |
| Figure 5.7 : Interface de Création d'un diagramme de GANTT. | 86 |
| Figure 5.8: Interface de Conversation l'IML vers le format XML de PLCopen. | 87 |
| Figure 5.9: Interface de Conversation le tableau GANTT au format IML. | 87 |

Listes des tableaux

| | |
|-------------------------------------------------------------------------|----|
| Tableau 3.1 : Normes prises en charge par AutomationML _____ | 24 |
| Tableau 4.2 : types d'instruments et leurs rôles _____ | 41 |
| Tableau4.3 : Dictionnaire des concepts. _____ | 53 |
| Tableau 4.4 : Table des relations binaires. _____ | 57 |
| Tableau 4.5 :Table des instances. _____ | 59 |
| Tableau 4.6 : Les actions de l'allumage du premier brûleur _____ | 71 |

Introduction générale

1 Contexte de la recherche

Avec la croissance exponentielle des systèmes industriels et la nécessité de disposer de systèmes informatiques robustes, le processus de développement des produits logiciels s'est avéré être une tâche extrêmement ardue, sa complexité ne faisant que s'accroître. Ainsi, la connectivité des logiciels, des équipements, des données massives à traiter et la cybersécurité sont des aspects cruciaux pour instaurer de l'intelligence au sein d'un système de fabrication, favorisant ainsi une adaptabilité accrue dans la production et une allocation plus efficace des ressources grâce aux technologies de l'information et de l'automatisation. Ceci caractérise la quatrième révolution industrielle, également connue sous le nom de l'« industrie 4.0 ». L'usine 4.0 se distingue par l'incorporation des technologies numériques au sein des procédés de production. Cela permet d'accroître la compétitivité et de bénéficier d'un avantage concurrentiel, d'améliorer la flexibilité de la production et de permettre au client de concevoir un produit sur mesure répondant à ses besoins spécifiques, tout en lui offrant la possibilité de suivre l'intégralité du processus de fabrication.

En l'absence de l'internet des objets, il serait impossible d'envisager l'avènement d'une quatrième révolution industrielle ou de l'industrie 4.0. La transition en cours de l'Internet traditionnel vers l'Internet des objets entraîne de profonds bouleversements dans les modèles de communication sous-jacents. L'accroissement de la complexité des installations industrielles et la diversité des parties prenantes impliquées dans le processus de construction des usines compliquent de manière croissante la communication et la validation des modèles d'ingénierie des usines.

L'AutomationML (Automation Markup Language, abrégé AML) est une norme de l'I4.0 conçue dans le but de proposer un format structuré d'échange de données basé sur XML, qui intègre diverses phases du processus d'ingénierie. Il repose sur le paradigme orienté objet qui a été employé pour la modélisation et la conservation des données provenant du secteur industriel. L'outil CAEX est employé dans le langage AML afin de proposer une représentation hiérarchique de divers objets. L'élément principal de ce dernier est désigné sous le nom d'InstanceHierarchy, lequel contient plusieurs Éléments Internes (IE) pouvant inclure des attributs décrivant les caractéristiques de l'objet.

Les diagrammes de Gantt sont des outils visuels couramment employés pour illustrer la séquence et la durée des activités dans les modèles d'événements discrets. Ces activités sont symbolisées par des barres, offrant ainsi une vue abstraite et structurée de leur ordre et de leur calendrier. Ils sont employés durant les phases initiales de l'ingénierie des installations afin de modéliser la séquence des opérations de production. Les éléments clés enregistrés dans les diagrammes de Gantt comprennent les horaires de début et de fin des tâches ainsi que les liens de précedence et de succession entre ces tâches.

Le Web sémantique facilite la compréhension par les machines de la sémantique et de la signification des données présentes sur le Web, favorisant ainsi la collaboration entre les ordinateurs et les individus. L'industrie 4.0 élargit ce concept à la communication machine-machine (M2M) au

sein des processus industriels. Le concept de "cyber-systèmes physiques" (CPS) a également été introduit. Un CPS surveille les processus physiques, crée des copies virtuelles du monde physique et prend des décisions décentralisées. Il est caractérisé à l'aide de l'outil de modélisation des données d'ingénierie AutomationML (AML), qui facilite la création d'un document CAEX (Computer Aided Engineering Exchange), permettant ainsi l'importation et l'analyse du document PLCopen XML (Programmable Logic Controller open XML).

La sémantique d'exécution de la programmation logique des contrôleurs (PLC), les unités d'organisation de la programmation et les langages de programmations sont définis dans la norme CEI 61131-3. La norme CEI 61131-3 fournit un format de spécification de machine d'état graphique appelé diagramme de fonction séquentiel (SFC) qui est fréquemment utilisé pour le contrôle de programmation des appareils et des machines.

2 Problématique

Le premier enjeu abordé concerne les systèmes industriels, caractérisés par leur complexité en termes de volume et de diversité des composants à entretenir. Nous abordons aussi une problématique majeure dans le domaine de l'Internet des objets, à savoir l'hétérogénéité sémantique. L'objectif est de concevoir des modèles capables de générer des structures fiables et flexibles afin de répondre aux besoins futurs d'un système interopérable.

Dans le contexte de l'industrie 4.0, les Cyber-Physical Systems (CPS) représentent un environnement de production intelligent interconnecté grâce à l'Internet des objets (IoT). Les systèmes d'intelligence artificielle jouent un rôle crucial dans l'exécution de tâches complexes tout en interagissant avec les opérateurs humains, les clients et divers partenaires. Dans ce cadre, la diversité de l'IoT constitue un défi majeur. Ainsi, les ontologies se révèlent être particulièrement avantageuses dans le cadre de l'approche sémantique de l'Internet des objets. Ces vocabulaires normalisés permettent de décrire de manière cohérente les objets, les capteurs, les actionneurs et leurs attributs spécifiques, réduisant ainsi de manière significative l'effet de l'hétérogénéité. Afin de garantir un développement fluide, il est essentiel de traiter de manière efficace la normalisation de la représentation des connaissances, afin de prévenir d'éventuels problèmes tout au long du processus de production. Les ontologies de haut niveau, issues de la normalisation des connaissances, constituent le fondement pour élaborer des ontologies plus spécifiques à un domaine donné. En général, les ontologies utilisées sont représentées en OWL, mais vue les insuffisances de ce langage, l'ontologie est étendue par des règles SWRL qui permettent d'augmenter l'expressivité d'OWL.

3 Contribution

La numérisation des processus organisationnels et de production présente de multiples avantages. En s'inscrivant dans le cadre de la 4ème révolution industrielle, les petites et moyennes entreprises bénéficient des nouvelles opportunités de croissance qui en découlent. L'ouverture aux marchés national et international implique que l'industrie se digitalise, ce qui se reflète dans les stratégies marketing favorisant l'utilisation d'Internet. Cela offre aux petites et moyennes entreprises la possibilité d'explorer de nouveaux débouchés commerciaux et d'accéder à de nouveaux marchés nationaux et internationaux. À l'heure actuelle, SONATRACH a rassemblé une quantité substantielle de données et de connaissances, réparties à la fois sur les plateformes numériques et dans les centres de documentation. Cette prise de conscience a initié la transition numérique de SONATRACH, visant

à moderniser et rationaliser ses activités. De même, les installations industrielles évoluent constamment vers l'Industrie 4.0, un concept mettant en avant l'interconnexion et l'intelligence.

Le nouveau format de données AutomationML aide à remédier aux problèmes d'hétérogénéité sémantique en offrant un format de fichier qui peut intégrer divers types de données liées à la planification et à l'ingénierie d'une installation de production. Il repose sur XML et fait appel à des normes largement reconnues, ouvertes et gratuites dans les domaines de l'ingénierie concernés. Par exemple, CAEX définit la structure et la hiérarchie d'une installation, tandis que PLCopen XML est utilisé pour décrire la partie dynamique de cette installation.

L'initiative de normalisation ontologique pour l'Industrie 4.0 repose sur l'utilisation des ontologies standards et englobe un ensemble d'ontologies spécifiquement conçues pour les domaines de la robotique, de l'automatisation et des Systèmes Cyber-Physiques (CPS). Il est possible d'élargir ces ontologies afin d'intégrer des concepts spécifiques à la chaudière industrielle.

L'enrichissement de l'ontologie par des règles SWRL consistant à injecter un système à base de règles dans la base de connaissance présente plusieurs avantages à savoir :

- Les règles assurent une plus haute expressivité relationnelle.
- Elles permettent la définition des relations sémantiques plus riches.
- Elles permettent aussi d'exprimer des prédicats d'arité plus grande que celle permise par la logique de description sur laquelle se base OWL.

Nous présentons une méthode de création d'ontologie semi-automatique basée sur les documents CAEX et PLCOPENXML dans le contexte de la surveillance industrielle. En se basant sur ces objectifs, nous sommes en mesure d'identifier les principaux axes à développer : Présentation de l'architecture globale de notre système.

- Modélisation statique et dynamique de la chaudière industrielle à l'aide de l'outil AutomationML.
- Élaboration d'une ontologie de domaine nommée "Boiler.OWL" basée sur la hiérarchie AML des concepts et des relations.
- La population de l'ontologie est réalisée à travers des instances provenant des documents CAEX et du fichier PLCopen XML (Mapping CAEX-OWL et PLCopen-SWRL).
- La représentation sémantique des activités des diagrammes de Gantt à l'aide du langage SWRL.
- Mise en place d'interfaces conviviales pour la navigation à travers un fichier XML. Ces interfaces facilitent la conversion des documents CAEX et PLCopen XML.

4 Organisation du document

Notre mémoire de fin d'études se compose en cinq chapitres :

Chapitre 1 : L'industrie 4.0

Ce chapitre se concentrera sur l'analyse des défis liés à l'industrie 4.0 et à l'Internet des objets. Nous exposons une définition de l'usine 4.0, son mode de fonctionnement ainsi que les bénéfices inhérents à ce concept.

Chapitre 2 : Ontologies Supérieures dans l'Industrie 4.0

Ce chapitre offre une brève présentation des technologies du Web sémantique telles que le langage OWL et le langage SWRL. Il met également l'accent sur l'exploration des ontologies de haut niveau les plus fréquemment employées dans le domaine industriel.

Chapitre 3 : AutomationML et La chaudière industrielle

Dans la première partie de ce chapitre nous présenterons les principes essentiels de l'AML, mettant en avant l'utilisation de CAEX pour la représentation de la structure des composants du système, ainsi que la description logique de ces éléments à l'aide de PLCopen XML. Puis nous traitons dans la deuxième partie le processus de génération de vapeur par les chaudières industrielles ainsi que les différents composants. Ensuite nous présentons leur mode de fonctionnement.

Chapitre 4 : Proposition du système

Dans ce chapitre nous proposons l'architecture de notre système ainsi qu'une description détaillée des différentes phases de développement.

Chapitre 5 : Implémentation

Le dernier chapitre est consacré à la présentation de différents environnements de développement et les langages de programmation utilisés (protégé OWL, JCreator, GANT.....) pour la réalisation de notre projet.

Chapitre 01

L'Industrie 4.0

1 Introduction

Grâce à l'Internet, le domaine des ordinateurs et des communications a connu une révolution sans précédent. En quelques années, elle est devenue le principal moyen de diffusion de l'information. Dans de nombreux secteurs, il s'est imposé comme une infrastructure indispensable pour les personnes, les entreprises et les institutions (Baghdadli Chahrazed et al., 2015).

La connectivité à large bande est de plus en plus abordable et répandue. Les dispositifs gagnent en puissance et en taille grâce à une diversité de capteurs intégrés. La multiplication des dispositifs connectés entraîne un nouveau paradigme : l'Internet des objets. L'objectif de ce concept est d'associer le monde virtuel d'Internet avec le monde physique des objets intelligents communicants, comme les capteurs et les actionneurs, pour garantir une meilleure accessibilité et utilisation des ressources du monde réel. Dans un tel paradigme, on prévoit une expansion considérable de l'Internet d'aujourd'hui, avec de nouvelles applications telles que la surveillance, la sécurité, la santé, les maisons et villes intelligentes, les systèmes de logistique et de transport intelligents. En réalité, l'IoT a incité les usines et les gouvernements à s'engager (N.Jazdi, 2014).

La notion d'Industrie 4.0 désigne une approche innovante pour organiser les moyens de production. Le but est donc de créer des usines intelligentes qui offrent une meilleure transparence dans la production et une allocation des ressources plus efficace. Les technologies de l'Internet des objets et les systèmes cyber-physiques sont les bases de l'industrie 4.0. Au cours de ce chapitre nous allons présenter la définition de I4.0, leur objectif, principe de conception puis en discute la technologie de l'IOT.

2 Définition et Objectifs

L'Industrie 4.0 (I4.0) est un terme inventé pour représenter la quatrième révolution industrielle basée sur les dernières avancées technologiques. Elle va bien au-delà des CPS, impliquant des systèmes avancés de communication de données (Wollschlaeger et al., 2017), l'intelligence embarquée et la normalisation de la sémantique des données (Ragavan et al., n.d.).

L'objectif principal d'I4.0 est de rendre la fabrication plus efficace en optimisant et en personnalisant les processus de production. L'I4.0 est basé sur le concept d'usines intelligentes, où les machines sont intégrées aux humains par le biais de CPS. Les usines intelligentes sont capables d'échanger automatiquement des informations entre les ressources de fabrication telles que les capteurs, les actionneurs, les machines, les robots et les convoyeurs. Elles peuvent également maintenir intelligemment le processus de production et être autonomes. Pour atteindre l'objectif de création d'usines intelligentes trois types d'intégration doivent être assurés : l'intégration verticale, au

sein d'une usine ou d'un atelier de production ; l'intégration horizontale, par le biais de réseaux de valeur ; et l'intégration technique de bout en bout sur l'ensemble de la chaîne de valeur pour soutenir la personnalisation des produits.

a) Intégration verticale dans l'I4.0 : Elle est liée à l'intégration de systèmes informatiques et comprend l'intégration de systèmes à différents niveaux hiérarchiques de fabrication en une solution globale. Cette intégration s'effectue depuis le niveau de l'atelier où se trouvent les dispositifs tels que les capteurs, les actionneurs et les CPS jusqu'au niveau de la planification de l'entreprise avec les systèmes de ressources d'entreprise (ERP).

b) Intégration horizontale dans l'I4.0 : l'intégration horizontale implique une collaboration entre les partenaires, les fournisseurs, les clients, mais aussi d'autres membres de l'écosystème, de la logistique à l'innovation, les flux ainsi que les parties prenantes. Les usines intelligentes atteignent le monde entier en utilisant des chaînes de production mondiales et des réseaux de données dans leurs opérations. Il est donc nécessaire de procéder à l'intégration des données entre tous ces participants. L'intégration horizontale doit garantir que l'usine est capable d'interagir en tant qu'usine intelligente sur un marché mondial.

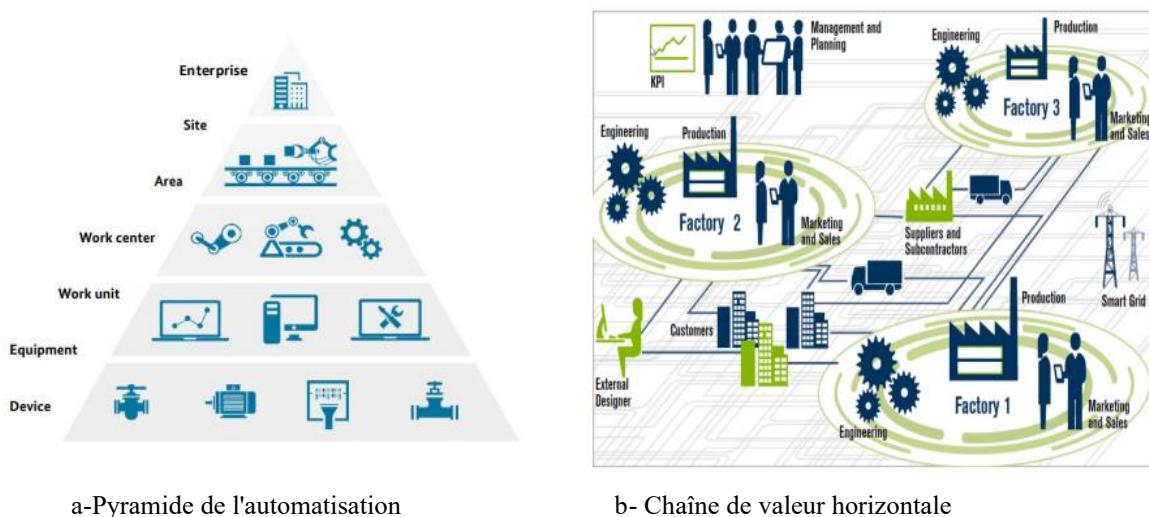


Figure 1.1: Paramètres horizontaux et verticaux des scénarios I4.0(Grangel-González, 2018)

c) Intégration de bout en bout : le développement du cycle de vie du produit comprend de nombreuses activités d'ingénierie pour créer un CPS, par exemple la conception, la production, l'utilisation et l'arrêt. Au cours de l'ingénierie de systèmes complexes tels que les CPS, les parties prenantes, qui appartiennent généralement à des disciplines d'ingénierie différentes, doivent collaborer efficacement. L'objectif du processus d'ingénierie d'un CPS est de fournir un produit final de haute qualité, par exemple la conception d'une usine de production complète, et de respecter des délais stricts. La présence de diverses disciplines d'ingénierie conduit à des environnements très complexes et à forte intensité logicielle, qui se caractérisent par :

- Une multitude d'outils d'ingénierie qui ne sont pas conçus pour coopérer les uns avec les autres ;
- Une variété de représentations et de formats d'échange de données spécifiques au domaine de l'ingénierie ; et

- Des différences dans les flux de travail adoptés par les disciplines impliquées (Grangel-González, 2018).

3 Principes de conception de l'industrie 4.0

Hermann (Hermann et al., 2016) définit quatre principes de conception guidant les praticiens et les scientifiques sur la marche à suivre pour transiger vers l'industrie 4.0. Il s'agit de l'interconnexion, la transparence de l'information, les décisions décentralisées et l'assistance technique. Ces principes sont illustrés dans la figure 2 et détaillés ci-dessous :

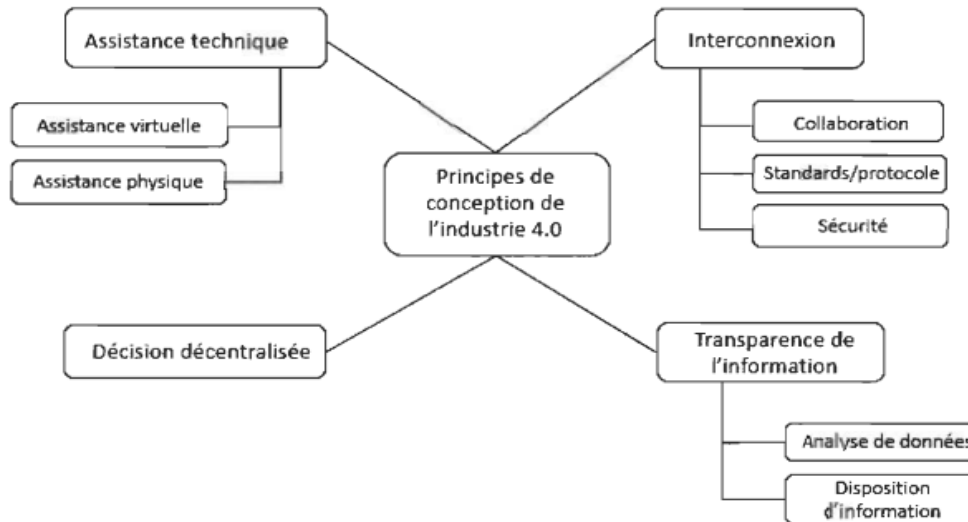


Figure 1.2: Principes de conception de I4.0 (Hermann et al., 2016).

- **Interconnexion** : Par l'intermédiaire de l'internet des objets, les machines, appareils, capteurs et personnes forment un système connecté, leurs permettant de collaborer et partager entre eux des informations (Hermann et al., 2016) Ce partage constitue la base d'une collaboration commune visant à atteindre des objectifs communs (Giusto et al., 2010). Ainsi, Il existe trois types de collaboration au sein de l'IoT : la collaboration homme-homme, la collaboration personne-machine et la collaboration machine-machine. La connexion hommes, machines, capteurs, et processus par des protocoles de communications est d'une grande importance. Ces protocoles permettent de réaliser une intégration transparente des processus de production. Cette intégration aide les usines intelligentes à s'adapter, avec souplesse, aux demandes fluctuantes et personnalisées du marché (Zuehlke, n.d.).
- **Transparence de l'information** : Selon (Hermann et al., 2016) ,la fusion du monde physique et virtuel permet une nouvelle forme de transparence de l'information, qui en reliant les données des capteurs aux modèles d'usine numérisés, une copie virtuelle du monde physique est créée. En outre, il précise que les informations contextuelles provenant du monde virtuel et physique sont indispensables pour que les systèmes accomplissent leurs tâches et prennent des décisions appropriées. Par exemple, les documents électroniques, les dessins et les modèles de simulation sont des informations provenant du monde virtuel, tandis que la position ou les conditions d'un outil sont des informations issues du monde physique. Ainsi pour analyser ce dernier, les données brutes des capteurs doivent être agrégées à des informations de plus grande valeur et interprétées, et les résultats de l'analyse des données doivent être intégrés, en temps

réel, dans des systèmes d'assistance accessibles à tous les participants du processus afin de créer de la transparence (Gorecky et al., 2014).

- **Décisions décentralisées** : Les décisions décentralisées reposent sur l'interconnexion d'objets et de personnes ainsi que sur la transparence des informations provenant de l'intérieur et de l'extérieur d'une unité de production. La combinaison de décideurs interconnectés et décentralisés permet d'utiliser simultanément des informations locales et mondiales pour une meilleure prise de décision et une productivité globale accrue (Malone, 1999). Les participants à l'IoT accomplissent leurs tâches de la manière la plus autonome possible. Ce n'est qu'en cas d'exceptions, d'interférences ou d'objectifs conflictuels que les tâches sont déléguées à un niveau supérieur (Kagermann, 2014). D'un point de vue technique, les décisions décentralisées sont activées par CPS. Leurs ordinateurs, capteurs et acteurs intégrés permettent de surveiller et de contrôler le monde physique de manière autonome.
- **Assistance technique** Dans les usines intelligentes de l'industrie 4.0, le rôle principal des humains passe d'un opérateur de machines à un décideur stratégique et à un résolveur de problèmes flexible. En raison de la complexité croissante de la production, où les systèmes cyber physique forment des réseaux complexes et prennent des décisions décentralisées, les humains doivent être soutenus par des systèmes d'assistance. Ces systèmes doivent regrouper et visualiser les informations de manière compréhensible pour garantir que les humains peuvent prendre des décisions éclairées et résoudre des problèmes urgents dans un bref délai (Gorecky et al., 2014).

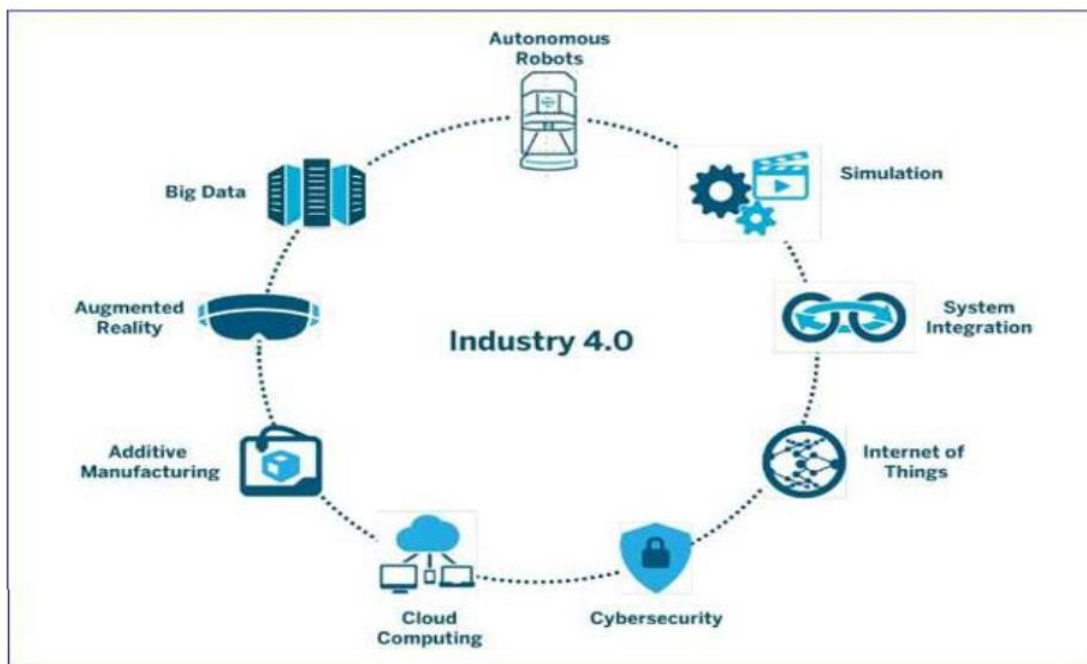


Figure 1.3: Périmètre de l'industrie 4.0(X. Lu et al., 2014)

4 Internet des objets

C'est un nouveau paradigme qui gagne du terrain de jour en jour dans le monde des télécommunications et des réseaux informatiques. L'apparition de ce concept est due à la variété des équipements et des objets qu'on utilise dans notre vie quotidienne par exemple les ordinateurs, les capteurs, les actionneurs, les smartphones, les véhicules connectés et les smart homes (Hammi, 2015).

Selon l'Union internationale des télécommunications, l'Internet des objets est une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ». En réalité, la définition de ce qu'est l'Internet des objets n'est pas figée. Elle combine trois aspects, ou visions technologiques, qui sont : la vision des objets, de l'internet et de la sémantique (Atzori, 2010).

Les visions sémantiques se rapportent à la compréhension des données, normes et technologies à utiliser, pour atteindre l'interopérabilité. Lors de l'interaction des objets, les objets sont au cœur de l'IoT et doivent être identifiés de manière unique. L'internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Chaque objet doit être autonome, capable d'interagir et coopérer avec d'autres objets afin d'atteindre des objectifs communs. Ce réseau crée une passerelle entre le monde physique et le monde virtuel. Les visions d'Internet, consiste à l'identification numérique directe et normalisée (adresse IP, protocoles smtp, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wi-Fi... ce qui signifie que les objets ont leur propre adresse IP, ou peuvent facilement communiquer avec Internet.

5 Systèmes cyber-physiques

Les systèmes cyber-physiques (CPS) constituent l'élément central du paradigme de l'Industrie 4.0. Les Cyber-Physical Systems (CPS) sont caractérisés par une diversité d'entités physiques telles que des capteurs et des modules de contrôle, qui interagissent au moyen de réseaux de communication, en collaboration avec des applications hébergées sur des infrastructures cloud et/ou des utilisateurs humains, dans le but de réaliser un objectif commun. Les systèmes de commande programmable intègrent un système informatique comprenant des composants mécaniques et électroniques interconnectés à des réseaux en ligne, facilitant ainsi la communication entre les machines de manière similaire aux réseaux sociaux. De manière générale, les Systèmes de Contrôle en Réseau (CPS) se composent de trois éléments principaux : un noyau de calcul cybernétique, un objet contrôlé de nature physique, et un réseau permettant la communication entre le noyau de calcul et l'objet physique. (Grangel-González, 2018)

6 L'impact positif de l'industrie 4.0 sur la production

Les dirigeants voient dans la révolution numérique un moyen de satisfaire toujours mieux leurs clients grâce à de nombreuses avancées en matière de gestion de production. La digitalisation de l'entreprise offre en effet de multiples opportunités pour (AMOKRANE Ahlem et al., 2019):

- Développer la personnalisation de la production industrielle.
- Améliorer la capacité d'adaptation.
- Permettre un suivi plus précis de l'état de la production.
- Offrir des gains de productivité.
- Avoir une plus grande réactivité de la production aux demandes des clients.

Les dirigeants voient dans l'industrie 4.0 une formidable opportunité pour développer des gains de compétences dans les équipes.

7 Conclusion

Nous avons discuté dans ce chapitre le concept de base « Industrie 4.0 » qui correspond à une nouvelle façon d'organiser les moyens de production. L'objectif est la mise en place d'usines dites intelligentes, capables d'une plus grande adaptabilité dans la production et d'une allocation plus efficace des ressources, pour répondre plus rapidement au marché, d'une façon plus personnalisée et à moindre coût. Dans le prochain chapitre nous allons présenter la notion d'ontologie.

Chapitre 02

Ontologies Supérieures dans l'Industrie 4.0

1 Introduction

Depuis sa création, le web a connu un succès gigantesque et il est en train de devenir peu à peu le premier outil pour la production, la publication, la diffusion et le partage de l'information. Cependant la plupart des informations se trouvant sur Internet sont effectivement lisibles par les hommes mais elles sont difficilement interprétables par les machines. La croissance accrue du nombre de publications et la liberté totale d'y accéder ont révélé plusieurs limites et inconvénients.

Le Web Sémantique est une extension ou amélioration du Web actuel, dans lequel la signification des données est prise en compte. L'expression de Web Sémantique, énoncée par Tim Berners-Lee fait d'abord référence à la vision du Web du futur comme un vaste espace d'échange de ressources supportant la collaboration entre humains et machines en vue d'exploiter plus efficacement de grands volumes d'informations et de services variés disponibles sur le Web.

Les ontologies consistent en une conceptualisation formelle de la représentation des connaissances et fournissent les définitions des concepts et des relations qui capturent les connaissances d'un domaine de manière interopérable. Ils sont utilisés dans plusieurs domaines comme l'ingénierie des connaissances, les systèmes experts et en informatique. Parmi les domaines où l'ontologie joue un rôle fondamental est le web sémantique, qui est fortement distribué (réparti) et qui constitue une prolongation du web actuel où il a ajouté des ontologies pour préciser le sens et la sémantique des ressources du web afin que les machines puissent les comprendre.

Les ontologies de premier niveau servent de racine pour définir des ontologies plus spécifiques. Wang (X.H. Wang et al., 2004) définissent une ontologie supérieure comme "Une ontologie de haut niveau qui capture les caractéristiques générales des entités contextuelles de base". L'objectif d'une telle ontologie supérieure est de permettre aux ontologies spécifiques de s'intégrer de manière fluide dans la même architecture cognitive commune, permettant ainsi le partage efficace du modèle de données entre des environnements hétérogènes. Dans ce chapitre, nous allons commencer par la définition du web sémantique puis nous présentons le concept d'ontologie et les différentes définitions proposées dans la littérature.

2 Web Sémantique

L'expression Web sémantique est une extension du web qui facilite l'automatisation du traitement des connaissances disponibles. Dans le web sémantique, les utilisateurs sont déchargés d'une bonne partie de leurs tâches de recherche, de construction et de combinaison des résultats,

grâce aux capacités accrues des machines à accéder aux contenus des ressources et y effectuer des raisonnements sur ceux-ci. Le Web actuel est essentiellement syntaxique, dans le sens où la structure des documents (ou ressources au sens large) est bien définie, mais que son contenu reste inaccessible aux traitements machines. Seuls les humains peuvent interpréter leurs contenus. Les ressources du Web 3 seront plus aisément accessibles aussi bien par l'homme que par la machine, grâce à la représentation sémantique de leurs contenus par les ontologies. Concrètement, le Web sémantique est donc une infrastructure qui permet l'utilisation de connaissances formalisées (ontologies) en plus du contenu informel actuel du Web.

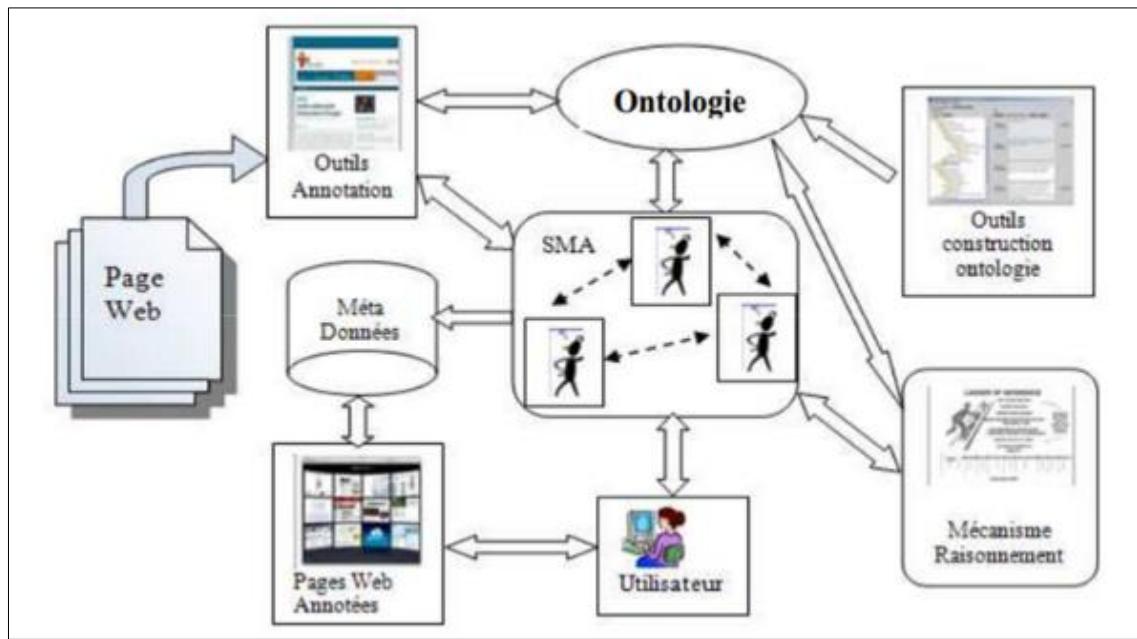


Figure 2.1: Vision du web sémantique (Gasmi Mohamed, 2017)

L'architecture du Web sémantique repose sur une hiérarchie des langages d'assertion et de description d'ontologies ainsi que sur un ensemble de services.

Les langages permettent :

- L'accès aux ressources au moyen de leurs références sémantiques ;
- La gestion de l'évolution des ontologies et
- L'utilisation des moteurs d'inférences capables d'effectuer des raisonnements complexes ainsi que des services pour la vérification de la validité sémantique de ces raisonnements.

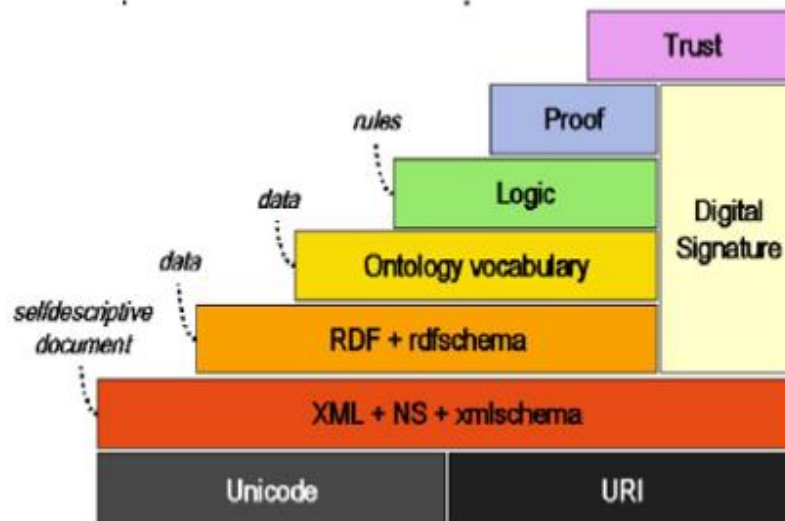


Figure 2.2: Architecture du web sémantique (Tim Berners-Lee et al., 2001).

3 Notion d'Ontologie

Historiquement, le terme Ontologie a été défini en philosophie comme une branche de la métaphysique qui s'applique à l'être en tant qu'être, indépendamment de ses déterminations particulières. En effet, ce terme grec est composé des mots « ontos » et « logos » qui veulent dire respectivement l'essence de l'être. Au début des années 90, des chercheurs en intelligence artificielle se sont intéressés à cette notion pour la formalisation des connaissances. Du point de vue de l'ingénierie des connaissances, différentes définitions de l'ontologie ont été données. La plus référencée et aussi la plus synthétique est sans doute celle formulée par GRUBER (T. R. Gruber, 1993) « Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée ».

3.1 Composants d'une ontologie

Une ontologie est composée de : concepts, propriétés, relations, axiomes et des instances (M. Uschold & M. King, 1995)

- **Le concept** (classe) : Il peut représenter un ensemble d'objet, une idée, un principe ou une notion abstraite. Il est caractérisé par une extension et une intension.
- **Les relations** : sont des relations entre les concepts d'un domaine. Elles sont définies comme tout sous- ensemble d'un produit cartésien de n ensembles, ces dernières permettent de structurer hiérarchiquement les concepts d'une ontologie
Par exemple : Identité, synonymie(symétrique), sort de(unidirectionnelle), l'équivalence, homonymie, antonymie.
- **Les axiomes** : constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie.
- **Les instances** (Individuals) : Elles constituent la définition extensionnelle d'une ontologie.

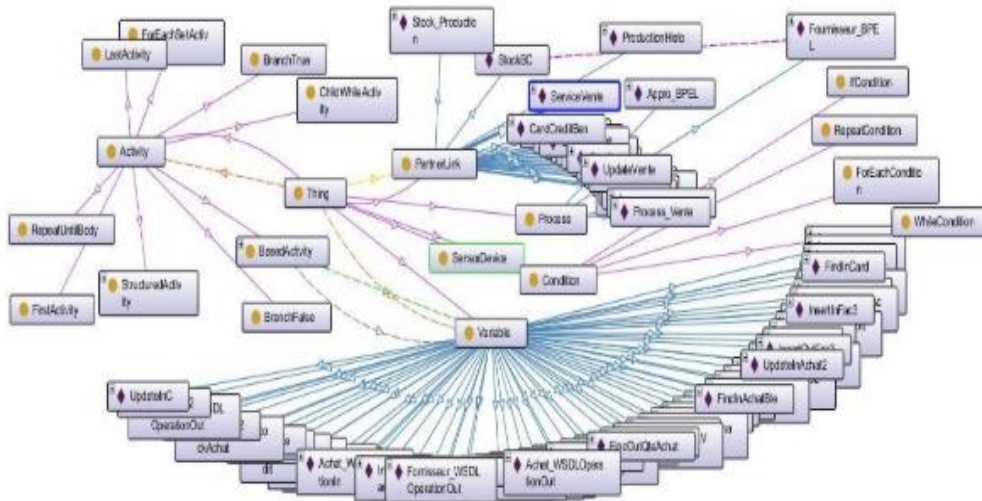


Figure 2.3: Une ontologie sous forme de graphe (L. Amir, 2017)

3.2 Rôle d'ontologies

Une ontologie permet de (N. Zouggar et al., 2006) :

- Acquérir et représenter les connaissances.
- Rechercher et faire l'extraction des connaissances c'est-à-dire inférer la connaissance qui est pertinente face à la requête de l'utilisateur.
- Partager et intégrer les connaissances.
- Gérer des connaissances : simplification du dialogue homme-machine.

3.3 Langages d'ontologies

Pour décrire une ontologie, plusieurs langages ont proposés dans la littérature tels que : XML, RDF, RDFS et OWL. XML fournit une syntaxe à des documents structurés, mais il n'impose pas les contraintes sémantiques. RDF est un modèle de données qui peut être représenté dans une syntaxe XML. RDF Schéma (RDFS) est un langage de définition pour la description de propriétés et de classes représentées par des ressources RDF. Il définit des graphes de triplets RDF, avec une sémantique de généralisation / hiérarchisation de ces propriétés et de ces classes. OWL ajoute des vocabulaires pour la description des propriétés et des classes, des relations entre classes, des cardinalités, des caractéristiques de propriétés (symmetry), et des classes énumérées. OWL est développé comme une extension du vocabulaire de RDF.

SWRL (Semantic Web Rule Language) (I. Horrocks et al., 2004) est un langage de règles pour le web sémantique combinant le langage OWL-DL et le langage RuleML (Rule Markup Language (Unary/Binary Datalog)). SWRL est un langage qui enrichit la sémantique d'une ontologie définie en OWL. SWRL permet contrairement à OWL de manipuler des instances par des variables ($?x$, $?y$, $?z$). SWRL ne permet pas de créer des concepts ni des relations. Il permet simplement d'ajouter des relations suivant les valeurs des variables et la satisfaction de la règle. OWL permet de spécifier des propriétés algébriques pour les relations. Mais il lui manque surtout des possibilités pour encoder des connaissances plus générales, relatives en particulier à la composition des relations. Ainsi, SWRL constitue un pas vers le rapprochement entre les langages OWL et les règles logiques. SWRL intègre une syntaxe abstraite de haut niveau pour les règles de Horn dans le langage OWL.

OWL et SWRL sont les langages fondamentaux du Web sémantique (Bouaroudj Samiya & Boufaïda, 2009). OWL est un langage d'ontologie développé pour construire des ontologies fournissant un haut niveau d'expressivité pour le contenu du Web. SWRL est basée sur OWL, il garde la puissance d'OWL DL mais au prix de la décidabilité. La figure 2.2 montre que l'ajout des règles à l'ontologie fournit une couche supplémentaire d'expressivité. SWRL permet aux utilisateurs d'écrire des règles pour raisonner sur les individus et pour en déduire de nouvelles connaissances sur ces individus.

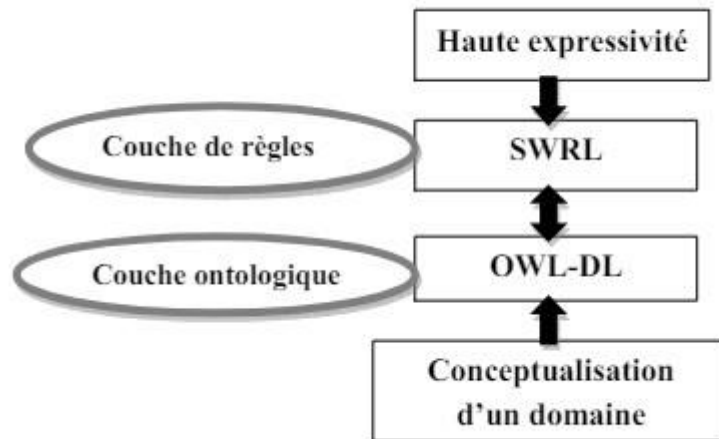


Figure 2.4: Augmentation du niveau d'expressivité avec SWRL (I. Horrocks et al., 2004)

3.4 Types d'ontologie

Les méthodes en Ingénierie des connaissances ont répertorié plusieurs types d'ontologie liés à l'ensemble des objets conceptualisés et manipulés au sein d'un SBC. Parmi ces types d'ontologies, nous citons :

3.3.1 Ontologie de domaine

Une ontologie de domaine est la représentation d'un domaine particulier de connaissances. Elle exprime des conceptualisations spécifiques à un domaine. Elle regroupe les concepts et les relations permettant de couvrir les connaissances d'un domaine. La plupart des ontologies disponibles sur le web sont des ontologies de domaines. A titre d'exemple d'ontologie de domaine médicale (MENELAS) qui regroupe les concepts de domaine de la médecine, l'ontologie du domaine du tourisme et celles de domaine de la bibliographie et l'anatomie humaine.

3.3.2 Ontologie de tâche

C'est une ontologie spécifique pour résoudre un problème bien défini comme les problèmes de planification et de diagnostic. Elle offre un ensemble de termes au moyen desquel nous pouvons décrire généralement comment résoudre un type de problèmes. Elle contient des noms, des verbes et des adjectifs pour la description de tâches.

3.3.3 Ontologies d'application

Ce sont les ontologies les plus spécifiques. Elles permettent de décrire des concepts dépendants à la fois d'un domaine et d'une tâche. Dans cette classification, la notion d'ontologie d'application définit le contexte d'une application qui décrit la sémantique des

informations et des services manipulés par une ou un ensemble d'applications sur un même domaine (G. Van Heijst et al., n.d.).

3.3.4 Ontologies supérieures

Selon Wang (X.H. Wang et al., 2004), une ontologie supérieure est définie comme une ontologie de haut niveau qui saisit les caractéristiques générales des entités contextuelles de base. Ce genre d'ontologies a pour objectif d'analyser les catégories des entités existant dans le monde, telles que les concepts abstraits de niveau supérieur incluant les entités, les événements, les processus, les actions, le temps, les relations et les propriétés.

Ameri et al. (Farhad Ameri & Debasish Dutta, s.d.) proposent une définition plus précise de l'ontologie supérieure comme étant une ontologie légère qui se restreint à des concepts méta, génériques, abstraits et philosophiques, ce qui la rend suffisamment générale pour englober divers objets dans le domaine considéré. Ainsi, en dépit de fournir une certaine normalisation, une ontologie de haut niveau doit être suffisamment flexible et adaptable pour répondre aux besoins d'environnements très diversifiés et changeants.

Afin d'être reconnue comme une ontologie de référence pour l'industrie 4.0, celle-ci doit :

- Couvrir le domaine de la fabrication, le domaine de l'industrie 4.0 ou être au moins utilisée dans des applications relevant de l'industrie 4.0 (domaine d'intérêt).
- Il ne doit pas être restreint à un domaine spécifique ou à une tâche particulière, mais plutôt être générique, présenter des caractéristiques générales et une architecture cognitive commune.
- Il est essentiel que le système facilite l'intégration et le partage des données à l'échelle intra-entreprise et inter-entreprises.

Lemaignan (S. Lemaignan, A. Siadat, & A. Semenenko, 2006) proposent une définition plus centrée sur l'objectif, selon laquelle une ontologie supérieure vise à faciliter l'intégration fluide des ontologies spécifiques au sein d'une architecture cognitive commune, favorisant ainsi un partage efficace des modèles de données entre des environnements hétérogènes tels que différents départements au sein d'une même entreprise ou différentes entreprises.

3.3.5 Exemples d'ontologies supérieures

Il y a trente ontologies qui pourraient être qualifiées d'ontologies de haut niveau. Étant donné que la plupart des ontologies dédiées à l'industrie 4.0 font usage des langages OWL et RDF. Les ontologies utilisant des langages différents ont été exclues. En se basant sur les critères de disponibilité publique et d'utilisation d'OWL, quatre initiatives d'ontologies de qualité supérieure ont été identifiées :

3.3.5.1 Ontologie AML

Cette section est consacrée à la description de l'ontologie utilisée pour la représentation de la norme AML. La norme AML représente une solution de partage de données spécifiquement conçue pour le domaine de l'ingénierie de l'automatisation. La norme émergente vise à faciliter l'interopérabilité des données entre les différents outils d'ingénierie. Une modélisation de la norme AML à partir d'une ontologie offre divers avantages (Grangel-González, 2018a) :

- La flexibilité de l'affinement des schémas et l'intégration de données hétérogènes ;
- L'exploitation de la pile technologique sémantique pour optimiser les processus d'ingénierie dans le domaine des systèmes cyber-physiques ;

- L'interconnexion avec d'autres normes industrielles déjà dotées de représentations sémantiques, telles que le catalogue eCl@ss (EClasse.V., EClassStandardizedMaterial and Service Classification, 2016) ou l'ontologie GoodRelations (M. Hepp, 2008), par le biais de la réutilisation d'ontologies ou de mécanismes de données liées ;
- La liaison avec les représentations d'autres domaines, comme l'eCore dans le cadre de l'ingénierie basée sur les modèles.

Diverses méthodes ont été développées pour proposer une représentation sémantique. Cependant, ces initiatives souffrent des limitations suivantes

- Ces ontologies ne sont pas exhaustives par rapport à la norme, mais sont conçues pour des cas d'utilisation particuliers ;
- De plus, elles ne respectent pas les meilleures pratiques en matière de conception d'ontologies et
- Ne sont pas accessibles publiquement pour être consultées, étendues ou améliorées dans leur conception actuelle.

L'ontologie AML (AMLO) présentée dans cette section représente une avancée significative en résolvant les conflits mentionnés précédemment de la manière suivante (O. Kovalenko et al., 2018):

Tout d'abord, AMLO est une ontologie OWL qui englobe l'intégralité de la norme de communication de données émergente utilisée dans le domaine de l'ingénierie des systèmes cyber-physiques. En second lieu, AMLO repose sur deux ontologies distinctes qui ont été développées de manière indépendante. Son objectif est de se conformer à la norme AML dans le cadre d'applications spécifiques. La première a été élaborée en utilisant une approche de modélisation descendante. Le but de cette ontologie était de saisir les concepts fondamentaux de l'AML, à savoir les classes et leurs relations, afin de faciliter la vérification de la cohérence entre les divers fichiers AML. La seconde ontologie a été élaborée dans le cadre de l'approche Alligator. Cette ontologie a été développée en suivant une approche ascendante de modélisation, ce qui se traduit par une structure de propriétés bien définie, bien que sa hiérarchie de classes soit limitée. En troisième lieu, AMLO est accessible librement, en accord avec les normes les plus élevées en matière de partage et de publication d'ontologies. Par conséquent, AMLO englobe 17 catégories et 87 caractéristiques, lesquelles sont conformes à trois ontologies largement reconnues, à savoir l'ontologie PROV, l'ontologie OM et le vocabulaire skos.

Objectif et champ d'application

La conception d'AMLO est basée sur les éléments suivants :

- Il est impératif qu'AMLO soit pleinement compatible avec le schéma AML XSD et respecte scrupuleusement la norme en vigueur ;
- De plus, le langage employé dans AMLO devrait être étroitement aligné sur la terminologie de la norme afin de faciliter la compréhension intuitive de l'ontologie par les professionnels de l'industrie familiarisés avec ladite norme mais non pas avec les technologies sémantiques.

3.3.5.2 Ontologie sémantique de la fabrication (MASON)

Le système MASON a été développé par Lemaignan (Lemaignan, Siadat, Dantan, et al., 2006). Et paru en 2006. Ceci représente l'une des premières initiatives visant à élaborer une ontologie avancée spécifique au secteur de la fabrication. MASON comprend un ensemble de 270 concepts

fondamentaux ainsi que 50 propriétés associées à ces concepts. La figure 5 illustre les concepts fondamentaux de MASON, à savoir Entité, Opération et Ressource.

Il est observable que MASON est approprié pour une utilisation dans l'industrie et par les entreprises. De plus, MASON distingue les entités géométriques dotées de caractéristiques géométriques des matières premières. La classe Entity et ses sous-classes sont illustrées dans la figure 6.

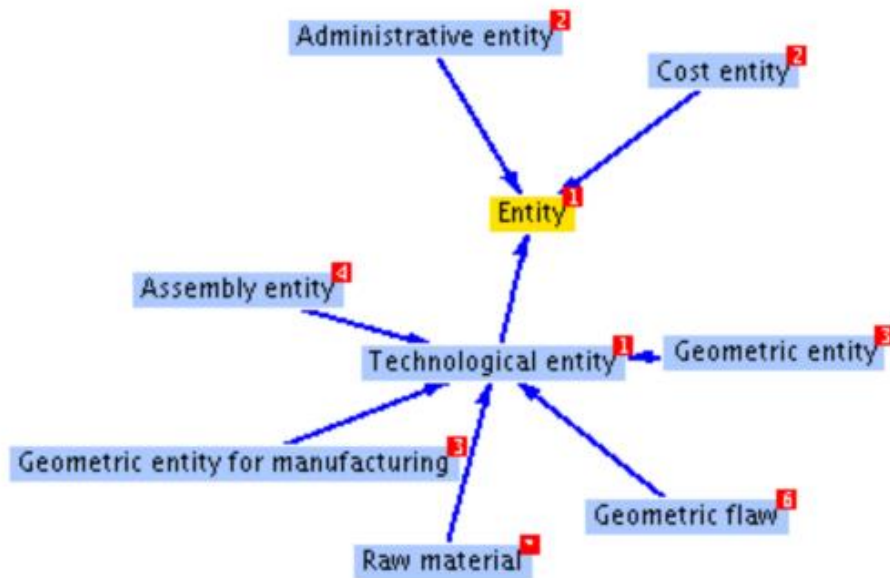


Figure 2.5 : Hiérarchie des classes d'entités dans l'ontologie MASON (S. Lemaignan, A. Siadat, J.-Y. Dantan, et al., 2006)

MASON définit les opérations comme tous les processus liés à la fabrication et à la logistique. En outre, les opérations humaines font également partie de l'ontologie. La figure 7 montre la classe Opération et ses sous-classes.

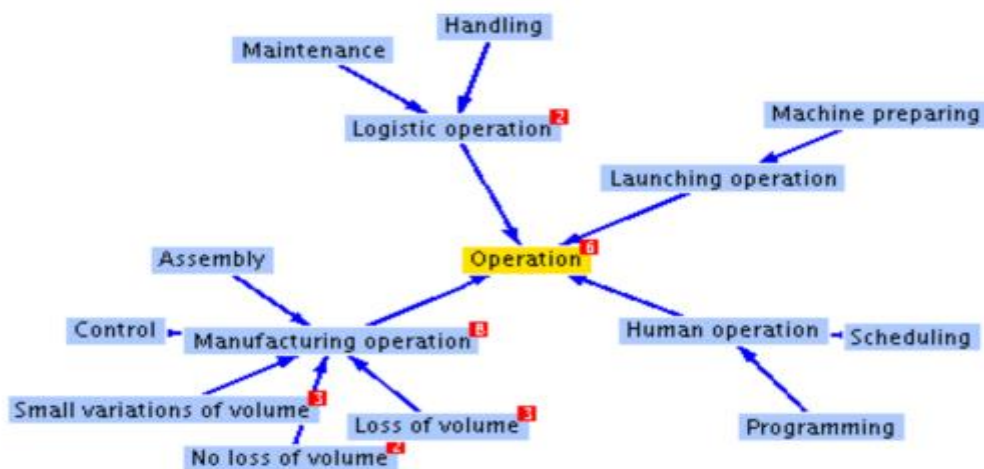


Figure 2.6: Hiérarchie des classes d'opérations dans l'ontologie MASON (S. Lemaignan, A. Siadat, J.-Y. Dantan, et al., 2006)

3.3.5.3 Langage de description des services de la machine (MSDL)

Tout comme MASON, MSDL a été initialement présenté en 2006 par Farhad Ameri et Debasish Dutta (Farhad Ameri & Debasish Dutta, s.d.). Tout comme MASON, MSDL ne comprenait pas initialement une ontologie de base. La MSDL se concentre principalement sur la découverte de services entre les fournisseurs et leurs clients, tandis que la MASON met l'accent principalement sur les processus.

Le document original était divisé en deux parties principales. Le noyau MSDL était conçu pour être une ontologie statique et universelle, constituée de classes fondamentales. L'extension MSDL avait pour objectif d'intégrer des taxonomies supplémentaires issues de la communauté

La MSDL de 2006 est structurée en cinq classes principales, tel qu'illustré dans la figure 7. Le fournisseur représente l'entité principale de la MSDL. Les attributs *hasCustomerFocus*, *hasProductFocus*, *hasIndustryFocus* et *hasMfgService* font référence à des instances des classes *Customer*, *Product*, *Industry* et *MfgService*. Les services proposés par un fabricant sont représentés par *MfgService* (KoreckyMasterThesisOntologiesinIndustry4.0Public, 2018.)



Figure 2.7: Classes principales de MSDL (Farhad Ameri & Debasish Dutta, n.d.)

En outre, MSDL 4.0 importe l'Ontologie des compétences et professions industrielles (ISPO) (Damiano Arena et al., 2018) et l'Ontologie des agents du Common Core Ontologies (CCO)4. ISPO sert de pont entre MSDL et BFO ainsi qu'avec l'Ontologie des agents.

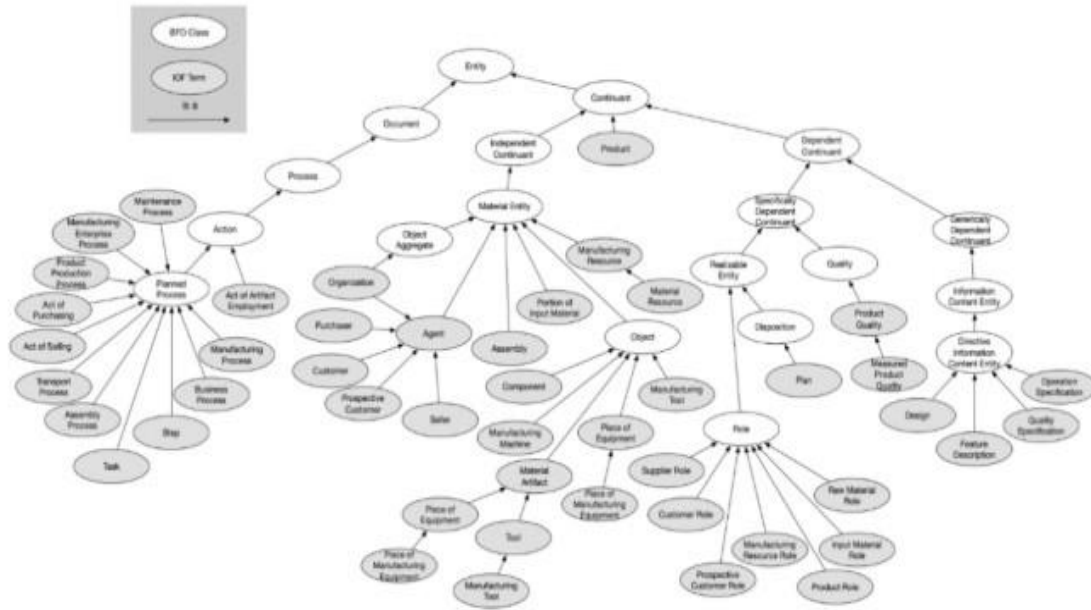


Figure 2.8 : Fragment de la hiérarchie des classes BFO aligné avec les classes portant les étiquettes des vingt-cinq premiers termes fournis par l'IOF (Barry Smith et al., 2019).

3.3.5.4 RAMI

Grangel et ses collègues ont élaboré une ontologie dédiée à RAMI 4.0, comme décrit par Irlán Grangel-González et al. En 2016. Cette ontologie est désignée par le préfixe RAMI, bien qu'elle n'ait pas de nom spécifique. Ainsi, dans le cadre de ce mémoire, nous l'identifierons sous le nom de RAMI. L'ontologie est principalement élaborée pour appuyer les Coquilles Administratives de l'Industrie 4.0, tout en étant compatible avec d'autres concepts du RAMI 4.0, notamment les niveaux hiérarchiques définis dans la norme IEC62264. De plus, l'ontologie est élaborée en vue d'incorporer les normes déjà en place. Les normes IEC61784 concernant la communication, IEC61360 pour l'ingénierie et IEC61508 pour la sécurité ont déjà été prises en compte. Chaque élément de l'industrie 4.0 est distingué par un identifiant URI HTTP unique et universel au sein de cette ontologie. L'ontologie des unités de mesure (OM) est utilisée en ontologie. De plus, il prend en considération l'alignement avec la taxonomie eCI@ss. La description des données des capteurs est conforme à l'ontologie sémantique du réseau de capteurs.

4 Conclusion

Dans ce chapitre, nous avons présenté la notion d'ontologie, ses composants et le rôle joué par celle-ci dans le contexte du Web Sémantique. Nous avons présenté l'ontologie supérieure et quelque exemple de cette ontologie pour illustrer son importance.

AutomationML est une initiative standard de l'industrie 4.0 qui vise à unifier la représentation des données et les API utilisées par les outils d'ingénierie. Elle fournit une base d'échange de données et de connaissances entre diverses disciplines d'ingénierie. La description de cet outil fait l'objet de notre prochain chapitre

Chapitre 3

AutomationML et La Chaudière Industrielle

Partie 1 : Automation ML

1 Introduction

L'ingénierie des systèmes de production est un processus complexe impliquant plusieurs ingénieurs de plusieurs disciplines d'ingénierie exécutant plusieurs activités et utilisant / créant plusieurs artefacts d'ingénierie nécessaires pour enfin être en mesure de construire, gérer et maintenir un système de production (W. Terkaj et al., 2009).

De nombreuses initiatives ont été mises en place afin de concevoir des méthodes d'automatisation performantes pour les secteurs de la fabrication (Sebastian Faltinski et al., 2012) . Depuis un certain temps, la conception des systèmes de production est améliorée grâce à l'utilisation de l'outil Automation Markup Language. La norme AutomationML (AML) est un format d'échange de données ouvert, et basé sur XML. Il a été conçu pour modéliser des systèmes d'automatisation allant de simples composants à des systèmes de production entiers, vastes et complexes. AML permet de représenter différents aspects de ces systèmes, tels que la topologie, la géométrie, la cinématique et le comportement de contrôle. Elle vise à garantir un échange de données cohérent et sans perte lors de la conception des systèmes de fabrication. AML n'est pas un format entièrement nouveau, mais combine et étend des formats existants de manière séquentielle, en allant de la conception de la structure de l'usine jusqu'aux séquences de processus. AML est basé sur CAEX, PLCopen XML et COLLADA. La norme flexible CAEX suit des concepts de modélisation modernes tels que la représentation métamodèle explicite (R. Drath et al., 2010) . PLCopen XML est un format de données XML supplémentaire exploité par AML pour la représentation du comportement (*PLCopen Association: PLCopen XML*, 2012). COLLADA est destiné à la représentation graphique. Dans ce chapitre nous allons présenter en détail l'architecture et la structure d'AML.

2 Architecture générale d'AML

En ce qui concerne l'architecture générale, les dispositions suivantes s'appliquent :

- **Informations de topologie d'installation** : la topologie d'installation constitue la structure de données de niveau supérieur des informations d'ingénierie d'installation et doit être modélisée au moyen du format de données CAEX conformément à la norme IEC 62424 :

- **Informations de référence et de relation** : Les relations entre les informations stockées en externe doivent être stockées avec les mécanismes CAEX. Si nécessaire, les partenaires de liaison associés doivent être publiés dans la description de la topologie d'installation CAEX au moyen des interfaces externes CAEX. Elles doivent être dérivées des classes d'interface standard AML.
- **Informations relatives à la géométrie et à la cinématique** : les informations pertinentes pour la géométrie et la cinématique doivent être stockées sous le format de données COLLADA. Les interfaces COLLADA devant être interconnectées au format de niveau supérieur doivent être publiées en tant que CAEX ExternalInterfaces.
- **Informations logiques** : les informations logiques doivent être stockées en utilisant le format de données PLCopen XML. Si des éléments logiques, par exemple les variables ou les signaux doivent être interconnectés dans le format de niveau supérieur, ils doivent être publiés en tant que CAEX ExternalInterfaces.
- **Référence à d'autres formats de données** : les livres blancs AML peuvent être étendus à l'avenir par des parties supplémentaires spécifiant l'intégration d'autres formats de données XML utilisant les mécanismes de référence AML (Boukadoum & Samia, 2018). La figure 2.1 suivante montre l'architecture générale d'AML.

Le tableau 2.1 présente les standards qui sont supportés par AutomationML aujourd'hui et quelles données ils peuvent représenter :

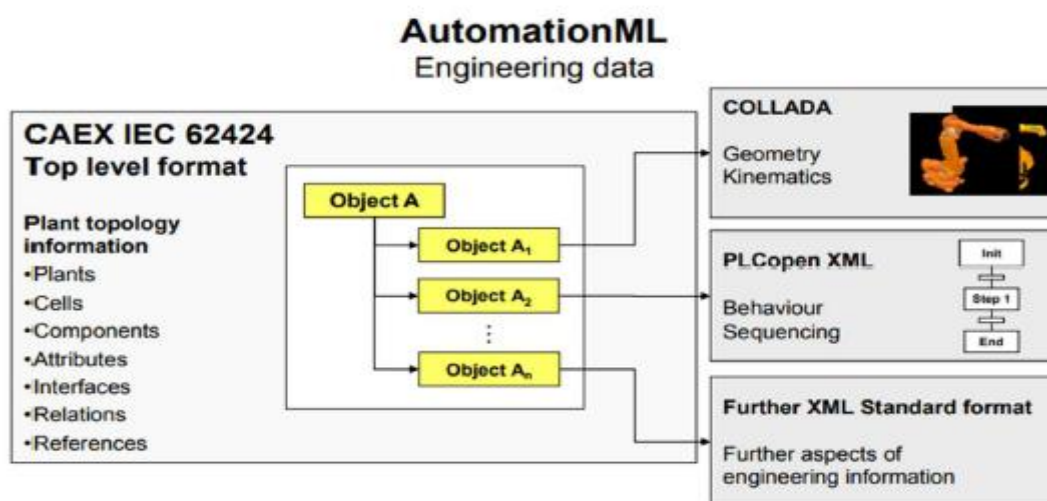


Figure3.1: L'architecture générale de l'AML (R. Drath et al., 2010).

Tableau 1 : Normes prises en charge par AutomationML (Wahlberg & Yixian, 2010)

| Norme | Données |
|-------------|------------------------------------------------------------------------------------------------------------------------------|
| CAEX | Les structures hiérarchiques, telles que la topologie des usines. Les attributs des différents objets et leurs relations. |
| Collada | Les descriptions cinématiques, graphiques et géométriques. |
| PLCopen XML | Le comportement logique, séquences opérationnelles et connexions. |

3 La structure CAEX

AutomationML est basé sur CAEX (Computer Aided Engineering Exchange) qui modélise la topologie du système suivant un paradigme orienté objet. Dans la structure CAEX, les objets sont organisés dans une structure hiérarchique. Ces objets peuvent être un instrument, un équipement ou un module de production complet. Chaque objet comprend un ensemble de propriétés de la géométrie, de dynamique et de logique (les séquences opérationnelles, le comportement, et l'information de contrôle) décrivant l'objet et peut être interconnecté avec un autre objet via des interfaces. Il est possible d'intégrer des bibliothèques d'utilisateurs et de fournisseurs ainsi que des bibliothèques de projets. En outre, une conception de système de haut en bas et de bas en haut est prise en charge. L'innovation technique de cette approche est l'unification syntaxique et sémantique des données. Cela permet de découpler les algorithmes de configuration requis des sources de données (M. Schleipen et al., 2008).

CAEX se compose de trois types de bibliothèques : RoleClassLibraries, InterfaceClassLibraries, et SystemUnitClassLibraries. En plus de ces bibliothèques, il y a la hiérarchie InstanceHierarchy où l'installation industrielle spécifique est modélisée. La figure 2.2 représente les éléments liés les uns aux autres :

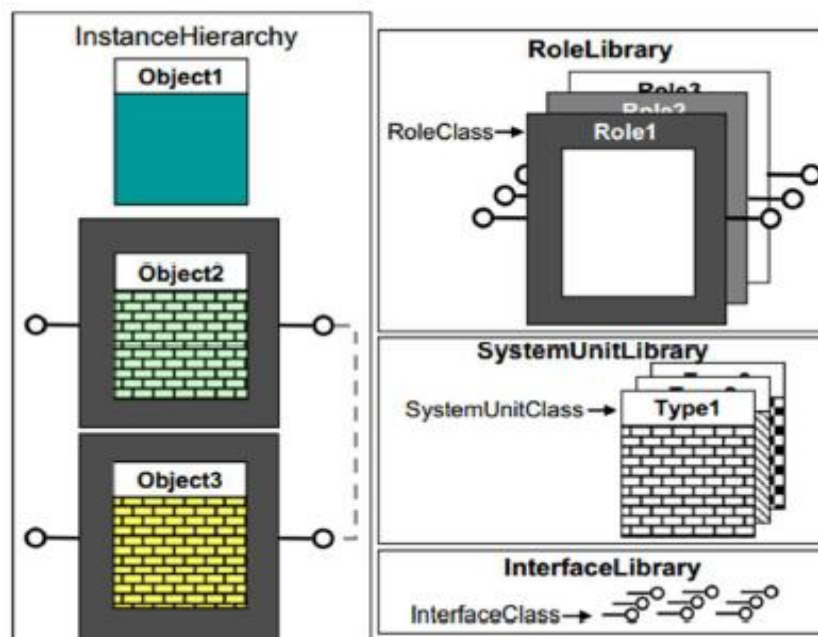


Figure 3.2: Le modèle CAEX (M. Schleipen & O. Sauer, 2009).

- A. **La bibliothèque RoleClassLibrary** contient une collection de fonctionnalités possibles qui peuvent être fournies par l'équipement de l'usine. Une classe de rôle (RC) définit un objet physique ou logique en tant qu'abstraction d'une réalisation technique concrète, indépendante du fournisseur, par exemple un robot ou un capteur. Une sémantique fonctionnelle est ainsi attribuée à un objet, ce qui permet à un outil de l'interpréter automatiquement. Les RC peuvent également définir des attributs généralement attendus pour ce type d'objet, par exemple une charge utile pour un RC de robot. En outre, les RC peuvent être assignés à des objets dans la bibliothèque SystemUnitClassLibrary et InstanceHierarchy pour spécifier un type d'objet.
- B. **La bibliothèque de classes d'interface** définit toutes les interfaces nécessaires pour décrire le modèle de l'usine. Une classe d'interface (CI) peut être utilisée : a) pour spécifier les

relations entre les objets d'une topologie d'installation, par exemple pour connecter un capteur à un automate ; ou b) pour spécifier des références à des informations stockées en dehors du fichier CAEX, par exemple pour attribuer une description 3D à un objet.

- C. **La bibliothèque SystemUnitClassLibrary** comprend des collections d'objets d'équipement de solutions spécifiques au fournisseur. Ces objets peuvent être mis en correspondance avec les exigences du système, définies par les classes de rôle, et utilisés pour mettre en œuvre la conception de l'installation. Une classe d'unité de système (SUC) définit une réalisation technique concrète d'un objet physique ou logique, représentant ainsi une instantiation spécifique pour un CR. Les classes d'unité de système sont instanciées dans l'InstanceHierarchy
- D. **InstanceHierarchy** décrit la topologie de l'usine, y compris l'équipement concret d'un projet réel, c'est-à-dire les données d'instance. La hiérarchie des instances contient toutes les données, y compris les propriétés, les interfaces, les classes de rôle, les relations et les références. L'application de l'AML aux projets d'ingénierie constitue déjà une amélioration importante, car elle facilite l'échange de données entre les parties prenantes du projet et définit un vocabulaire spécifique au projet auquel tous les ingénieurs peuvent se référer. Néanmoins, il y a encore un manque d'infrastructure pour soutenir l'AML. Les activités d'ingénierie avancées dans les différentes disciplines d'ingénierie et leurs outils correspondants, par exemple l'établissement de liens entre les données, la propagation des changements dans les ensembles de données connectés, l'analyse des données et la vérification de la cohérence. L'existence d'une ontologie pour l'AML permet de combler ces lacunes (Grangel-González, 2018).

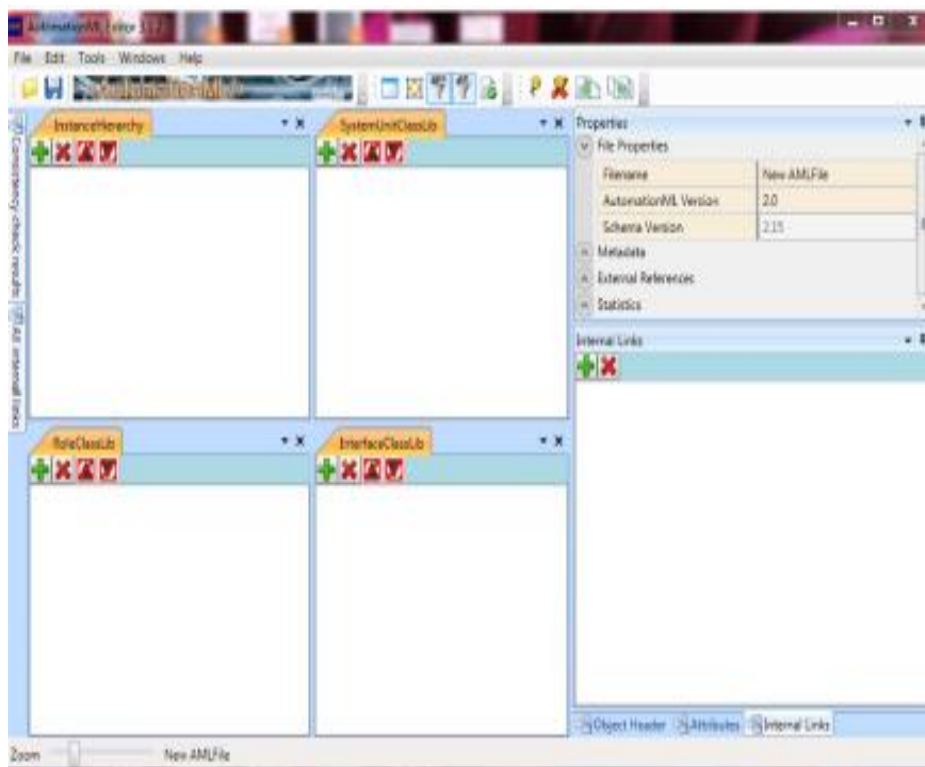


Figure 3.3: L'éditeur AutomationML (Schmidt & Lüder, 2015).

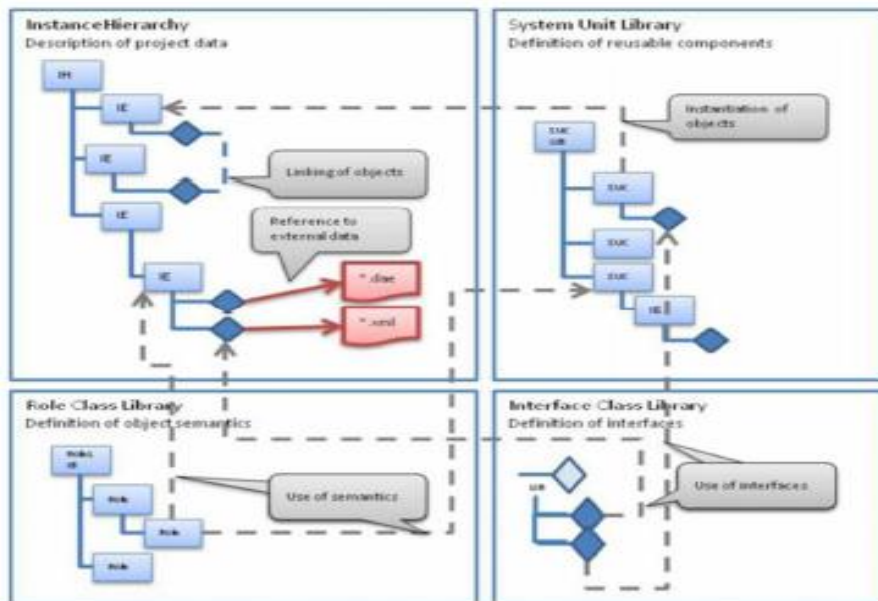


Figure 3.4: Description Topologique de l'Architecture de l'AutomationML(AutomationML e.V.: AutomationML , 2010)

4 Relations entre objets CAEX

L'établissement des relations entre les objets réels est également indispensable. Elles expriment des relations physiques ou logiques entre objets. AutomationML utilise les concepts supportés par CAEX pour ce faire :

- Relation parent-enfant :

Entre les instances d'objet,

Entre les classes du même type (Mit Beitragen Von Rainer Drath et al., 2010)

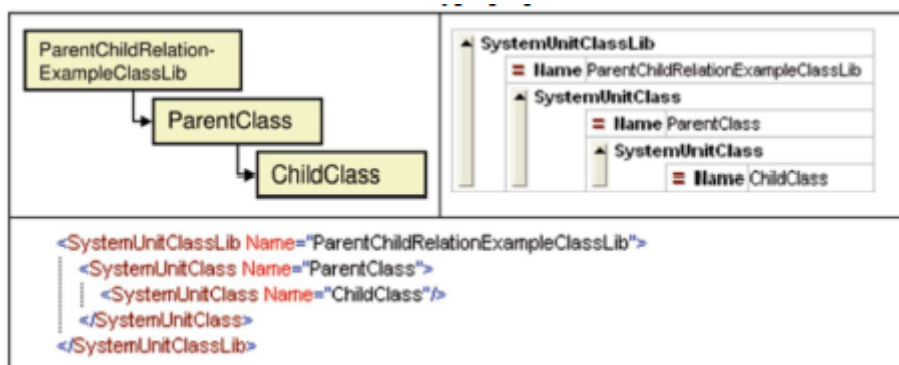


Figure 3.5: Relation parent-enfant dans une bibliothèque de classes.

- Relation d'héritage :

Relation d'héritage entre SystemUnitClasses

Relation d'héritage entre RoleClasses

Relation d'héritage entre les relations d'instance de classe InterfaceClasses.

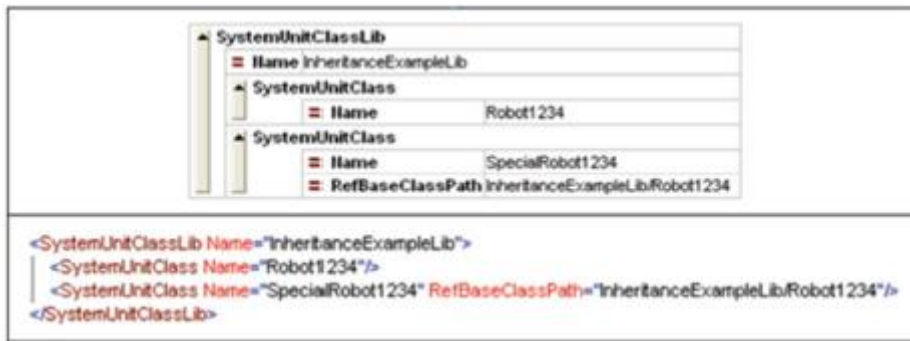


Figure 3.6: Exemple de relation d'héritage entre classes.

- Relation de classe-instance :

Entre un <SystemUnitClass> et une instance de celle-ci

Entre un <RoleClass> et une instance de celle-ci

Entre une interface <InterfaceClass> et une instance de celle-ci.

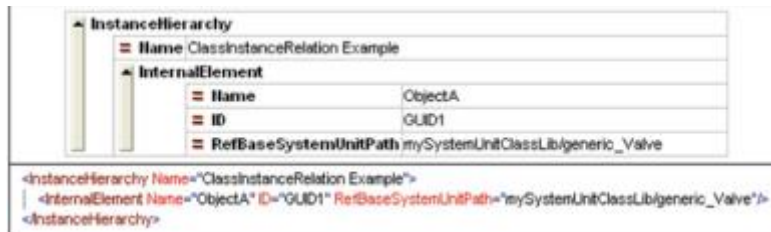


Figure 3.7: Exemple de relation de classe-instance.

- Relation instance-instance :

Entre les instances d'objet

Entre les données stockées en dehors de CAEX.

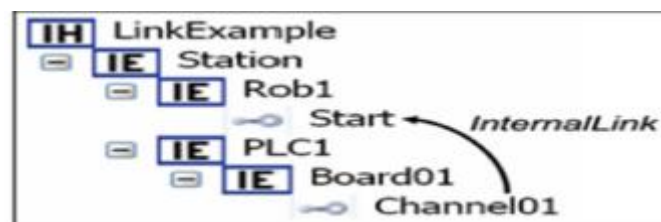


Figure 3.8: Relation entre deux instances.

Les quatre relations présentées ci-dessous sont illustré dans la figure suivante :

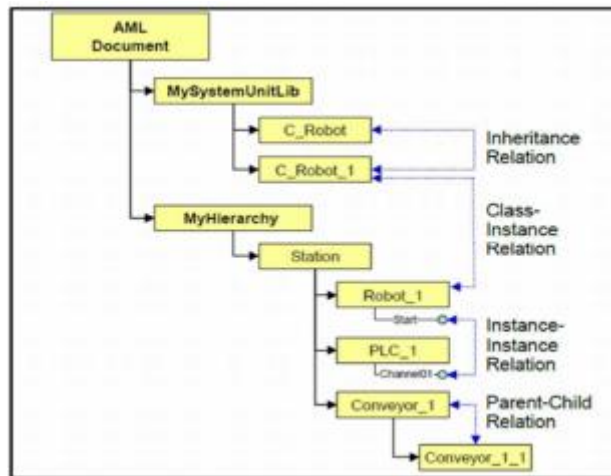


Figure 3.9: Les relations entre les objets CAEX.

5 Stockage des descriptions logiques dans AutomationML

AutomationML utilise un modèle de données commun pour décrire les deux types d'informations logiques, à savoir les diagrammes de fonctions séquentielles (SFC) comme l'un des langages de programmation PLC définis dans la norme CEI 61131-3 [CEI61131-3]. Ainsi, les règles de transformation vers et depuis les SFC à partir de différents formats d'entrée sont des éléments essentiels du concept logique d'AutomationML.

Les représentations logiques complètes sont stockées dans des documents externes, tandis que le format de niveau supérieur ne contient que des informations publiées sélectionnées. Le format de données cible pour les informations logiques est PLCopen XML. Pour stocker une description logique donnée dans AutomationML, il faut la traduire en SFCs décrits par une représentation XML PLCopen en suivant strictement les règles définies dans ce document.

Pour découpler le format cible PLCopen XML des divers formats de données d'entrée et de sortie, AutomationML définit une couche intermédiaire de modélisation (IML). De cette manière, les règles de transformation complexes de et vers les spécificités PLCopen XML ne doivent être définies qu'une seule fois, pour chaque format d'entrée spécifique, seules les règles de transformation vers la couche intermédiaire plus simple doivent être définies, ce qui améliore l'extensibilité d'AutomationML pour les nouveaux formats d'entrée.

Pour résumer les principales catégories et leur donner un contexte formel, la couche intermédiaire de modélisation (IML) est définie pour clarifier et simplifier le processus de mise en correspondance des différents types de modèles avec AutomationML. Néanmoins, elle ne constitue pas un nouveau langage de modélisation.

Pour l'importation d'une description logique à partir de SFC, un outil doit les interpréter de manière appropriée afin de restaurer le format d'origine ou de créer une autre représentation de l'information. Ainsi, les outils peuvent également utiliser AutomationML pour transférer des formats logiques. Dans ce cas, l'utilisateur doit être conscient de la perte possible d'informations, car AutomationML ne peut pas compenser la puissance de modélisation différente des formats, par exemple un diagramme PERT peut être interprété comme un diagramme de Gantt, mais certaines informations de synchronisation spécifiques à PERT seront perdues (Boukadoum & Samia, 2018).

6 Modèles de séquence et de comportement dans AutomationML

Pour les descriptions logiques, AutomationML couvre les données de différents outils et disciplines, et prend en charge différentes phases de l'ingénierie avec différents niveaux de détail. AutomationML simplifie l'ingénierie des systèmes de production, depuis les premières étapes de l'ingénierie, telles que la conception de l'usine, jusqu'à la mise en service finale des systèmes complets. Ainsi, différents types d'informations logiques appartenant à un système de production ou à des pièces individuelles doivent être stockés. Cette variété d'informations peut être différenciée en deux concepts principaux, à savoir le séquençement et le comportement, mais peut se chevaucher en fonction du point de vue et de l'utilisation des éléments dans AutomationML.

- Description du comportement des objets AutomationML en tant que modèle non contrôlé de réaction d'une pièce à des entrées externes, généralement utilisé dans une boîte blanche.
- Description du séquençement en tant que comportement contrôlé requis d'un système ou d'une pièce, qui représente souvent un programme exécuté dans un ou plusieurs contrôleurs, par exemple un contrôleur de robot, un automate programmable, etc.

Les deux concepts sont généralement utilisés dans AutomationML dans les contextes suivants

Comportement incontrôlé d'une seule unité mécatronique : Un exemple de ce type est le comportement d'une pince. Les interfaces de cet élément sont nécessaires pour déclencher le comportement ou signaler ses états. Dans la plupart des cas, elles représentent les mêmes signaux que ceux de l'élément physique réel.

Séquences de cellules ou d'installations : Les séquences de cellules ou d'installations sont généralement des descriptions d'actions ultérieures qui servent de données d'entrée de haut niveau pour la programmation d'automates, par exemple sous la forme de diagrammes de Gantt. Elles sont normalement liées à des éléments composés tels qu'une cellule ou un contrôleur. Les interfaces sont nécessaires pour l'interaction avec d'autres éléments de la description logique et consistent en des signaux réels ou des simplifications de ces signaux. L'évolution des séquences commence par une description abstraite de haut niveau des opérations requises d'une unité à plus grande échelle (cellule, ligne, usine, etc.) et se termine par des programmes exécutables, généralement avec plusieurs raffinements entre les deux.

Les dispositifs intelligents peuvent avoir un comportement et une séquence. D'un point de vue externe, les programmes des dispositifs peuvent être contrôlés au moyen du comportement. D'un point de vue interne, les programmes individuels peuvent être décrits comme des séquences. Un exemple de ce type combiné est le comportement d'un robot réalisé par des programmes interagissant avec un automate programmable (PLC) en tant que contrôleur de cellule global. Il est important de noter que cet exemple peut être interprété à la fois comme un comportement ou une séquence, selon le point de vue.

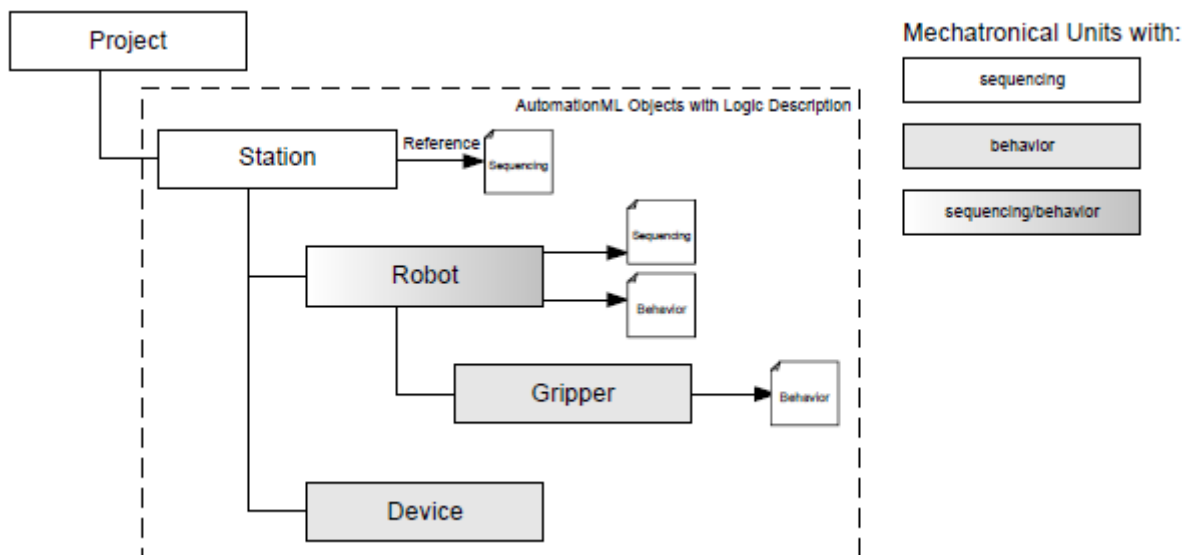


Figure 3.10: Types de descriptions logiques dans AutomationML

AutomationML fournit une riche sélection de modèles typiques pour les phases importantes du processus d'ingénierie et des données :

- Les diagrammes de Gantt sont généralement utilisés pour définir les séquences d'opérations au cours des premières phases de planification.
- Les diagrammes PERT sont utilisés de manière similaire, mais ils peuvent également exprimer des conditions temporelles complexes des processus.
- Les diagrammes d'impulsion permettent de décrire les séquences en détail et d'introduire des signaux réels et des conditions supplémentaires.
- Les diagrammes d'état sont utilisés pour décrire le comportement des unités mécatroniques.
- Les SFC peuvent être utilisés pour décrire des séquences complexes avec des boucles et une exécution conditionnelle ; dans les phases d'ingénierie ultérieures, les SFC peuvent décrire des programmes PLC exécutables, y compris le mappage avec le matériel de commande réel. La description logique d'AutomationML est conçue pour être extensible à d'autres modèles.

6.1 Les graphiques de Gantt

Les graphiques de Gantt sont une représentation graphique des modèles d'événements discrets généralement utilisés pour décrire l'ordre et le temps d'exécution des activités, qui sont représentées sous forme de barres, à un niveau élevé d'abstraction.

Ils sont utilisés dans les premières phases d'ingénierie de l'usine pour représenter le calendrier des processus de fabrication. Les informations principales stockées dans les diagrammes de Gantt sont les heures de début et de fin des barres et les relations prédécesseur / successeur entre les barres. Par conséquent, les principaux moyens de modélisation des diagrammes de Gantt sont :

- Barres avec les points de début et de fin ainsi que la durée,
- Ordonner des relations représentant une relation prédécesseur-successeur entre des barres (AutomationML, 2010).

Habituellement, les diagrammes de Gantt permettent de modéliser des ensembles ordonnés d'activités séquentielles et simultanées. Ainsi, les diagrammes de Gantt permettent la modélisation d'une divergence et la convergence des flux de contrôle.

Ils ne fournissent pas de moyens de modélisation pour la modélisation des alternatives ou des cycles. Les diagrammes de Gantt n'ont pas d'échelle de temps fixe, c'est-à-dire qu'il est possible de représenter les points de début et de fin des barres par rapport à une horloge globale mais également par rapport à plusieurs horloges locales démarrées par des dates de fin de barres. Néanmoins, la version la plus utilisée est l'horloge globale, également prise en charge par AutomationML.

Un exemple de diagramme de Gantt est donné à la figure 2.11. Il décrit les processus de transport d'une pièce et son usinage par un robot dans une cellule.

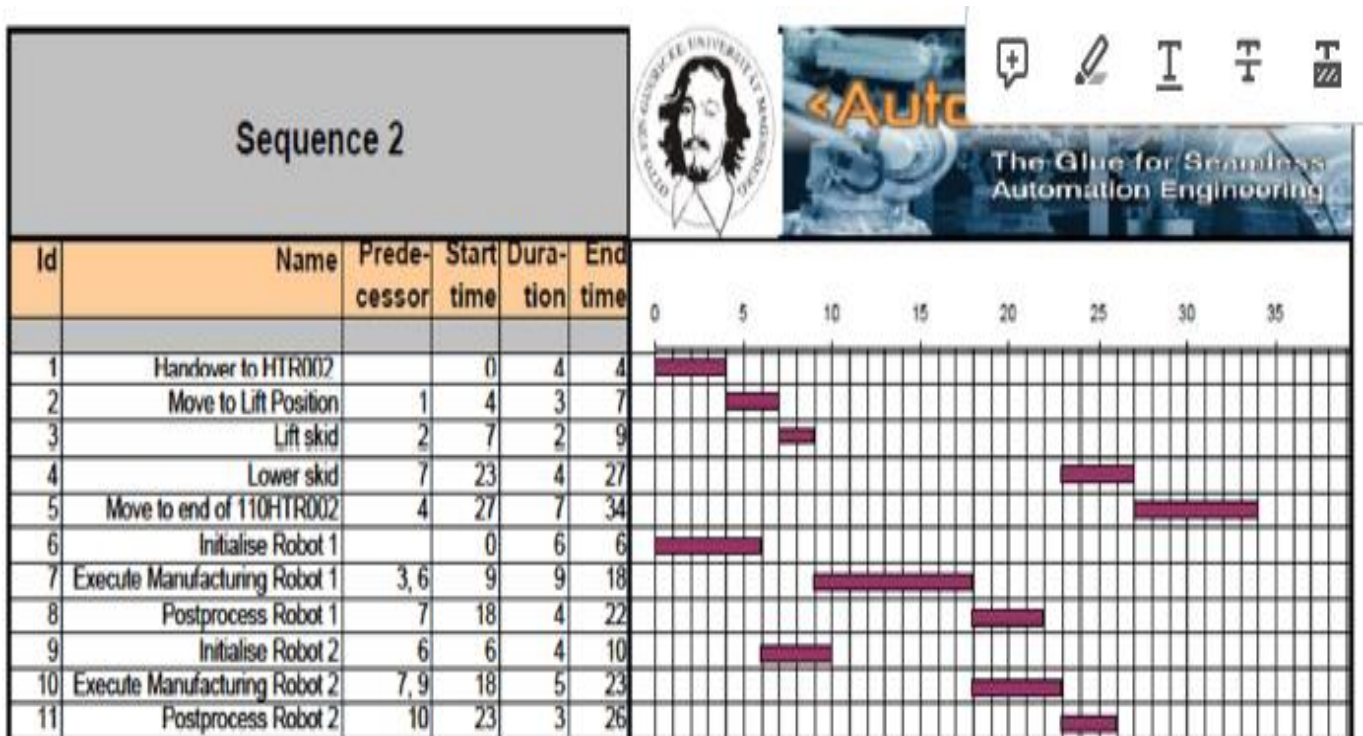


Figure 3.11: Exemple d'un diagramme de Gantt.

6.2 Les graphiques de PERT

Les diagrammes PERT appartiennent au groupe de plans de réseau. Ils sont utilisés pour décrire et analyser les relations temporelles de l'exécution d'un ensemble de nœuds interdépendants. Les plans de réseau sont appliqués dans les premières étapes du processus d'ingénierie pour représenter le calendrier et l'interdépendance des processus de fabrication avec une capacité enrichie pour la description du temps par rapport aux diagrammes de Gantt. Les informations principales stockées dans les graphiques PERT sont :

Nœuds avec le point de départ le plus récent, le point de fin le plus proche et le plus récent, la durée, ainsi que le délai ordre des relations représentant une relation prédécesseur-successeur entre nœuds. Généralement, les relations d'ordre des plans de réseau permettent de définir les règles suivantes pour deux nœuds:

1. Relation de fin de démarrage : le nœud 2 commence après la fin du nœud 1.
2. Relation start-start : le nœud 2 démarre après le début du nœud 1.

3. Relation début-fin : le noeud 2 se termine après le début du noeud 1.
4. Relation de fin : Le noeud 2 se termine après la fin du noeud 1.

Étant donné que seules les relations de début et de fin sont couramment utilisées dans les diagrammes PERT, AutomationML ne prend en charge que ce type de relation d'ordre.

Habituellement, les diagrammes PERT permettent la modélisation de noeuds concurrents, c'est-à-dire la modélisation d'une divergence ET et la convergence de flux de contrôle. Comme les diagrammes de Gantt, ils ne fournissent pas de moyens pour la modélisation des alternatives ou des cycles. La figure 2.12 suivante montre un exemple de graphique PERT.

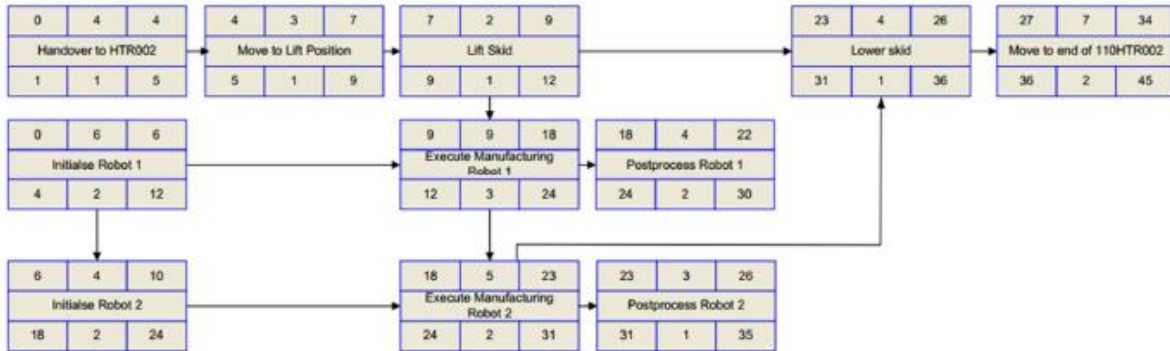


Figure 3.12: Exemple de graphique PERT.

6.3 Les diagrammes d'impulsion

Les diagrammes d'impulsion sont utilisés pour décrire le cours temporel des états des composants du système ainsi que des signaux entre eux pour activer les changements d'état. Pour décrire les séquences de valeurs et les relations des signaux dans le temps, une échelle de temps globale ou des horloges internes des composants peuvent être utilisées. Certaines conventions sont faites pour la désignation des éléments de modélisation des diagrammes d'impulsion comme illustré dans la figure 2.13.

Un objet dans le diagramme représentant un signal ou un état d'une unité mécatronique avec différentes valeurs possibles sera appelé une ressource. Dans le diagramme Impulse chaque ressource sera représentée par un ensemble de lignes indiquant les états possibles de la ressource. Les ressources peuvent être organisées au sein de groupes représentant les hiérarchies de l'unité mécatronique. Une ressource peut atteindre des états de ressource, par ex. indiquant des valeurs discrètes différentes d'un signal ou d'une variable d'E / S réel associé. Les modifications de l'état de la ressource sont nommés changements d'état des ressources.

Une séquence d'états de ressources et de changements d'état de ressource dans le diagramme est appelée un flux d'état de ressource. Rester dans un état de ressource est indiqué par une ligne horizontale ; Les changements d'état des ressources sont visualisés par des lignes reliant l'état des ressources qui les ont précédées et qui ont réussi.

Les signaux dans le diagramme Impulse sont des "liens" entre les flux d'états de ressources représentant les dépendances. Lors de la saisie d'un nouvel état de ressource, les signaux peuvent déclencher des changements d'état d'une autre ressource (signal interne). Les ressources peuvent également déclencher des signaux internes lorsqu'elles quittent les états après une durée donnée.

La chronologie dans un diagramme Impulse représente une échelle de temps avec des points discrets dans le temps qui sont utilisés globalement dans le système. À tout moment, les signaux peuvent être déclenchés à partir de la chronologie (signal externe).

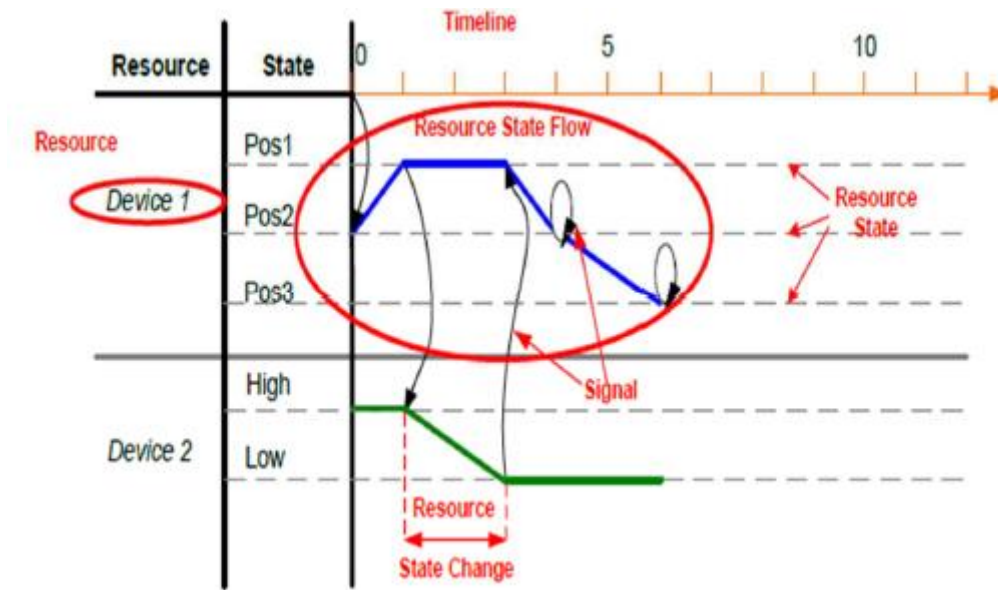


Figure 3.13: Exemple de diagramme d'impulsion avec conventions de dénomination.

6.4 Diagramme d'état

Le diagramme d'état est un modèle à événements discrets. Ils sont largement connus dans le cadre du langage UML (Unified Modeling Language). Habituellement, ils sont utilisés pour décrire le comportement d'un système en termes de son espace d'état accessible. Par conséquent, dans la chaîne d'ingénierie des systèmes de fabrication, ils sont principalement appliqués lors de la conception des composants du système pour représenter leur comportement incontrôlé. Dans la simulation et la vérification des systèmes contrôlés, ils servent généralement de contrepartie au contrôleur pour fermer la boucle de contrôle. Un exemple de diagramme d'état considéré dans AutomationML est donné à la figure 2.14 :

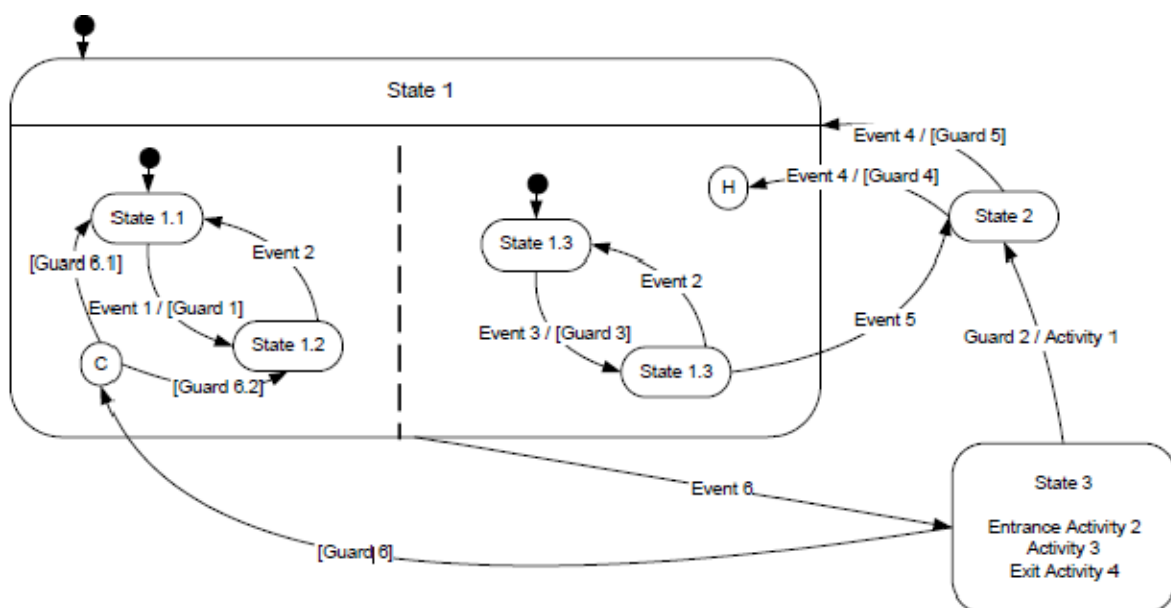


Figure 3.14: exemple de diagramme d'état

6.5 Les graphiques de fonctions séquentielles

Les tableaux de fonctions séquentielles (SFC) sont l'un des langages de programmation de la IEC 61131-3. Ils ont été développés pour fournir des moyens permettant de mettre en oeuvre facilement des séquences d'opérations dans un système contrôlé.

Par conséquent, les SFC servent à la représentation de séquences d'opérations basées sur l'état avec une variété de conditions temporelles pour contrôler le minutage d'exécution des opérations et fournissent des règles de base pour le développement de programmes de contrôle. Les principaux moyens de modélisation des SFC, mais non exhaustifs, sont les suivants :

- Les étapes représentent des situations stables au sein du système permettant l'exécution d'opérations.
- Les transitions représentent le changement basé sur la condition entre les étapes.

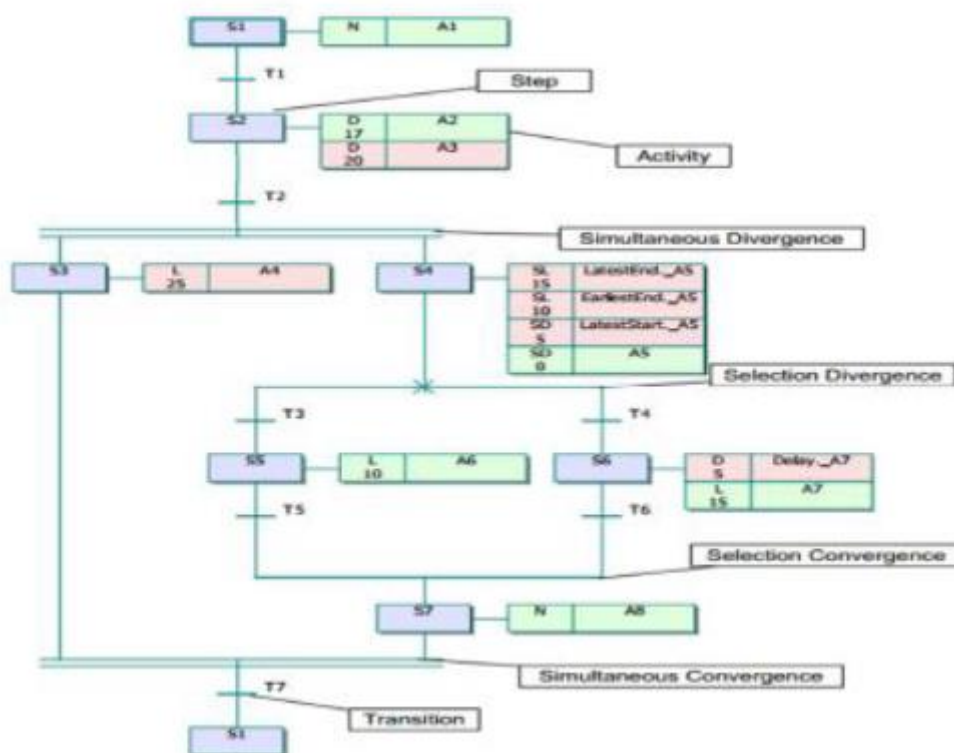


Figure 3.15: Exemple de SFC dans la CEI 61131-3.

7 Conclusion

Dans cette partie, nous avons exposé les principes fondamentaux de l'AML qui utilisent CAEX pour représenter la structure des éléments du système, ainsi que la description logique de ces éléments avec PLCopen XML. La chaudière industrielle a été sélectionnée comme sujet d'étude car elle joue un rôle essentiel et crucial dans tous les complexes pétrochimiques. La présentation de celle-ci est abordée dans la partie suivante.

Partie 2 : La chaudière Industrielle

1 Introduction

Cette partie se base sur l'étude de la supervision d'une chaudière industrielle dans le complexe CP2K (Complexe Pétrochimique de Skikda). La chaudière est l'équipement le plus cher et le plus sensible dans les complexes industriels. Son fonctionnement doit être permanent et se faire en toute sécurité. Elle constitue un élément critique dans beaucoup de systèmes industriels puisqu'elle produit l'énergie nécessaire pour le fonctionnement d'autres procédés. Son arrêt implique l'arrêt total de toutes les unités dépendantes du complexe voire de toute la production de l'usine, ce qui engendre des pertes financières très importantes. Par conséquent, cet équipement doit être supervisé et surveillé de façon très minutieuse car son dysfonctionnement peut représenter de grands dangers aussi bien pour l'usine que pour l'environnement géographique.

La chaudière est un générateur de vapeur, servant à élever la température de l'eau jusqu'au changement d'état, c'est-à-dire produire la vapeur qui est nécessaire pour le fonctionnement des turbopompes, des turbocompresseurs, des turbos-ventilateurs.

2 Présentation du Complexe Pétrochimique CP2K

2.1 Implantation de l'usine

Le complexe pétrochimique CP2K est implanté à l'intérieur de la zone industrielle de Skikda, d'une superficie de 166800 m². Il se trouve sur la côte à 06 km de la wilaya de Skikda. La position géographique est limitée comme suit (BOUNEB lobna, 2015):

- Au Nord : La mer Méditerranée
- Au sud : La route principale de la zone industrielle.
- A l'Est La force d'intervention et de réserve.
- A l'Ouest : CP1K (complexe matière plastique).



Figure 3.16 : Image satellitaire de zone industrielle Skikda et Position géographique d'unité CP2K (Google earth, 2024).

2.2 Description de l'unité CP2K SKIKDA

L'Unité CP2K a été mise en exploitation en 2005. C'est une unité opérationnelle de la Société ENIP. Elle est située dans la zone industrielle « plateforme et pôle pétrochimique de SKIKDA ».

La production de polyéthylène à haute densité à partir d'éthylène en tant que principale matière première, est basée sur le procédé de réaction chimique fortement exothermique (800 Kcal/Kg approximatif.) à une température se situant dans l'intervalle [85-110] °C et à une pression de 42-44 kg/cm² (PHILLIPS PETROLEUM COMPANY) (procédé à particules). Ce dernier consiste essentiellement en la polymérisation catalytique de l'éthylène dans un réacteur tubulaire continu fermé, en phase liquide (formant une suspension dans l'isobutane) (El. Mechhoud & M. Rouainia, 2014).

Le complexe est composé de trois zones principales qui sont :

- Zone d'off site : Elle comporte les utilités (chaudières, air, azote, eau dessalée, eau anti-incendie, eau potable et détente de gaz) ; l'activation du catalyseur se fait aussi dans cette zone.
- Zone humide : Elle s'appelle aussi zone de réaction, les matières premières avant d'entrer dans le réacteur, elles passent par les traiteurs afin d'éliminer les impuretés.
- Zone sèche : Elle contient les éléments suivants : extrudeuse, soufflantes, silos de stockage des produits finis (poudre et granulé) et ensachage (BOUNEB lobna, 2015).

2.3 Présentation de l'unité utilité

Cette unité est le cœur du complexe de matière plastique de Skikda, elle alimente ce dernier en eau et en énergie. Elle se compose de deux sections :

2.3.1 Section d'eau

2.3.1.1 Prise de l'eau

Cette section a pour fonction l'alimentation de toutes les unités de complexe en cette matière dont une partie servira comme matière première pour la fabrication de l'eau distillée.

Aussi elle assure l'alimentation du réseau anti-incendie. Elle est équipée de :

- Quatre filtres rotatifs.
- Quatre pompes centrifuges verticales dont l'une est entraînée par une turbine et débitant chacune 7500 m³/h.
- Deux turbines avec une production de 7500 m³/h.
- Deux pompes anti-incendie un diesel et l'autre électrique destinées pour l'alimentation du réseau anti-incendie et débitant chacune 1000 m³/h.
- L'eau de mer est en effet, traitée avant son utilisation avec l'hypo chlorure de sodium NaCl.

2.3.1.2 Distillation de l'eau de Mer

Cette section a pour fonction la fabrication de l'eau distillée à partir de l'eau de mer qui servira comme eau de refroidissement pour divers équipements de toutes les unités du complexe dont une partie va servir comme eau d'appoint pour la fabrication de l'eau déminéralisée. Elle assure l'alimentation des complexes voisins : RA1K, GL1K.

2.3.1.3 Déminéralisation de l'eau

La section de déminéralisation purifie l'eau distillée et le condensat de vapeur. Les sections de déminéralisation de l'eau sont composées de deux polysheurs complets avec réseau d régénération automatique, et un groupe d'équipements de régénération de réservoir :

- Deux réservoirs d'eau.
- Un réfrigérant de condensat.
- Trois filtres de condensat.
- Un réservoir de condensat.

2.3.1.4 Traitement des effluents

C'est une section très importante dans le complexe qui permet le traitement des effluents provenant de toutes les unités de production, ainsi que les eaux de pluie pour éviter la pollution de la mer. Le traitement est effectué en faisant passer les eaux usées à travers une série de bassin, des filtres et de déshuileurs pour séparer la boue et les huiles. La boue décantée sera séchée dans de grands bassins. Les huiles seront recyclées vers l'unité éthylène pour être utilisées comme combustible des fours. Les eaux passent dans le bassin de traitement biologique pour éliminer les particules microscopiques qui sont très toxiques grâce aux bactéries. Ces bactéries sont cultivées dans un bassin et entretenues par des aérateurs d'oxygène et des injections de Méthanol (CH_2OH) et de l'Ammoniaque (NH_3).

Après ces étapes l'eau passe dans des bassins pour enlever les matières en suspension de cette eau par précipitations, par floculation et enfin par décantation puis elle passe à travers des filtres de gravier et de sable pour arriver dans le bassin de rejet à la mer pour être neutralisée par H_2SO_4 à $\text{pH}=7.00$ et une injection d'eau de Javel pour être rejetée à la mer.

2.3.2 Section d'énergie

2.3.2.1 Génération de vapeur (La Chaudière)

➤ Définition

La chaudière est un récipient alimenté en eau douce qui par chauffage se transforme de façon continue en vapeur. Les particularités de la chaudière sont :

- Type Mitsubishi CE VU 60,
- Pression à la sortie de la surchauffeur 44 kg/cm^3 ;
- Température à la sortie des surchauffeurs 45°c ;
- Température d'eau d'alimentation 145°c ;
- Température de l'air Ambient 20°c ;
- Combustible : - Gaz naturel
- Gaz secours (ville)

2.3.2.2 Les principaux composants de la chaudière

2.3.2.2.1 Les équipements

➤ **Chambre de combustion (foyer)**

La chaudière verticale est du type à circulation naturelle, avec un ballon d'eau, le faisceau de convection, se compose de tubes de 63,3 mm de diamètre, et le foyer se compose de tubes de 76,3mm de diamètre les faisceaux de tubes sont disposés en ligne. Le foyer est refroidi à l'eau par des tubes placés coté à coté ce qui assure un taux de refroidissement élevé par rapport au volume du foyer, la vapeur produite dans les tubes du foyer et ceux de la chaudière est envoyée au ballon de vapeur.

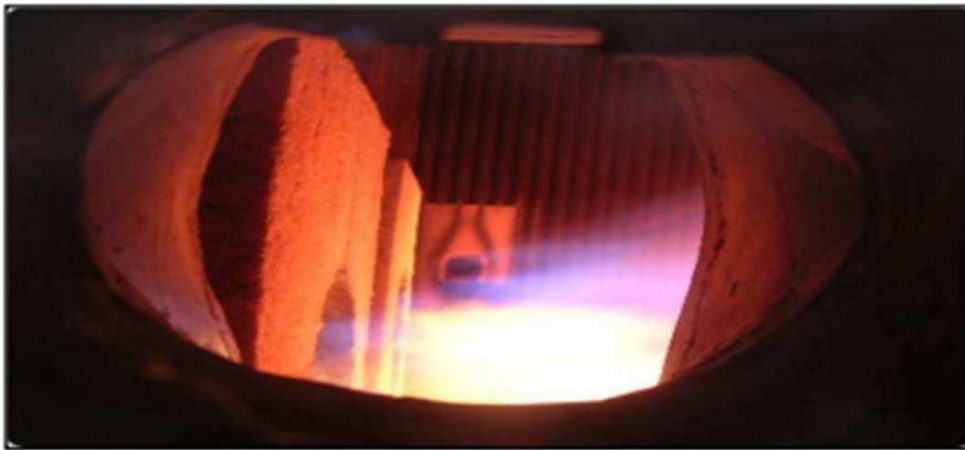


Figure 3.17: la chamber de la combustion (Khadiche Manel & Amiar Bouchra, 2018).

➤ **Surchauffeur**

La surchauffeur est du type vertical, à surface de chauffage suffisante pour élever la température de la vapeur.

➤ **Enveloppe**

Elle se compose d'une paroi fine interne et d'une proie extérieure en tôle keeston.

➤ **Réfractaires et calorifuge**

La chaudière est totalement calorifugée au moyen de matériaux isolant pour minimiser la perte de chaleur par l'enveloppe.

➤ **Souffleur de suie**

La chaudière est équipée des souffleurs de suie pour nettoyer les tubes de la chaudière et de surchauffeur.

➤ **Économiseur**

L'économiseur réchauffe l'eau d'alimentation en récupérant la chaleur des gaz de combustion.

➤ **Réchauffeur d'air**

La récupération de chaleur des gaz de combustion de la chaudière est réalisée par le réchauffeur d'air celui-ci absorbe la chaleur des gaz de combustion et la transmet à l'air entrant (Ferrag Sihèm & Korzet Yassine, 2013).

➤ **Ballon supérieur**

À la sortie de l'économiseur, l'eau d'alimentation monte vers un réservoir situé dans la partie supérieure de la chaudière, dont lequel se collecte le mélange eau vapeur. C'est à son niveau que s'effectue la séparation entre la phase liquide et la phase vapeur.

➤ **Ballon inférieur**

C'est un autre réservoir en bas de la chambre de combustion. Il joue le rôle de collecteur d'eau, il est toujours plein pour des raisons de sécurité de la chaudière, afin de pouvoir évacuer de l'eau à travers le tube porteur vers les écrans au cas de niveau bas (Khadiche Manel & Amiar Bouchra, 2018)

➤ **Désurchauffeur**

Ce dispositif consiste à injecter dans la vapeur une quantité d'eau réglable (Amira & Samiya, n.d.).

➤ **Les brûleurs**

La chaudière se compose de quatre brûleurs alimentés en gaz naturel. Ce sont des éléments générateurs de calories servant à produire la vapeur en chauffant l'eau qui circule dans les tubes de la chambre de combustion.



Figure 3.18: les brûleurs (Khadiche Manel & Amiar Bouchra, 2018).

➤ **Tubes**

Ce sont des ensembles de tubes alimentés par l'eau provenant des deux ballons et qui sont soumis à l'action des brûleurs dans la chambre de combustion.

➤ **Ventilateurs de combustion**

La chaudière dispose de deux ventilateurs dits de tirage forcé, pour fournir l'air de combustion nécessaire à l'écoulement des brûleurs. Ils aspirent l'air de l'extérieur vers la chambre pour permettre la combustion ou pour le balayage lors de l'allumage de la

chaudière.





➤ **Vannes de sécurité**


La chaudière possède des vannes de sécurité afin de protéger une partie de pression. Il existe plusieurs types de vanne de sécurité comme la vanne régulatrice et la vanne tout ou rien.[4]

2.3.2.2.2 Les instruments

Il existe plusieurs types d'instruments. Nous allons citer les plus utilisés dans la réalisation de notre projet. Le rôle de ces instruments est de contrôler les quatre paramètres ou (grandeur) physiques pression, niveau, débit et la température.

Tableau 2 : types d'instruments et leurs rôles

| Types d'instrument | Description | Illustration |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Indicateur de pression | <p>Les transmetteurs de pression peuvent être électroniques intelligents, analogiques ou basse tension. Ces capteurs existent dans une gamme très variée de configurations destinée aux applications de mesure de la pression différentielle. Ils ont des cartes de circuit imprimé enfichables à base de circuits intégrés.</p> |  |
| Indicateur de niveau | <p>Tous les systèmes pétrochimiques utilisent ces capteurs pour mesurer le niveau de fluide. Il convient de connaître la nature de fluide, leur propriété physique et chimique, la distance et les conditions de sécurité.</p> |  |
| Indicateur de débit | <p>Sont des capteurs utilisés pour mesurer le débit. Ils utilisent des techniques qui peuvent être mise en oeuvre. On trouve essentiellement ce type dans les organes déprimogènes : diaphragme, venturi, tuyère.</p> |  |
| 3 Indicateur de température | <p>Les thermocouples : Dans un circuit fermé constitué de deux conducteurs de nature différente, il circule un courant lorsqu'on maintient entre les deux jonctions une différence de température</p> |  |

| | | |
|--|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| | <p>Les sondes : La sonde est essentiellement utilisée pour les mesures continues et détections de niveau dans les liquides [4].</p> |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|

4 Séquence de mise en service chaudière ¹

4.1 Préventilation

On entend par préventilation (ou balayage) le passage ininterrompu à travers la chambre de combustion d'un volume d'air égal au moins à 4 fois le volume du foyer et des circuits de fumées. La séquence préventilation permet de s'assurer que le circuit fumé de la ligne de procédé est exempt d'imbrûlés gazeux au moment où l'on va allumer le premier brûleur. On peut effectuer la préventilation si les conditions correspondantes sont vérifiées (cf. chapitre 1: sécurisés). L'action de l'opérateur sur le B.P "Preventilation" provoque

| | | |
|-----------------|---------------------------------------------------------------------|-----------------------------------------------|
| T0 | Ventilateur(s) d'air de combustion en service | (B.GB102 ou B.GB103) |
| T0+1 | Ouverture des volets d'air combustion. | (FCV316-1 et FCV316-1) (ordre préventilation) |
| | Ouverture des ventelles des brûleurs (1à4). | (FCV 357 1 à 4) |
| | Ouverture de la vanne générale alimentation gaz principale. | EMV |
| 333 | | |
| | Ouverture de la vanne générale alimentation gaz allumeur. | EMV |
| 330 | | |
| T0-6 et FSH-TI | Contrôle du débit air max. (80%) | FSH316 |
| | Lancement de la temporisation "préventilation" (Tp) | |
| | Maintien en ouverture des ventelles des brûleurs | |
| | Maintien en ouverture des volets d'air combustion | |
| T1-TP-FCV316-1) | Fermeture des volets d'air combustion | (FCV316-1 et |
| | Lancement de la temporisation "délais autorisation d'allumage (Taa) | |
| | Maintien en ouverture des ventelles des brûleurs | |
| T1+TP et FSL | Contrôle du retour au débit air min. | (20%) |
| | Autorisation d'allumage des brûleurs | FSL316 |

¹ Manuel de fonctionnement du complexe CP2K

Remarque : Après 2 tentatives d'allumage infructueuses le temps de préventilation est doublé et impose un arrêt immédiat et des mesures correctrices avec appel à du personnel compétent.

- Le temps de "préventilation" (T_p) devra être réglé à la mise en service afin de respecter la condition citée ci-dessus. Toutefois la temporisation de préventilation est pré réglé à 10 minutes.
- Le temps "délais autorisation d'allumage" (T_{aa}) est de 3 minutes (délai accordé à l'opérateur pour allumer le premier brûleur).
- Si le débit d'air de combustion n'atteint pas le seuil 80% ou si ce seuil n'est pas atteint dans un temps déterminé (fixer à la mise en service).
- Si une sécurité principale apparaît.
- Si fin de la temporisation "délais autorisation d'allumage" avant le lancement de la mise en service d'un brûleur.
- La séquence de préventilation s'arrête
- Provoque l'activation de l'avertisseur sonore et l'apparition du message associé au défaut ayant provoqué l'arrêt il faut relancer la séquence de préventilation après avoir éliminé le défaut.

3.2 BRULEURS GAZ

3.2.1 ALLUMAGE DU PREMIER BRULEUR - (Ex. brûleur 1)

L'opérateur peut indifféremment choisir d'allumer fun de 4 brûleurs comme premier brûleur, toutefois il est conseillé de commencer par l'un des brûleurs du bas. L'opérateur dispose sur la face avant de l'armoire locale, d'un bouton poussoir "mise en service" et d'un bouton poussoir d'arrêt" par brûleur.

Not l'allumage du premier brûleur chaudière froide ne peut être réalisé que localement (armoire locale)

| | | |
|------------|-----------------------------------------------------|---------------------|
| T0 | Ouverture de la vanne purge | EMV350 |
| | Maintien en ouverture des ventelles des 4 brûleurs | FCV357-1à4 |
| T0+0.1s=T1 | (Valeuil à régler à la mise en service) | |
| | Mise en service du transformateur d'allumage | BZ332-1 |
| | Ouverture des vannes gaz d'allumage brûleur | EMV354-1 et EMV356- |
| 1 | Maintien en ouverture des ventelles des 4 brûleurs. | FCV357-1 à4 |
| | Maintien en ouverture vanne de purge | EMV350 |
| T1+4s | Arrêt du transformateur d'allumage | BZ332-1 |
| | Prise en compte de la flamme torche | BSL332-1 |
| | Maintien en ouverture des vannes gaz d'allumage | EMV354-1 et EMV356- |
| 1 | Maintien en ouverture des ventelles des 4 brûleurs | FCV357-1 à4 |
| | Maintien en ouverture vanne purge | EMV350 |

| | | |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| T1+9s 1 | Ouverture vanne amont gaz. Maintien en ouverture des ventelles des 4 brûleurs Maintien en ouverture des vannes gaz d'allumage | EMV351-1 FCV357-1à4 EMV354-1 et EMV356- |
| T1+13s 1 | Maintien en ouverture vanne amont gaz. Fermeture de la vanne d'évent. Ouverture vanne aval gaz. Fermeture de la vanne de purge. Maintien en ouverture des ventelles des 4 brûleurs. Maintien en ouverture des vannes gaz d'allumage. | EMV351-1 EMV352-1 EMV353-1 EMV330 FCV357-1à4 EMV354-1 et EMV356- |
| T1+17s 1 | Maintien en ouverture des vannes gaz d'allumage. Prise en compte de la détection de flamme principale. Maintien en ouverture de la vanne aval gaz. Maintien en fermeture de la vanne d'évent. Maintien en ouverture de la vanne amont gaz. Maintien en fermeture de la vanne de purge. Maintien en ouverture des ventelles des 4 brûleurs. | EMV354-1 et EMV356- FFA331-1 EMV351-1 EMV352-1 EMV353-1 EMV330 FCV357-1à4 |
| T1+19s | Allumage terminé. Lancement temporisation de stabilisation allumage. | |

Dès cet instant le brûleur est régulé en pression afin de faciliter l'allumage éventuel du 2 brûleur.
L'opérateur devra :

- Soit allumer le 2 brûleur
- Soit passer le commutateur "Allumage/Marche normale" en position "marche normale".

Lors du basculement du commutateur

- Les ventelles des brûleurs à l'arrêt se ferment
- La régulation passe automatiquement en régulation de débit si la temporisation (5 secondes) de stabilisation allumage terminée (temporisation à régler au plus court à la mise en service), la puissance de la chaudière est limitée à 25%

Signalisation

- Pendant la mise en service du premier brûleur le voyant "autorisation d'allumage" s'éteint.
- Pendant la mise en service le voyant du brûleur concerné "brûleur xxx" clignote.
- Dès que le brûleur est en service le voyant du brûleur concerné "brûleur xxx" passe en feux fixe.

5 Le schéma de principe de fonctionnement de l'unité Eau

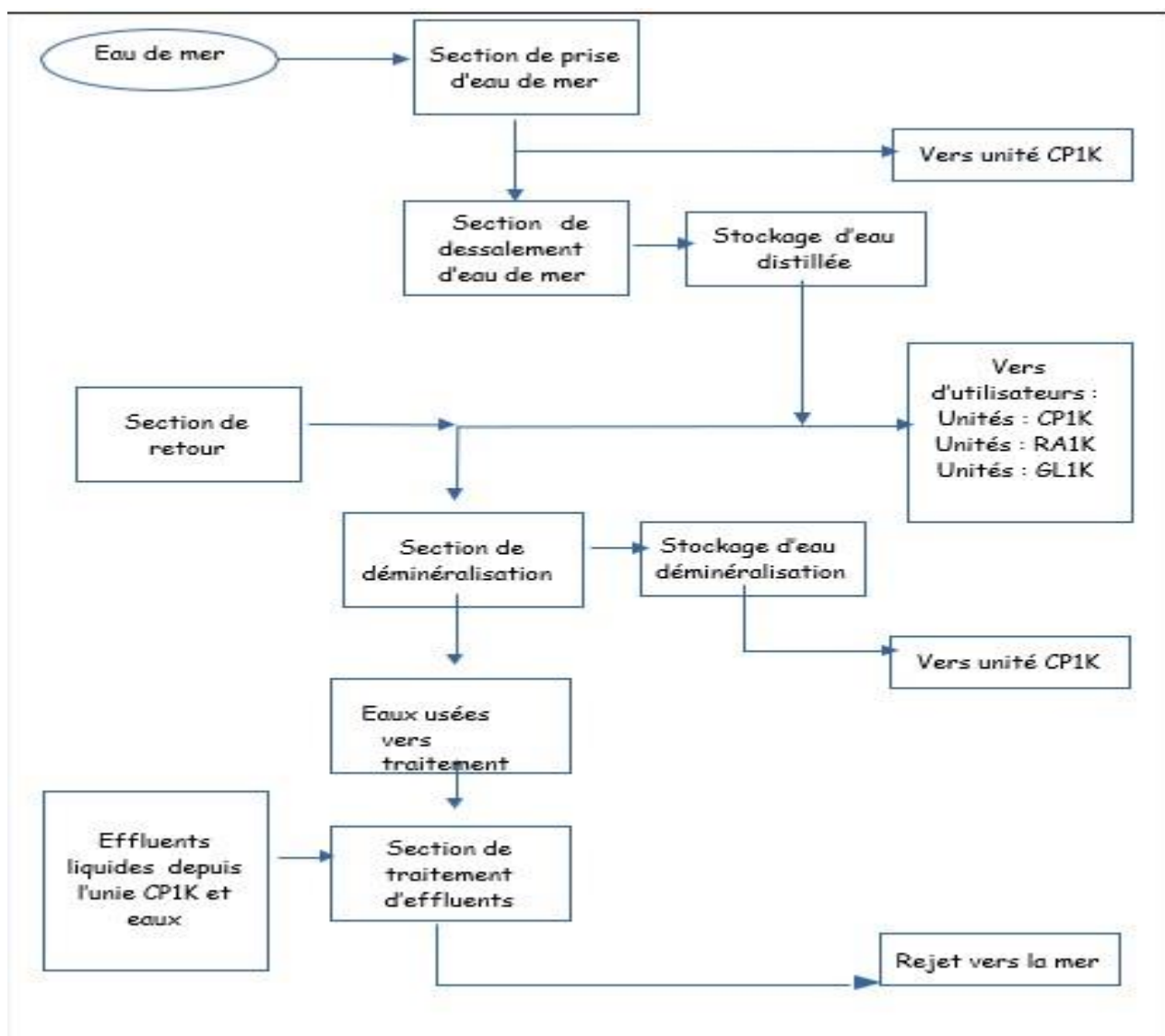


Figure3.19 : Principe de fonctionnement de l'unité d'Eau (Ferrag Sihèm & Korzet Yassine, 2013).

6 Conclusion

Nous avons abordé dans ce chapitre les concepts de base de l'outil AML. Il est basé sur le format CAEX pour une description topologique des éléments d'un système ainsi que le format PLCopenXML pour la description logique et comportementale décrivant le fonctionnement dynamique d'une installation industrielle (la chaudière). Nous avons également montré les principaux composants de cette dernière ainsi que son fonctionnement.

Chapitre 4

Conception

1 Introduction

Dans nos jours les installations industrielles connaissent une complexité croissante ainsi qu'une grande diversité d'utilisateurs qui participent au processus de la construction d'une usine, ce qui rend l'échange et la validation des modèles d'ingénierie plus difficile. Même s'ils sont hétérogènes, ces derniers doivent avoir une collaboration afin de concevoir et créer un système de production complexe. L'échange de données entre les outils d'ingénierie a été identifié comme un goulot d'étranglement majeur dans le flux de travail d'ingénierie.

AutomationML vise à combler cette lacune en soutenant l'échange de divers aspects de l'ingénierie de l'automatisation et sa capacité à gérer plusieurs sémantiques dans un paysage d'outils d'ingénierie hétérogène.

Dans notre système, nous construisons tout d'abord une ontologie de domaine basée sur la norme AutomationML et nous décrivons comment les données représentées dans le fichier CAEX et le fichier PLCopen XML pour le format d'entrée GANTT.

Les technologies du Web sémantique offrent des mécanismes pour interroger et raisonner sur des modèles de domaine exprimés sous forme d'ontologies. En particulier, le langage OWL est doté d'une sémantique formelle, qui explique une formulation précise des restrictions sur les modèles de domaine qui peuvent être validées en utilisant les outils de raisonnement et d'interrogation disponibles.

1.1 Objectif du travail

Dans notre travail, les règles de transformation (mapping) vers les spécificités PLCopen XML sont définies pour le format d'entrée GANTT. Nous avons d'abord défini les règles de transformation vers la couche intermédiaire, ce qui améliore l'extensibilité d'AML pour les nouveaux formats d'entrée.

Pour résumer les principales catégories et leur donner un contexte formel, la couche intermédiaire de modélisation (IML) est définie pour clarifier et simplifier le processus de mise en correspondance des différents types de modèles avec AutomationML. Néanmoins, elle ne constitue pas un nouveau langage de modélisation. Pour l'importation d'une description logique à partir de PLCopen XML, nous allons utiliser AML pour

transformer les formats logiques de GANTT.

- ✓ Description de l'architecture globale de notre système.
- ✓ Modélisation statique et dynamique de la chaudière industrielle avec l'outil AutomationML.
- ✓ Construction d'une ontologie de domaine « Boiler.OWL » inspirés de la hiérarchie AML des concepts et des relations.
- ✓ La population et l'instanciation de l'ontologie développé par des instances issues des documents CAEX et le fichier PLCopen XML (Mapping CAEX-OWL et PLCopen-SWRL).
- ✓ Représentation sémantique des activités des diagrammes de GANTT avec le langage SWRL.
- ✓ Implémentation des interfaces conviviales permettant de parcourir un fichier XML. Ces interfaces permettent de convertir les documents CAEX et PLCopen XML.

1.2 Travaux existants

Lisa Abele, Martin Kleinsteuber, Thorbjørn Hansen. Ont décrit dans (Lisa Abele et al., 2011) une usine au moyen de modèles et de règles basées sur la connaissance pour mettre en oeuvre un système de surveillance générique qui peut être défini avec un effort initial réduit et qui peut être adapté rapidement aux modifications. Un avantage de cette approche est que la consommation d'énergie et de ressources de chaque composant dans une usine peut être associée à une séquence d'opérations et que les effets sur l'ensemble du système deviennent visibles. Un autre avantage des systèmes basés sur la connaissance combinée aux règles est qu'ils offrent des solutions indépendantes de l'application et de la flexibilité. Ils ont décrit plusieurs approches, telles que les systèmes de surveillance existants, les moteurs de règles et les outils de modélisation. En outre, ils ont aussi décrit un exemple représentatif que nous utiliserons dans notre travail ultérieur pour évaluer quels outils sont appropriés pour un système de surveillance des ressources.

Le travail présenté dans (Abele et al., 2011) décrit l'installation industrielle au moyen de modèles et de règles basé sur la connaissance pour mettre en place un système de surveillance générique. Un avantage de cette approche est l'utilisation des technologies pertinentes telles que les systèmes de surveillance, les moteurs de règles et les outils de modélisation AML, mais les auteurs n'ont pas présenté un algorithme détaillé pour le mapping AML ontologie. Ils ont indiqué brièvement une simple correspondance entre les deux formalismes. Dans notre travail, nous avons proposé des algorithmes détaillés pour assurer un mapping complet.

Stefan Zander et al. Ont proposé dans (Zander & Hua, 2016) une nouvelle approche basée sur l'ontologie pour l'accomplissement des spécifications techniques des

composants d'un system cyber - physique en utilisant la classification et le raisonnement ontologique. Ils ont proposé une approche de transformation de données issues d'AML en un graphe RDF. Les auteurs ont montré comment des informations complémentaires d'une fonctionnalité d'un composant, son environnement de fonctionnement et son but peuvent être déduites par une classification automatique. La base de connaissance générée dans ce travail est constituée juste des graphes RDF. Dans notre travail, nous avons généré une ontologie riche basé sur le langage OWL ce qui augmente le niveau d'expressivité.

2 Architecture du système de création semi-automatique d'ontologie à partir des documents CAEX et PLCopen

L'architecture du système proposée est illustrée dans la figure 4.1. Elle vise à répondre aux différents objectifs identifiés précédemment. Elle se base sur une approche de peuplement automatique d'une ontologie à partir des documents CAEX et PLCopen générés par l'outil AML.

Le processus supporté par cette architecture comprend trois phases de traitement : la première phase est la phase de développement du domaine de chaudière, la deuxième phase est la phase de description AML Statique des composants matériels et la troisième phase est la phase de description du comportement dynamique.

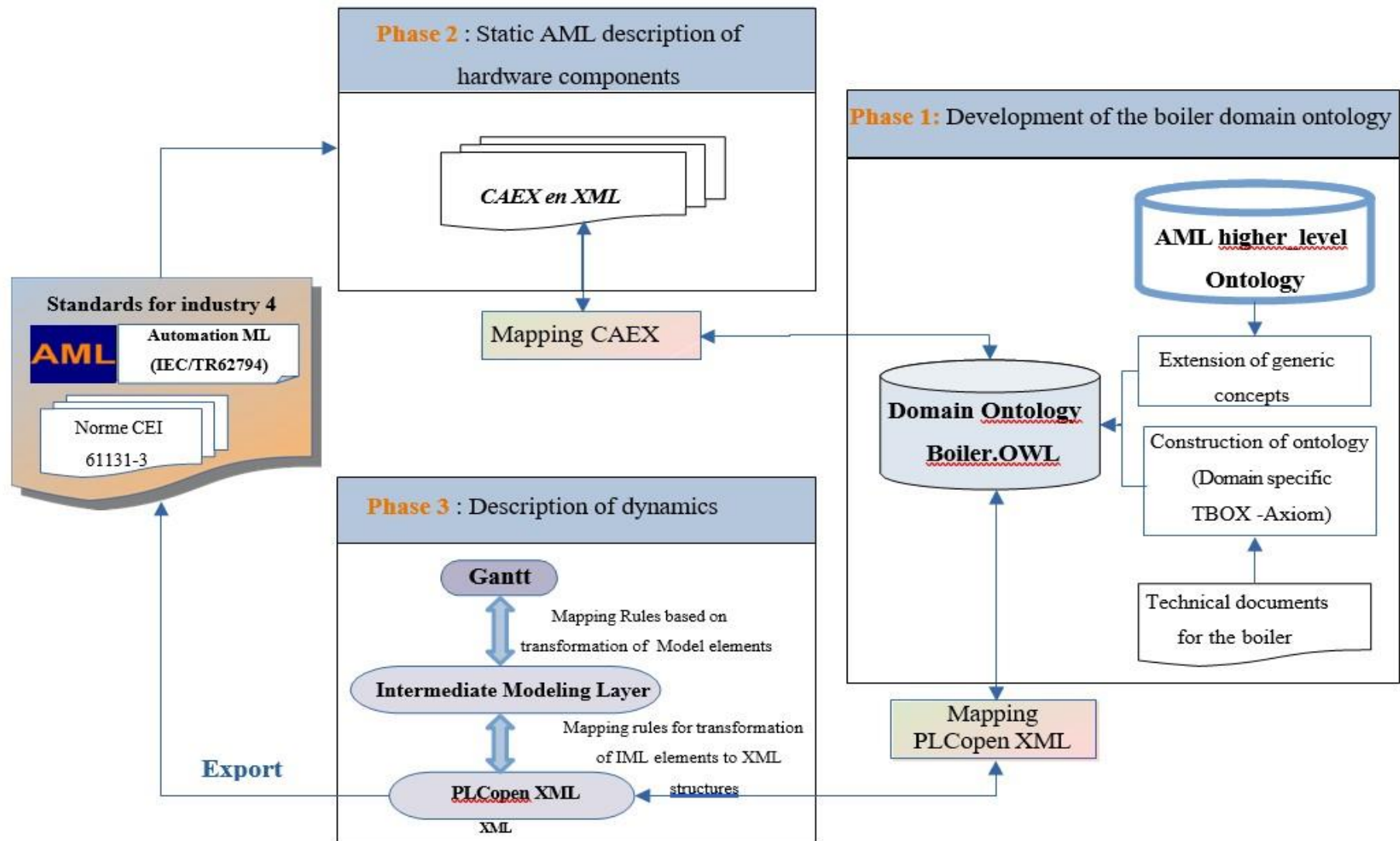


Figure 4.1: Architecture générale.

2.1 Phase de développement de l'ontologie de domaine « Boiler.OWL »

En utilisant une ontologie de niveau supérieur, nous pouvons créer une ontologie de domaine plus organisée et structurée, tout en évitant toute redéfinition des termes. De plus, l'intégration d'autres ontologies, dérivées ou connectées au même domaine, peut être facilement réalisée sans aucune complication due aux termes et relations partagés. L'objectif de cette étape est d'organiser la connaissance du domaine dans un modèle conceptuel. Cette phase est composée des tâches suivantes :

2.1.1 Construction de la hiérarchie des concepts et les relations binaires

Afin de conceptualiser notre ontologie de domaine, nous nous sommes inspirés de la hiérarchie AML des concepts et des relations. Nous avons réutilisé un fragment d'AML dans Protégé, illustré à la figure 4.2.

2.1.1.1 Construction du dictionnaire des concepts

Le dictionnaire des concepts comprend les concepts d'un domaine spécifique, y compris leurs descriptions et les super-classes.

Tableau 3: Dictionnaire des concepts.

| Concept | Description | Parent |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| aml:CAEXFile | ✓ Cette classe représente le document AML et constitue l'élément central de l'ontologie. | Thing |
| aml:RoleClassLib | ✓ La RoleClassLibrary contient une collection de fonctionnalités possibles qui peuvent être fournies par l'équipement de l'usine. | Identifiable |
| aml:RoleClass | <p>✓ aml:SystemUnitClass représente le concept de classe d'unité de système AML et définit une réalisation technique d'un objet physique ou logique propre à un fournisseur.</p> <p>✓ Les instances de aml:SystemUnitClass peuvent contenir et faire partie d'autres classes d'unités de système.</p> | Identifiable |
| aml:SystemUnitClassLib | ✓ La SystemUnitClassLibrary comprend des collections d'objets d'équipement de solution spécifiques au fournisseur. | Identifiable |
| aml:SystemUnitClass | ✓ aml:SystemUnitClass représente le concept de classe d'unité de système AML et définit une réalisation technique d'un objet physique ou logique propre à un fournisseur. | Identifiable |

| | | |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| aml:InstanceHierarchy | <p>✓ La hiérarchie des instances décrit la topologie de l'usine, y compris l'équipement concret d'un projet réel, c'est-à-dire les données d'instance.</p> <p>✓ La hiérarchie d'instance contient toutes les données, y compris les propriétés, les interfaces, les classes de rôle, les relations et les références.</p> | Identifiable |
| aml:InternalElement | <p>✓ aml:InternalElement modélise le concept d'éléments internes dans AML et définit l'équipement concret d'un projet réel, c'est-à-dire les dispositifs utilisés dans une centrale à béton.</p> <p>✓ La classe aml:InternalElement peut contenir d'autres classes aml:InternalElement et en faire partie.</p> | Identifiable |
| aml:InterfaceClassLib | <p>✓ Cette classe définit toutes les interfaces nécessaires pour décrire le modèle de l'usine.</p> | Identifiable |
| aml:InterfaceClass | <p>✓ aml:InterfaceClass représente le concept de classe d'interface dans le langage AML et est généralement utilisé pour spécifier soit a) des relations entre les différents éléments de la topologie, soit b) des références à diverses sources d'informations externes.</p> | Identifiable |
| Instrument | <p>✓ Décrire les capteurs de mesures comme les indicateurs de température, de pression....</p> | RoleClass |

| | | |
|--------------------------------------|----------------------------------------------------------------------------------|-------------------|
| Equipement | ✓ Décrire les équipements de la chaudière comme l'économiseur, les brûleurs,.... | RoleClass |
| Boiler | ✓ Le concept qui décrit la chaudière industrielle | SystemUnitClass |
| Water_Drum | ✓ Un tambour pour collecter l'eau | Boiler |
| Water_Pump | ✓ Pompe d'eau | Boiler |
| Valves | ✓ Les vannes de contrôle | Boiler |
| FCV | ✓ Vanne de contrôle du débit | Valves |
| HCV | ✓ Vanne de contrôle du niveau | Valves |
| EMV | ✓ Vanne de contrôle de la pression | Valves |
| TCV | ✓ Vanne de contrôle de la température | Valves |
| Burner | ✓ Les brûleurs de la chambre de combustion | Boiler |
| PIC | ✓ Contrôleur d'indicateur de pression | Boiler |
| TIC | ✓ Contrôleur d'indicateur de température | Boiler |
| FIC | ✓ Contrôleur d'indicateur de débit | Boiler |
| LIC | ✓ Contrôleur d'indicateur de niveau | Boiler |
| Interface_open/close | ✓ les interfaces ouverture et fermeture des vannes | InterfaceClass |
| Interface_Working/Stopped | ✓ les interfaces de travail et arrêté des moteurs | InterfaceClass |
| Interface_Absence/ Presence_Flame | ✓ Les interfaces d'absence et de présence de la flamme du brûleur | InterfaceClass |
| Interface_open | ✓ L'interface ouverture des vannes | ExternalInterface |
| Interface_close | ✓ l'interface fermeture des vannes | ExternalInterface |
| Interface_Working | ✓ l'interface de travail des moteurs | ExternalInterface |

| | | |
|--------------------------|-------------------------------------------------|-------------------|
| Interface_Stopped | ✓ la interfaces arrêté des moteurs | ExternalInterface |
| Interface_Presence_Flame | ✓ L'interface présence de la flamme du brûleur | ExternalInterface |
| Interface_Absence_Flame | ✓ L'interface d'absence de la flamme du brûleur | ExternalInterface |

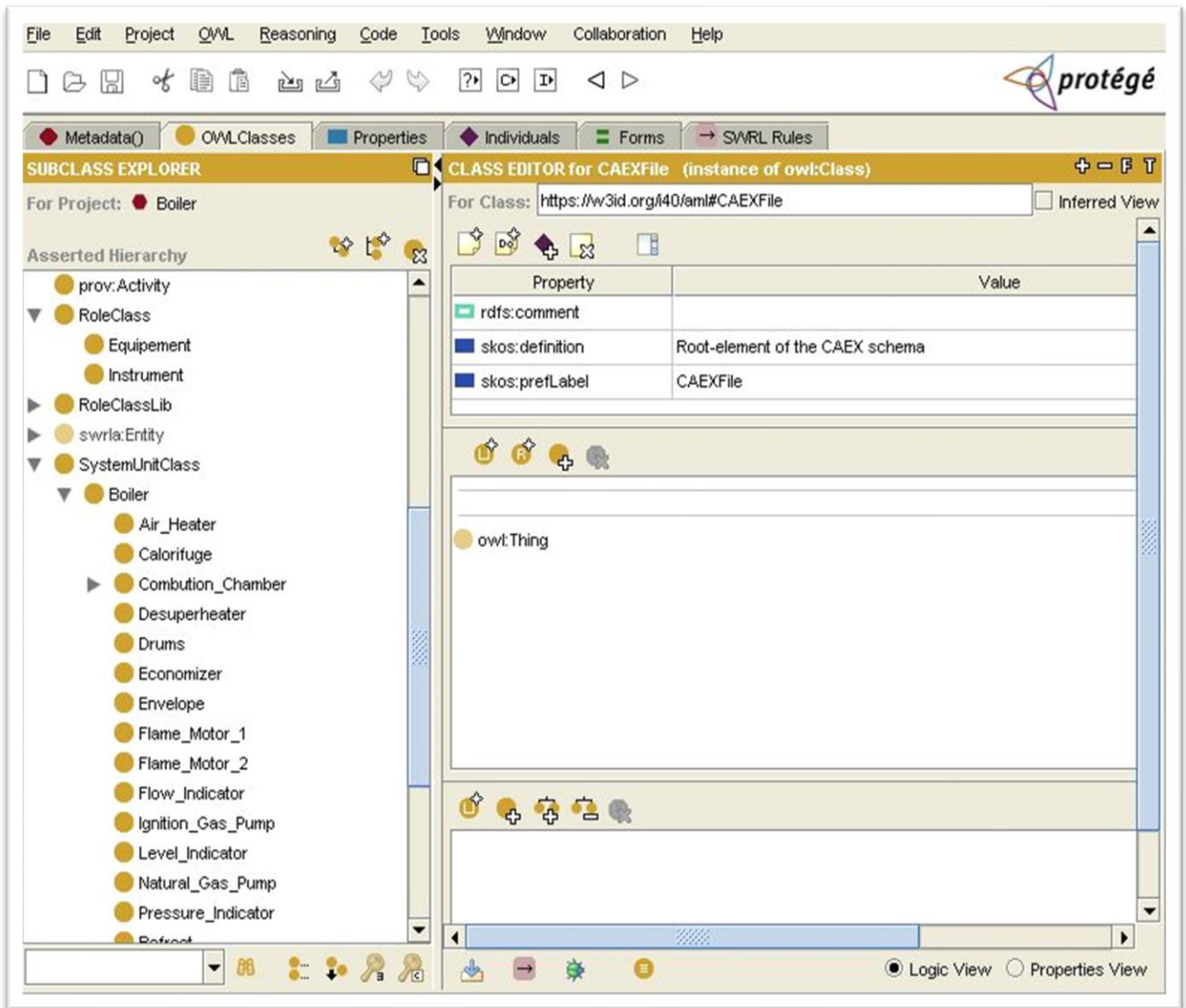


Figure 4.3: Hiérarchie des classes.

2.1.1.2 Construction de la table des relations binaires

Pour chaque relation, la table des relations binaires définit le nom de la relation ainsi que les concepts sources et cibles.

Tableau 4 : Table des relations binaires.

| Relation | Description | Concept(s) source(s) | Concept(s)cible(s) |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|-------------------------|
| aml:contains | ✓ La propriété aml:contains a la même sémantique pour les quatre constructions. | aml :CAEXFile | aml :RoleClassLib |
| | | | aml :SystemUnitClassLib |
| | | | aml :InstanceHierarchy |
| | | | aml :InterfaceClassLib |
| aml:hasRoleClass | ✓ La classes de rôles comporte des classes de rôles La classe aml:RoleClass est utilisée pour attribuer une sémantique fonctionnelle générique aux instances des classes aml:InternalElement et aml:SystemUnitClass. | aml :RoleClassLib | aml :RoleClass |
| aml:hasSupportedRole Class | ✓ L'objectif est de décrire les capacités fonctionnelles des éléments d'équipement. ✓ Pour ce faire, les Propriétés aml:hasSupportedRoleClass sont utilisées pour décrire les capacités fonctionnelles des éléments d'équipement. | aml :RoleClass | aml :SystemUnitClassLib |

| | | | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|-----------------------|
| aml:hasSystemUnitClasses | ✓ aml:hasSystemUnitClasses Lib sont définies comme des sous-propriétés de aml:contains. Cette approche permet d'utiliser un vocabulaire moins complexe pour les applications où seule l'interrogation est importante. | aml :SystemUnitClass Lib | aml :SystemUnitClass |
| aml:hasInternalElement | ✓ la hiérarchie d'instance a un élément interne | aml :InstanceHierarchy | aml:InternalElement |
| aml:hasExternalInterface | ✓ l'élément interne à une interface externe | aml:InternalElement | aml:ExternalInterface |
| aml:hasInterfaceClassLib | ✓ la classe d'interface Library a des classes d'interface. | aml:InterfaceClassLib | aml:InterfaceClass |

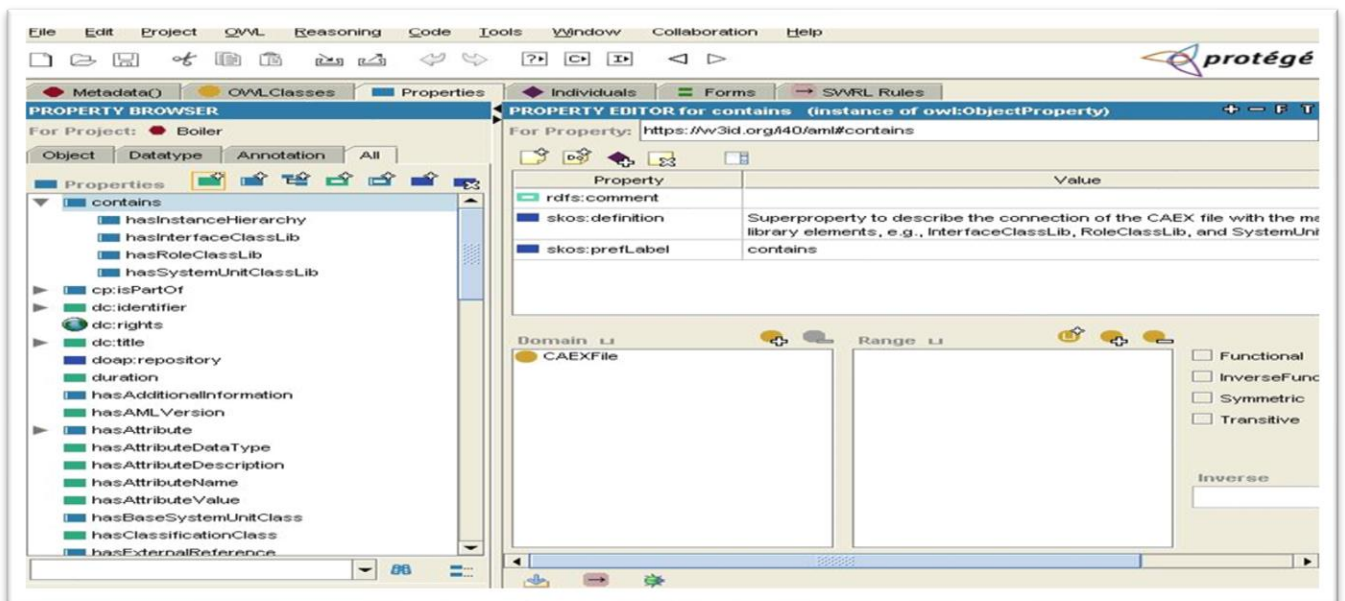


Figure 4.4: Définition des propriétés

2.1.1.3 Construction de la table des instances des concepts

Toutes les instances sont décrites dans la table des instances des concepts.

Tableau 5: Table des instances.

| Concept | Nom de l'instance |
|------------|-------------------|
| Water_Drum | Water_Drum_1 |
| Water_Pump | Water_Pump_1 |
| EMV | EMV330 |
| | EMV323 |
| | EMV333 |
| | EMV350 |
| | EMV351_1 |
| | EMV351_2 |
| | EMV351_3 |
| | EMV351_4 |
| TCV | TCV319 |
| | TV317 |
| FCV | FCV311 |
| | FCV316_1 |
| | FCV316_2 |
| HCV | HCV317 |
| | HCV371 |
| | HCV316 |
| Burner | Br_1 |
| | Br_2 |
| | Br_3 |
| | Br_4 |

| | |
|----------------------------------|--------------------------|
| PIC | PIC311 |
| | PIC329 |
| | PT312 |
| | PT322 |
| | PT320_1 |
| | PT320_2 |
| TIC | TT321_1 |
| | TT321_2 |
| | TT321_3 |
| | TT321_4 |
| | TT321_5 |
| FIC | FIC311 |
| | FIC316 |
| | FT318 |
| | FT319 |
| | FT311 |
| LIC | LIC320 |
| | LT320_1 |
| | LT320_2 |
| Interface_open/close | Interface_open |
| | Interface_close |
| Interface_Working/Stopped | Interface_Working |
| | Interface_Stopped |
| Interface_Absence/Presence_Flame | Interface_Absence_Flame |
| | Interface_Presence_Flame |

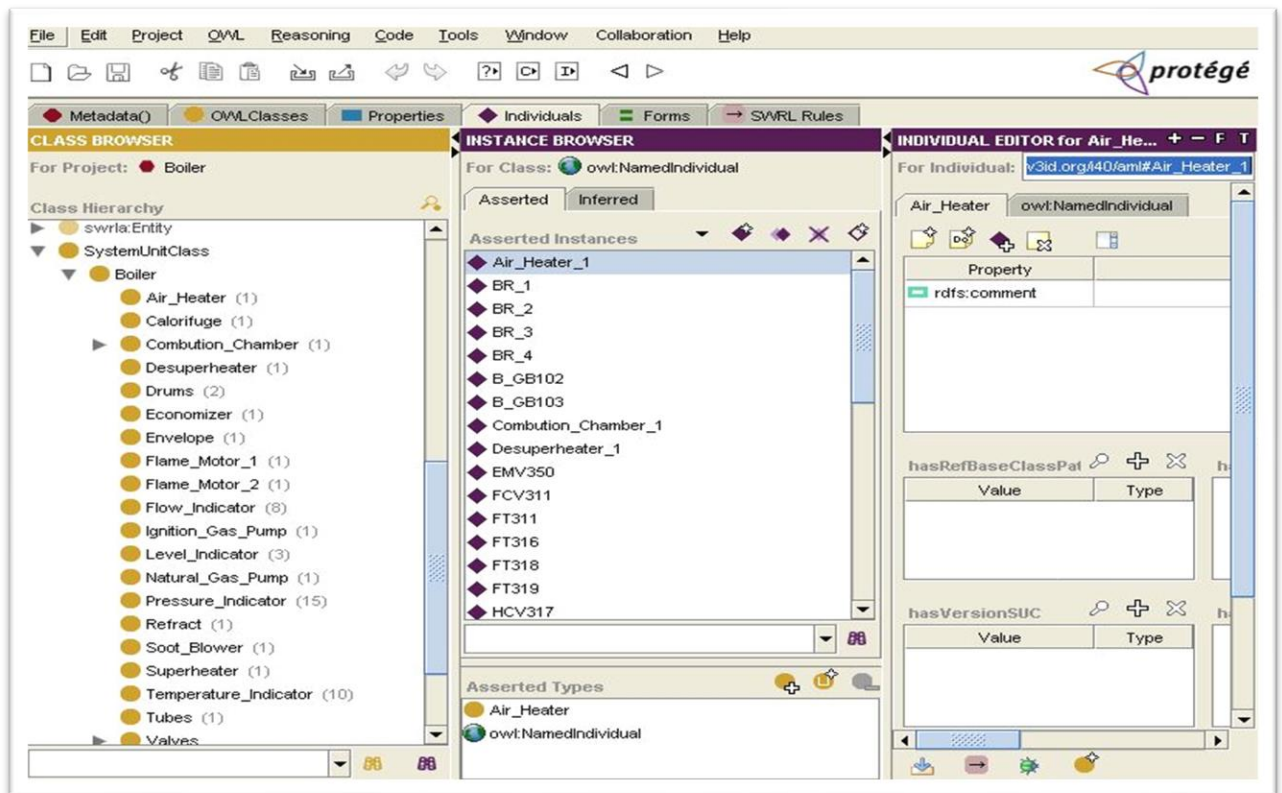


Figure 4.5: Définition des instances.

2.1.1.4 Représentation sémantique des activités d diagramme de GANTT avec le langage SWRL

L'ontologie développée dans ce travail modélise la topologie et la partie dynamique de la chaudière industrielle. L'ontologie est formulée dans le langage de représentation de connaissance OWL, cette implémentation est réalisée par l'outil Protégé et validée par le raisonneur faCT++. Le langage OWL pose certains problèmes à cause de quelques limitations en particulier dans la définition des propriétés (Bouaroudj Samiya & Boufaïda, 2009) Pour résoudre ce problème, SWRL, une extension à OWL, a été développée pour rendre le langage encore plus expressif pour la description de ces propriétés.

L'enrichissement de l'ontologie par des règles consistant à injecter un système à base de règles dans la base de connaissance permet de :

- Ajouter via les règles SWRL des axiomes sur les propriétés décrites dans le fragment d'ontologie

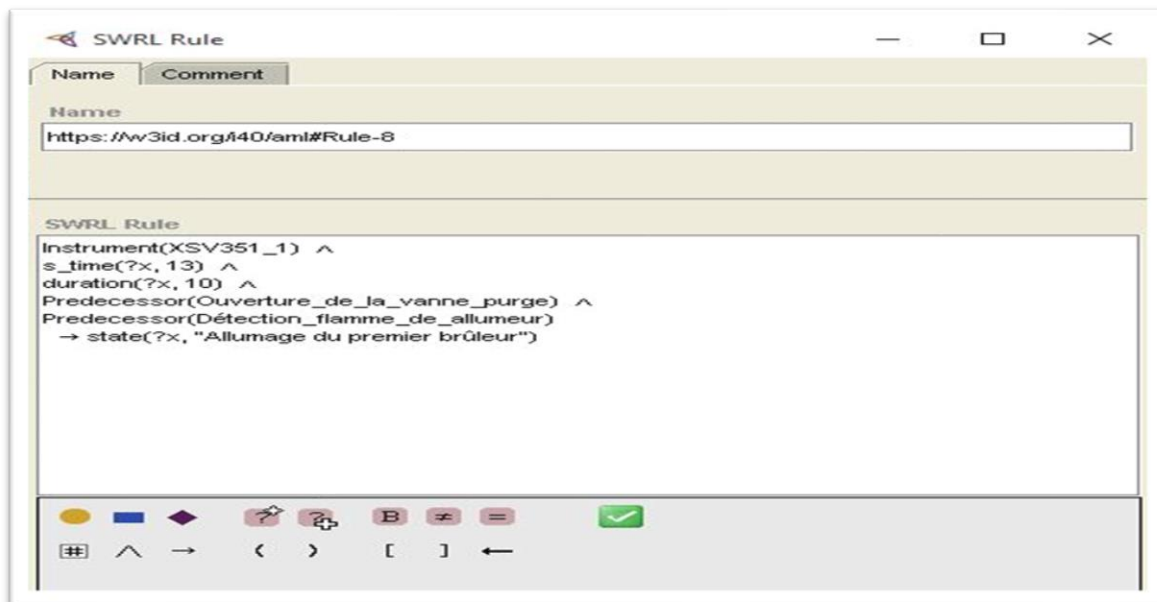
- $\text{Instrument}(\text{FCV316_1}) \cap \text{s_time} (? x, 1) \cap \text{duration} (? x, 1) \rightarrow \text{state} (? x, \text{"prevention"})$
- $\text{Instrument}(\text{FSL316}) \cap \text{s_time} (? x, 20) \cap \text{duration} (? x, 2) \text{Predecessor} (\text{"Lancement_de_la_temporisation"}) \rightarrow \text{state} (? x, \text{"prevention"})$

- $\text{Instrument}(\text{XSV351_1}) \cap \text{s_time} (? x, 13) \cap \text{duration} (? x, 10) \cap \text{Predecessor} ("Ouvverture_de_la_vanne_purge") \cap \text{Predecessor} ("D\u00e9tection_flamme_de_allumeur") \rightarrow \text{state} (? x, "Allumage\ du\ premier\ br\u00f4leur")$
- $\text{Instrument}(\text{BSL322_1}) \cap \text{s_time} (? x, 5) \cap \text{duration} (? x, 4) \cap \text{Predecessor} ("Mise_en_service_du_transformateur_allumage") \rightarrow \text{state} (? x, "Allumage\ du\ premier\ br\u00f4leur")$
- $\text{Instrument}(\text{XSV353_1}) \cap \text{s_time} (? x, 13) \cap \text{duration} (? x, 4) \cap \text{Predecessor} ("Ouvverture_de_la_vanne_purge") \cap \text{Predecessor} ("D\u00e9tection_flamme_de_allumeur") \cap \text{Predecessor} ("Ouvverture_de_la_vanne_amont_gaz") \rightarrow \text{state} (? x, "Allumage\ du\ premier\ br\u00f4leur")$

➤ Editer les r\u00e8gles avec l'\u00e9diteur SWRL

L'\u00e9diteur SWRL (Bouaroudj Samiya & Boufa\u00efda, 2009) est un plugin de Prot\u00e9g\u00e9 OWL permettant l'\u00e9dition interactive des r\u00e8gles SWRL. Les utilisateurs peuvent cr\u00e9er, modifier, lire et \u00e9crire les r\u00e8gles SWRL.

L'\u00e9diteur SWRL permet aux utilisateurs de saisir des r\u00e8gles compl\u00e8tement sous forme de texte. Toutefois, il permet \u00e9galement aux utilisateurs de s\u00e9lectionner des classes OWL, propri\u00e9t\u00e9 et individus de la base de connaissances (Ontologie) charg\u00e9e et les ins\u00e9rer dans la r\u00e8gle en cours de cr\u00e9ation



d'enregistrer des règles incomplètes ou erronées. Le biais de l'onglet de règles SWRL (SWRL Tab). Cet onglet est présenté dans la figure 4.7. Il affiche toutes les règles SWRL de l'ontologie sous forme de table

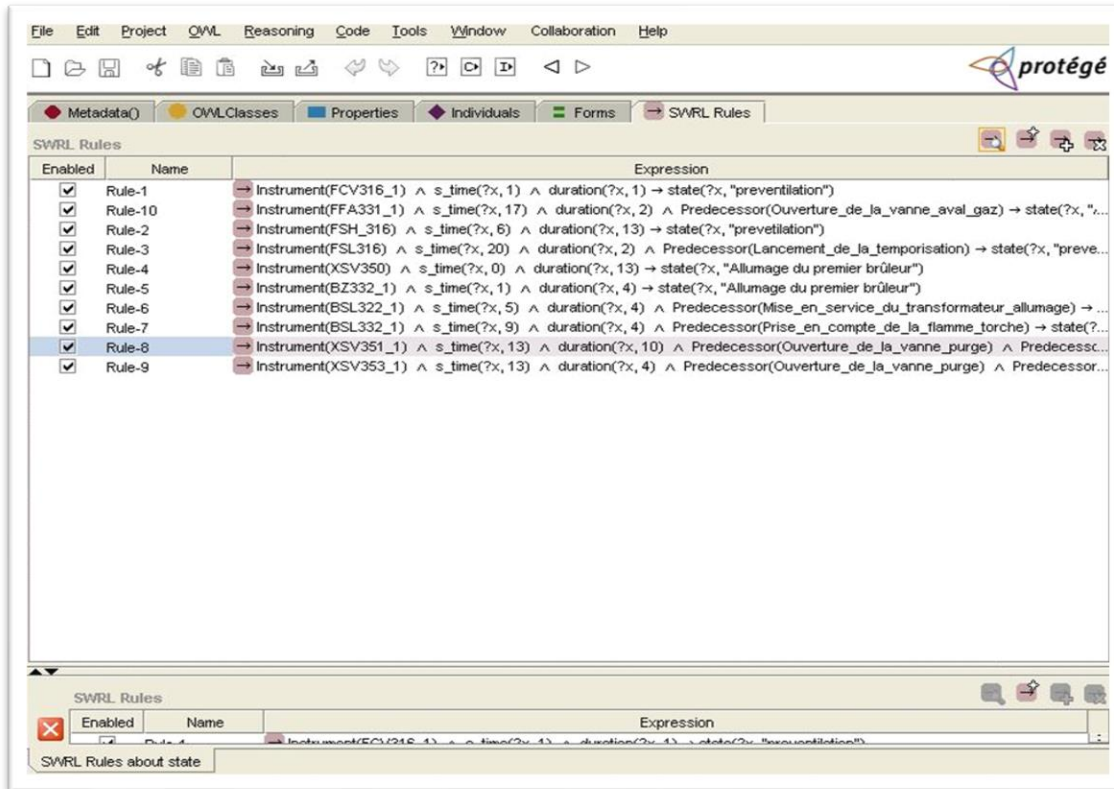


Figure4.7 : les règles SWRL de l'ontologie.

2.1.2 Phase de description AML Statique des composants matériels

2.1.2.1 Création d'un document AML de la chaudière

Le document AML présente une description des composants matériels de la chaudière par l'éditeur AutomationML. L'architecture d'AutomationML contient les blocs suivants :

- **RoleClassLib**

Les concepts spécifiques au domaine sont modélisés sous forme de RoleClasses et organisés comme taxonomie dans la bibliothèque RoleClass. On peut définir de nouvelles classes de rôle selon les besoins de notre système. La bibliothèque de rôle est nommée my RoleClassLib. Les classes de rôle Equipement et Instrument sont héritées

de la bibliothèque my RoleClassLib. Chaque classe de rôle doit avoir un nom unique dans l'arborescence des rôles d'une bibliothèque. Il peut être référencé de manière unique par ce chemin hiérarchique. La figure 4.8 présente la définition de notre bibliothèque avec une description XML.

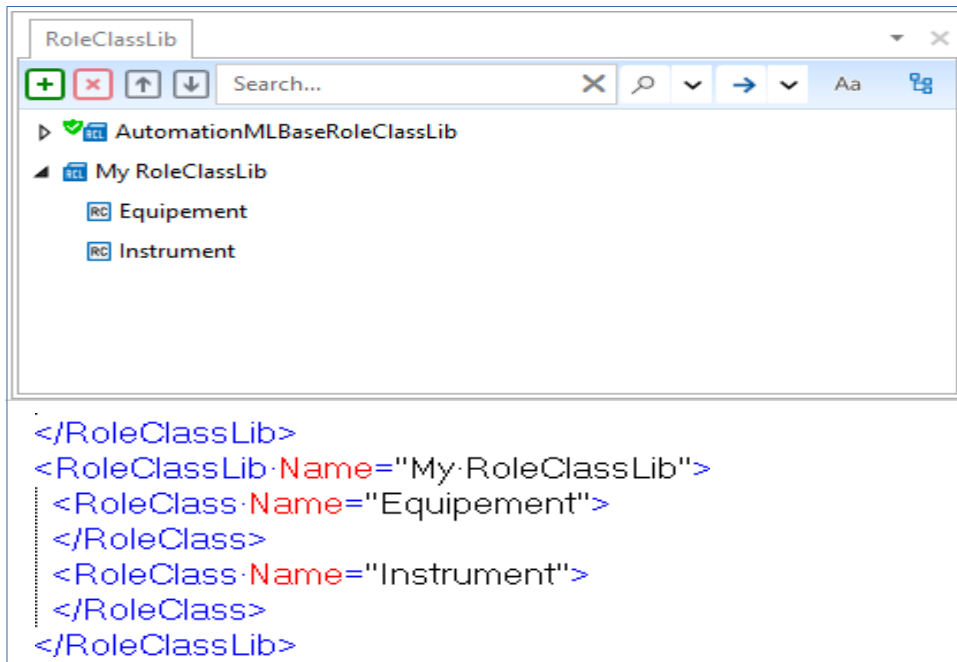


Figure 4.8: La définition de la bibliothèque my RoleClassLib avec la description XML.

- **InterfaceClassLibrary**

Elle définit les relations entre les objets. AutomationML définit certaines interfaces abstraites pour les systèmes d'automatisation générale, qui peuvent être étendues par l'utilisateur.

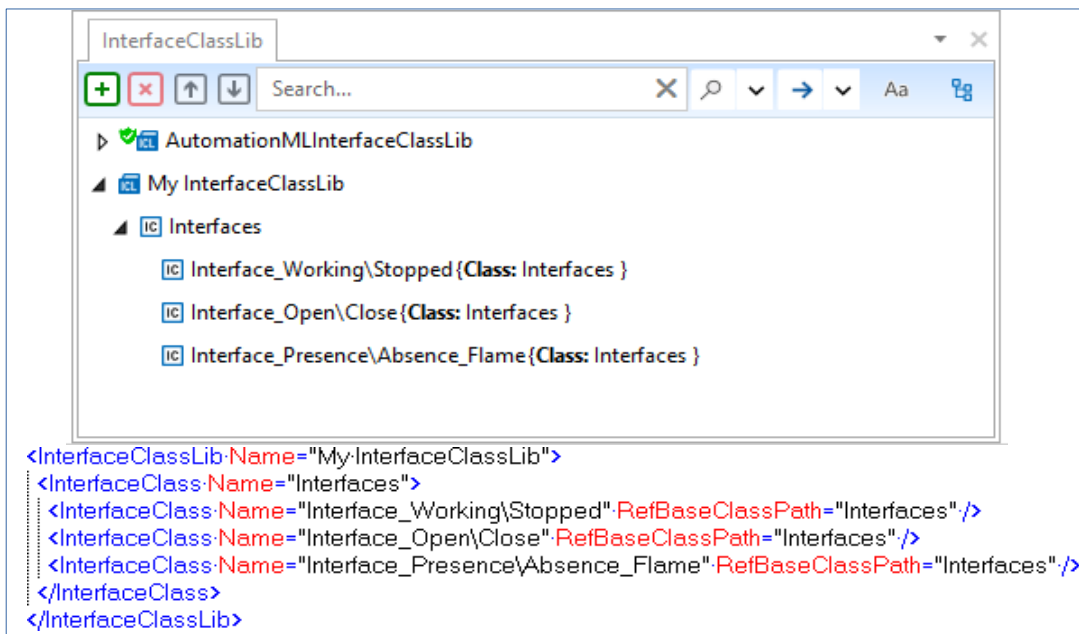
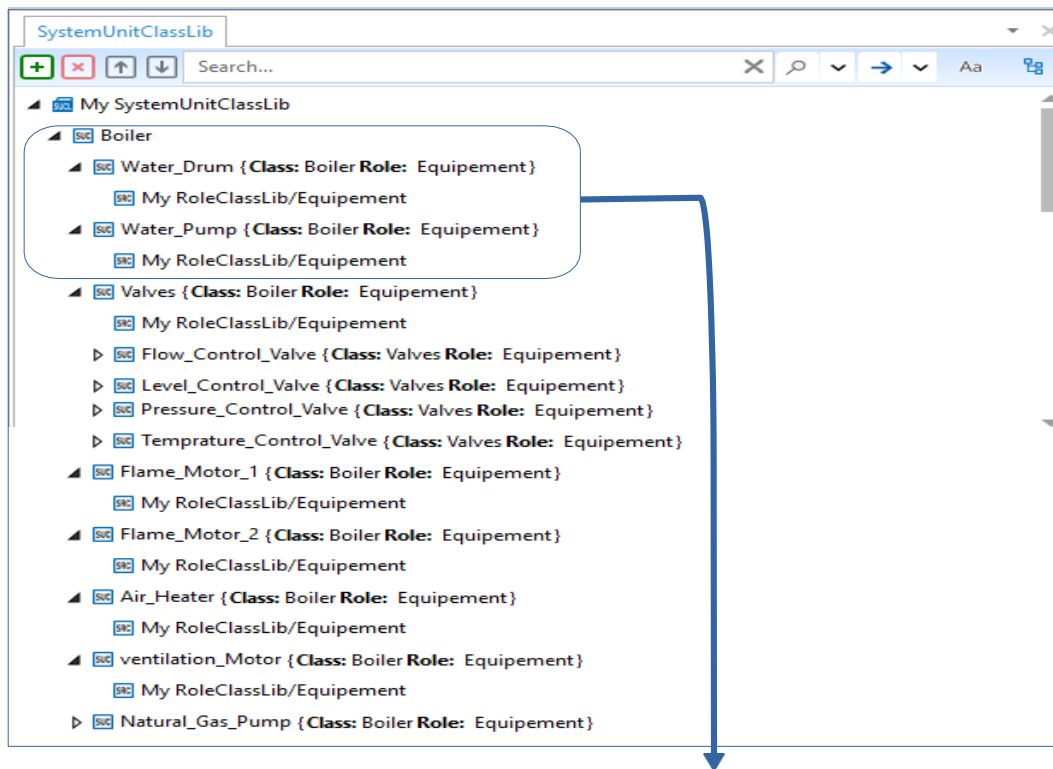


Figure 4.9: La définition de la bibliothèque my InterfaceClassLib avec la description XML.

- **SystemUnitClassLib**

Les modèles de composants d'un système de production peuvent être modélisés en tant que SystemUnitClasses. Ce sont des classes AML spécifiques à l'utilisateur et généralement dépendantes du fabricant. Les bibliothèques des unités de système (SystemUnitClass ou SUC)

peuvent être considérées comme un système de composants réutilisable pour une modélisation d'une installation industrielle. Dans notre cas d'étude, elles reflètent un répertoire des composants de la chaudière.

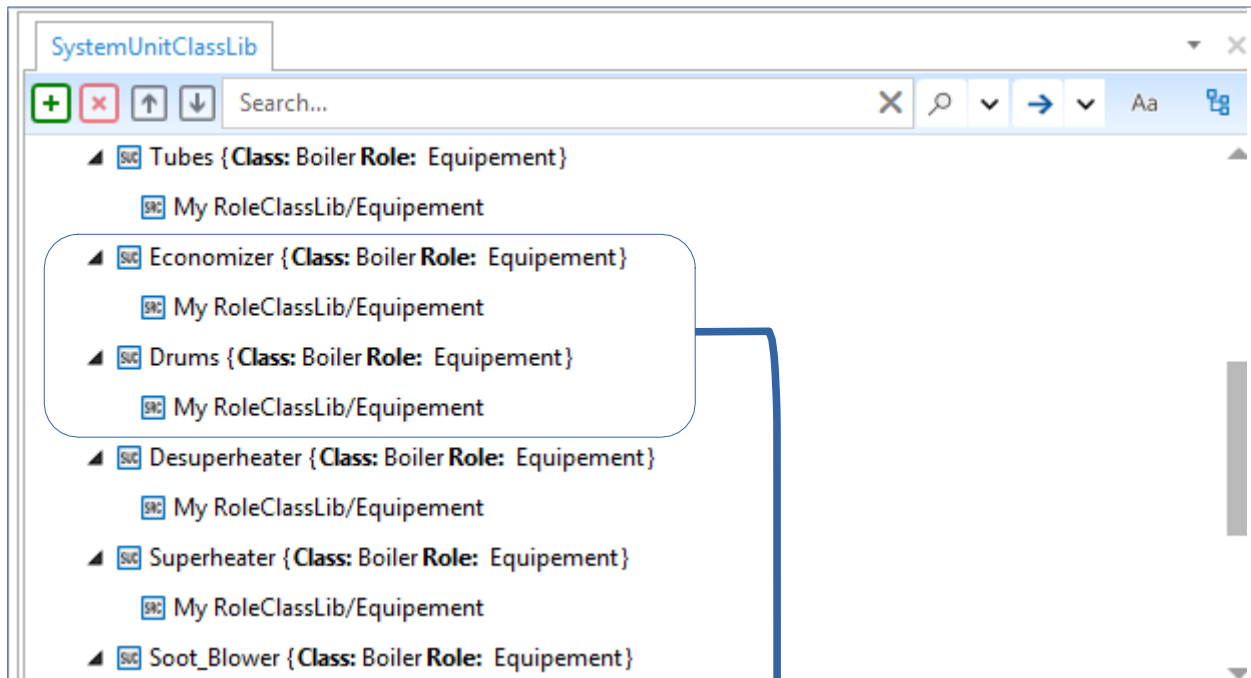


```

<SystemUnitClassLib Name="My:SystemUnitClassLib">
  <SystemUnitClass Name="Boiler" ID="6dd624af-afab-4c1e-9214-4be53f02381b">
    <SystemUnitClass Name="Water_Drum" ID="51198a7a-3963-423f-8238-bd21b61a406c" RefBaseClassPath="My-SystemUnitClassLib/Boiler">
      <SupportedRoleClass RefRoleClassPath="My-RoleClassLib/Equipement" />
    </SystemUnitClass>
    <SystemUnitClass Name="Water_Pump" ID="a808e9e3-0164-4513-8922-42fc5e126a35" RefBaseClassPath="My-SystemUnitClassLib/Boiler">
      <SupportedRoleClass RefRoleClassPath="My-RoleClassLib/Equipement" />
    </SystemUnitClass>
  </SystemUnitClassLib>

```

Figure 4.10: Un exemple d'unité de système « Réservoir d'eau » et « Pompe d'eau » avec une représentation XML



```

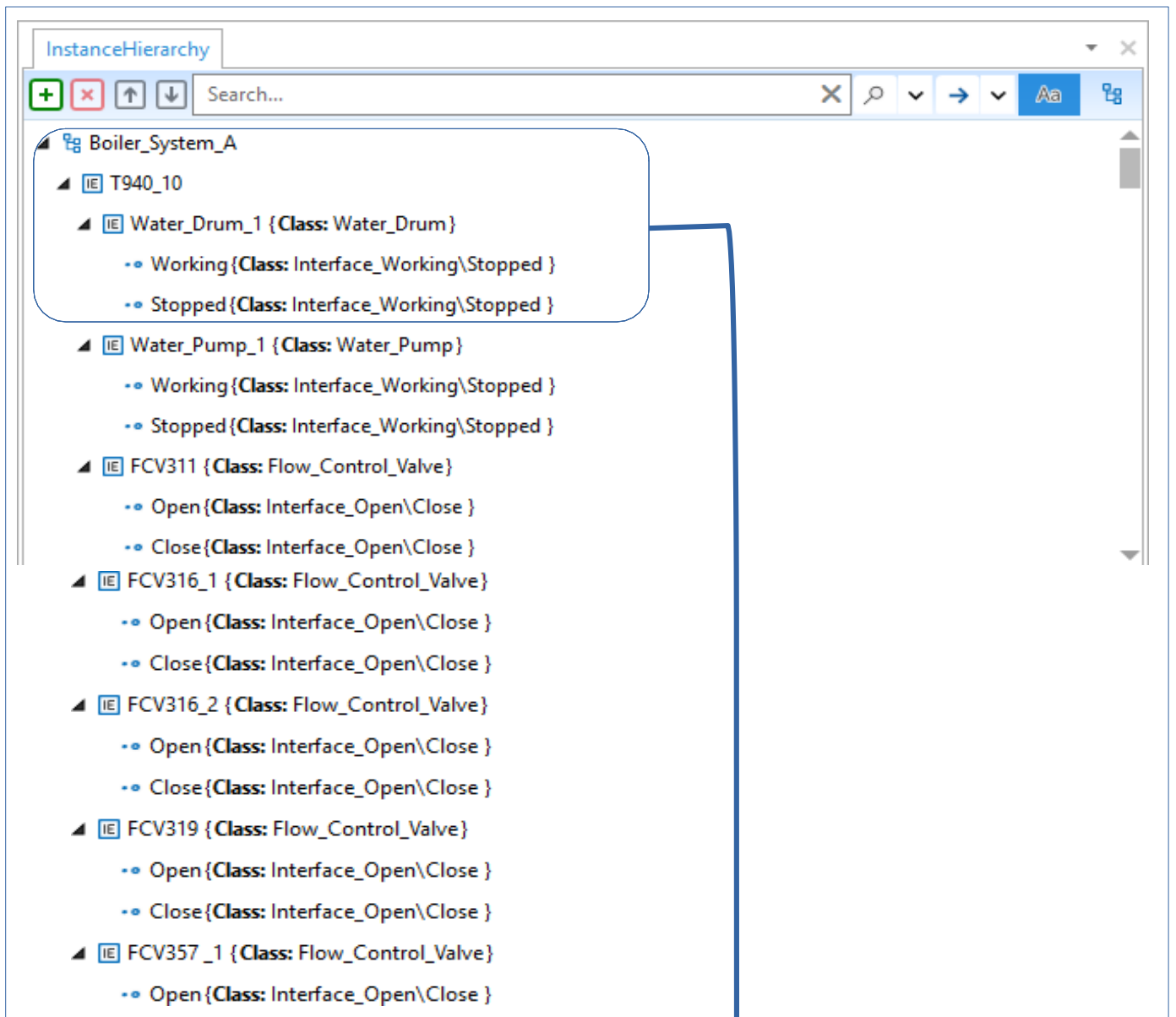
<SystemUnitClass Name="Economizer" ID="196b085b-88a9-422c-bc35-762d417bd932" RefBaseClassPath="My\SystemUnitClassLib/Boiler">
  <SupportedRoleClass RefRoleClassPath="My\RoleClassLib/Equipement" />
</SystemUnitClass>
<SystemUnitClass Name="Drums" ID="00cab039-e3ee-4fed-9423-c05b18141d1b" RefBaseClassPath="My\SystemUnitClassLib/Boiler">
  <SupportedRoleClass RefRoleClassPath="My\RoleClassLib/Equipement" />
</SystemUnitClass>

```

Figure 4.11: Un exemple d’unité de système « Economiseur » et « les ballons d’eau » avec une représentation XML.

- **InstanceHierarchy**

Les instances concrètes des modèles de composants peuvent être stockées dans la hiérarchie des instances. Ces exemples supportent les données du projet et représentent la configuration réelle de l’installation dans le monde numérique. Enfin, les composants de la chaudière sont modélisés par le biais d’une hiérarchie “instance hierarchy”(IH) semblable à une hiérarchie interne des éléments(IE).

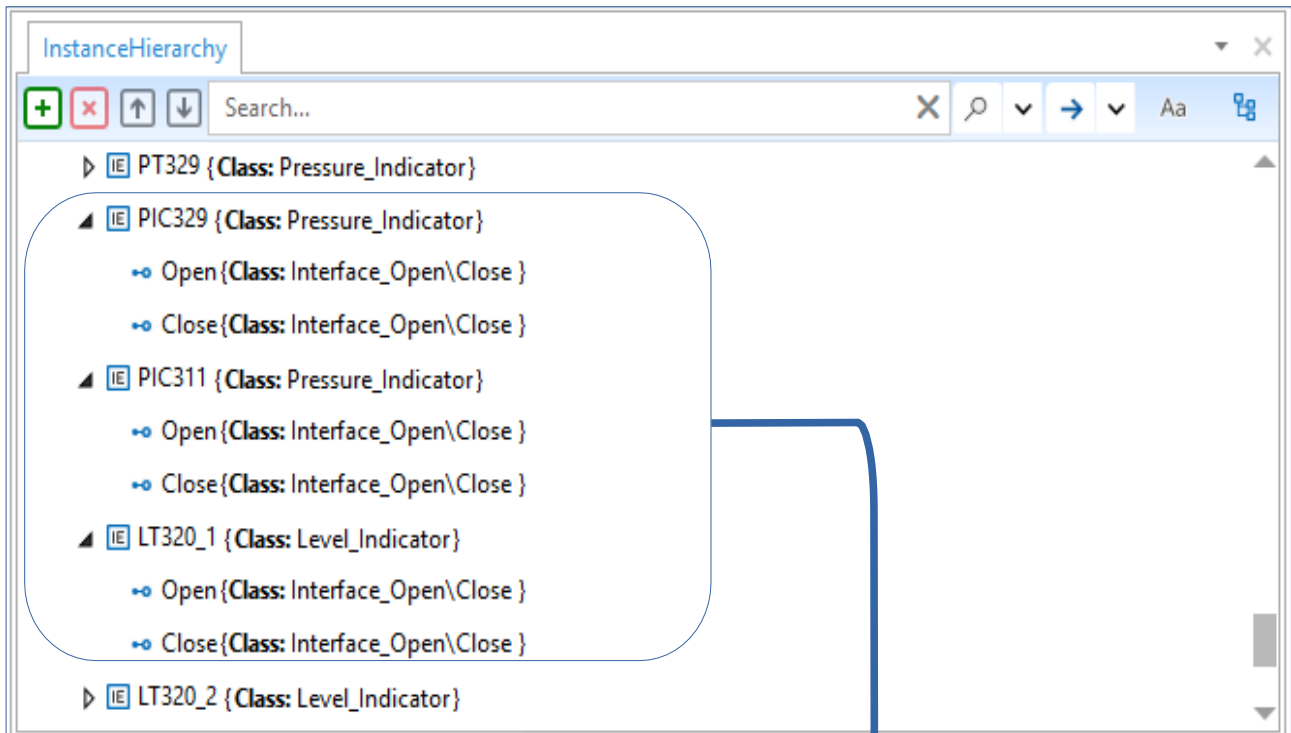


```

<InstanceHierarchy Name="Boiler_System_A">
  <Version></Version>
  <InternalElement Name="T940_10" ID="fb52e84f-0785-4ea8-aa4f-02d7f92cd6ad">
    <Attribute Name="name" AttributeDataType="xs:string">
      <Value>Chaudière A</Value>
    </Attribute>
    <Attribute Name="type" AttributeDataType="xs:string">
      <Value>MITSUBISHI-CVU-60</Value>
    </Attribute>
    <Attribute Name="pression-HP" AttributeDataType="xs:integer" Unit="kg/cm²">
      <Value>43</Value>
    </Attribute>
    <Attribute Name="température" AttributeDataType="xs:integer" Unit="C">
      <Value>430</Value>
    </Attribute>
    <Attribute Name="débit" AttributeDataType="xs:integer" Unit="T/H">
      <Value>112</Value>
    </Attribute>
    <InternalElement Name="Water_Drum_1" ID="f2c0d951-b425-4394-a19f-418af9f3dc98" RefBaseSystemUnitPath="My\SystemUnitClassLib\Boiler\Water_Drum">
      <ExternalInterface Name="Working" ID="2da7512e-ca98-4bc7-9899-8c1f7f34d2b6" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Working\Stopped"/>
      <ExternalInterface Name="Stopped" ID="ccc642b8-c789-4d54-8ea8-4af3a0fcc601" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Working\Stopped"/>
    </InternalElement>
  </InternalElement>

```

Figure 4.12: Présentation de la hiérarchie de la chaudière en XML (exemple 1).



```

<InternalElement Name="PIC329" ID="12f7e2e7-4571-4380-8dc0-9dea401fb722" RefBaseSystemUnitPath="My-SystemUnitClassLib/Boiler/Pressure_Indicator">
  <ExternalInterface Name="Open" ID="757029d5-413b-496a-8e80-dbea37126691" RefBaseClassPath="My-InterfaceClassLib/Interfaces/Interface_Open\Close"/>
  <ExternalInterface Name="Close" ID="6cd652fe-d362-460e-88d5-794c72df9064" RefBaseClassPath="My-InterfaceClassLib/Interfaces/Interface_Open\Close"/>
</InternalElement>
<InternalElement Name="PIC311" ID="a0492f98-4748-444e-b32a-98a93da85b0f" RefBaseSystemUnitPath="My-SystemUnitClassLib/Boiler/Pressure_Indicator">
  <ExternalInterface Name="Open" ID="9dbf4017-647c-4349-a477-71e9917e20a4" RefBaseClassPath="My-InterfaceClassLib/Interfaces/Interface_Open\Close"/>
  <ExternalInterface Name="Close" ID="fc58c547-72c4-4f05-9e44-36a2a579507f" RefBaseClassPath="My-InterfaceClassLib/Interfaces/Interface_Open\Close"/>
</InternalElement>
<InternalElement Name="LT320_1" ID="02b64cb6-49b2-4285-b9d2-96fcf795cbf8" RefBaseSystemUnitPath="My-SystemUnitClassLib/Boiler/Level_Indicator">
  <ExternalInterface Name="Open" ID="2544a76c-2ae5-4490-a063-19f2d03fb682" RefBaseClassPath="My-InterfaceClassLib/Interfaces/Interface_Open\Close"/>
  <ExternalInterface Name="Close" ID="594ef0c3-8538-4965-8f2d-cf83abeaec0a" RefBaseClassPath="My-InterfaceClassLib/Interfaces/Interface_Open\Close"/>
</InternalElement>

```

Figure 4.13: Présentation de la hiérarchie de la chaudière en XML (exemple 2).

- ▲ IE Combution_Chamber_1 {Class: Combution_Chamber}
 - ▷ IE BR_1 {Class: Burner_1}
 - ▷ IE BR_2 {Class: Burner_2}
 - ▷ IE BR_3 {Class: Burner_3}
 - ▷ IE BR_4 {Class: Burner_4}
- ▲ IE Economizer_1 {Class: Economizer}
 - Working {Class: Interface_Working\Stopped }
 - Stopped {Class: Interface_Working\Stopped }
- ▲ IE Desuperheater_1 {Class: Desuperheater}
 - Working {Class: Interface_Working\Stopped }
 - Stopped {Class: Interface_Working\Stopped }

```

<InternalElement Name="Combution_Chamber_1" ID="b6c33031-a5cd-49c3-b8f7-26927bb03778" RefBaseSystemUnitPath="My\SystemUnitClassLib\Boiler\Combution_Chamber">
  <InternalElement Name="BR_1" ID="ce631b0c-2d88-4580-b02d-5e95b0be9779" RefBaseSystemUnitPath="My\SystemUnitClassLib\Boiler\Combution_Chamber\Bumer_1">
    <ExternalInterface Name="presence_Flame" ID="c8c9b294-3303-4871-b01a-9ae335a64dc2" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame"/>
    <ExternalInterface Name="Absence_Flame" ID="9a73f230-46ad-4fa3-9ec7-ffa5435fa1fe" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame">
      <Attribute Name="temperature" AttributeDataType="xs:integer"/>
    </ExternalInterface>
  </InternalElement>
  <InternalElement Name="BR_2" ID="90c14243-4e7e-4cfe-bcac-f39e69f3ef7e" RefBaseSystemUnitPath="My\SystemUnitClassLib\Boiler\Combution_Chamber\Bumer_2">
    <ExternalInterface Name="Presence_Flame" ID="35b83f38-5b6f-4012-9647-782e7ab00725" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame"/>
    <ExternalInterface Name="Absence_Flame" ID="c6830fac-0e7c-4fe6-9e8f-cb8643b3699" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame">
      <Attribute Name="temperature" AttributeDataType="xs:integer"/>
    </ExternalInterface>
  </InternalElement>
  <InternalElement Name="BR_3" ID="f7d7d8b7-1234-426e-a674-d373578c2c0" RefBaseSystemUnitPath="My\SystemUnitClassLib\Boiler\Combution_Chamber\Bumer_3">
    <ExternalInterface Name="presence_Flame" ID="ee3a1d4d-cbe5-4183-871e-dcc7f7c1394a" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame"/>
    <ExternalInterface Name="Absence_Flame" ID="190ab2d8-4282-4b86-b200-5009be135061" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame">
      <Attribute Name="temperature" AttributeDataType="xs:integer"/>
    </ExternalInterface>
  </InternalElement>
  <InternalElement Name="BR_4" ID="4aafc76d-81a8-47ab-b508-46de54934baa" RefBaseSystemUnitPath="My\SystemUnitClassLib\Boiler\Combution_Chamber\Bumer_4">
    <ExternalInterface Name="presence_Flame" ID="4150a7fc-f8c3-4fa4-a148-fe7afeddd8899" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame"/>
    <ExternalInterface Name="Absence_Flame" ID="d5dde93c-2238-42a8-ab89-0029da525b6c" RefBaseClassPath="My\InterfaceClassLib\Interfaces\Interface_Presence\Absence_Flame">
      <Attribute Name="temperature" AttributeDataType="xs:integer"/>
    </ExternalInterface>
  </InternalElement>
</InternalElement>

```

Figure 4.14: Présentation de la hiérarchie de la chaudière en XML (exemple 3).

2.1.3 Phase de description du comportement dynamique

2.1.3.1 Modélisation de diagrammes de Gantt

Les diagrammes de Gantt sont une représentation graphique de modèles d'événements discrets généralement utilisés pour décrire l'ordre et le temps d'exécution des activités, qui sont représentés sous forme de barres, à un niveau d'abstraction élevé. Ils sont utilisés dans les premières phases d'ingénierie de l'usine pour représenter la synchronisation des processus de fabrication. Les informations principales stockées dans les diagrammes de Gantt sont les heures de début et de fin des barres et les relations prédécesseur / successeur entre les barres.

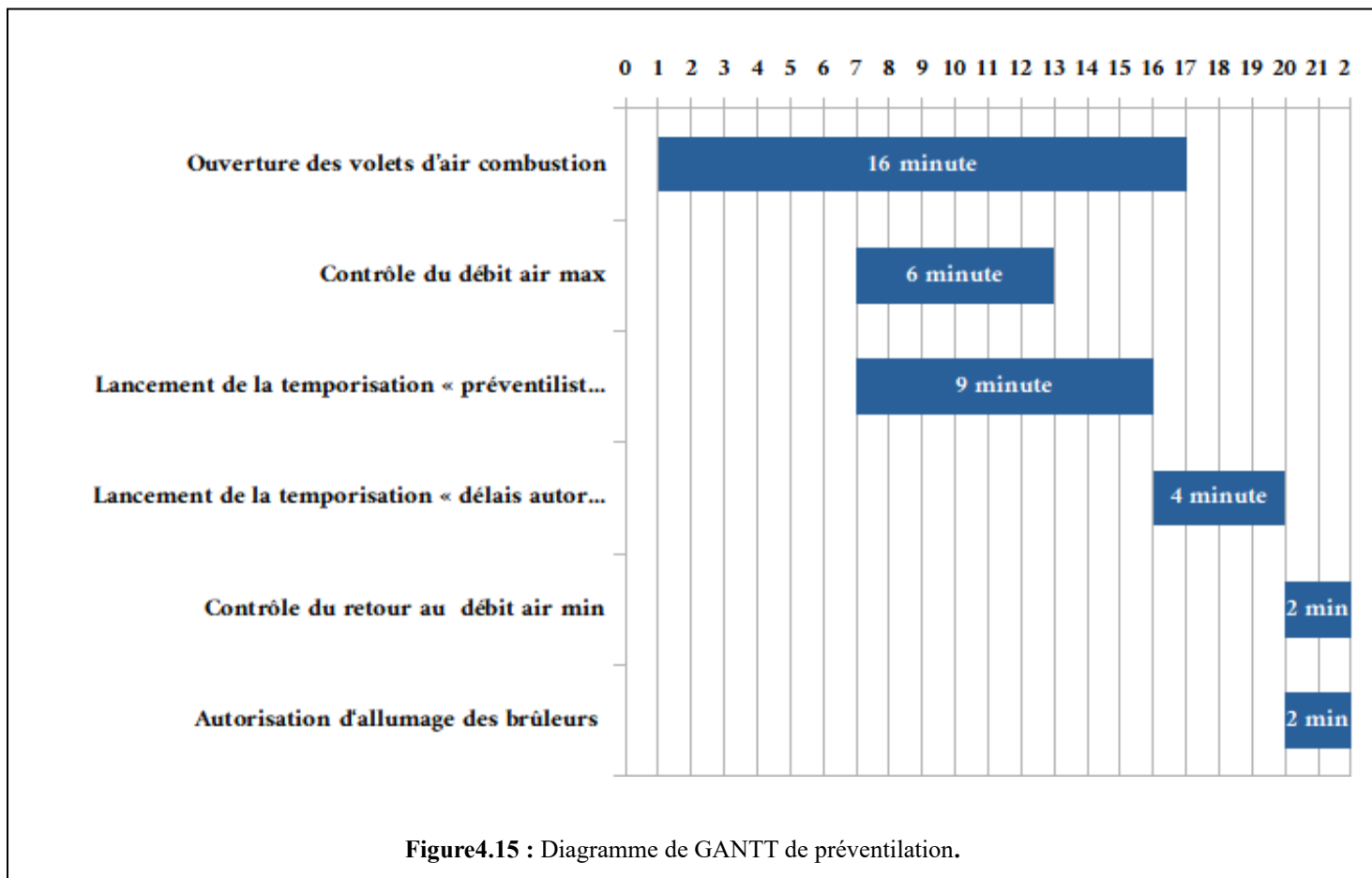
Un exemple de diagramme de Gantt est donné aux figures suivantes. Il décrit les processus des séquences de mise en service chaudière.

➤ Préventilation

Les actions de la préventilation sont les suivantes :

Tableau 4 : La table Les actions de la préventilation.

| <i>ID</i> | <i>Name</i> | <i>Predecessor</i> | <i>Start Time</i> | <i>Duration</i> | <i>End Time</i> |
|-----------|-------------------------------------------------------------------------|--------------------|-------------------|-----------------|-----------------|
| 1 | Ouverture des volets d'air combustion | / | 1 | 16 | 17 |
| 2 | Contrôle du débit air max | / | 7 | 6 | 13 |
| 3 | Lancement de la temporisation « préventilistion » | / | 7 | 9 | 17 |
| 4 | Lancement de la temporisation « délais autorisation d'allumage » | 3 | 16 | 4 | 20 |
| 5 | Contrôle du retour au débit air min | 4 | 20 | 2 | 22 |
| 6 | Autorisation d'allumage des brûleurs | 4 | 20 | 2 | 22 |



➤ **Allumage du premier brûleur**

Les actions de l'allumage du premier brûleur sont les suivantes :

Tableau 6: Les actions de l'allumage du premier brûleur

| <i>ID</i> | <i>Name</i> | <i>Predecessor</i> | <i>Start Time</i> | <i>Duration</i> | <i>End Time</i> |
|-----------|---------------------------------------------------|--------------------|-------------------|-----------------|-----------------|
| 1 | Ouverture de la vanne purge | / | 0 | 13 | 13 |
| 2 | Mise en service du transformateur d'allumage | / | 1 | 4 | 5 |
| 3 | Prise en compte de la flamme torche | 2 | 5 | 4 | 9 |
| 4 | Détection flamme de l'allumeur | 3 | 9 | 4 | 13 |
| 5 | Ouverture de la vanne amont gaz | 1, 4 | 13 | 10 | 23 |
| 6 | Ouverture de la vanne aval gaz | 1, 4, 5 | 13 | 4 | 17 |
| 7 | Prise en compte de la détection flamme principale | 6 | 17 | 2 | 19 |
| 8 | Lancement temporisation de stabilisation allumage | 7 | 19 | 1 | 20 |

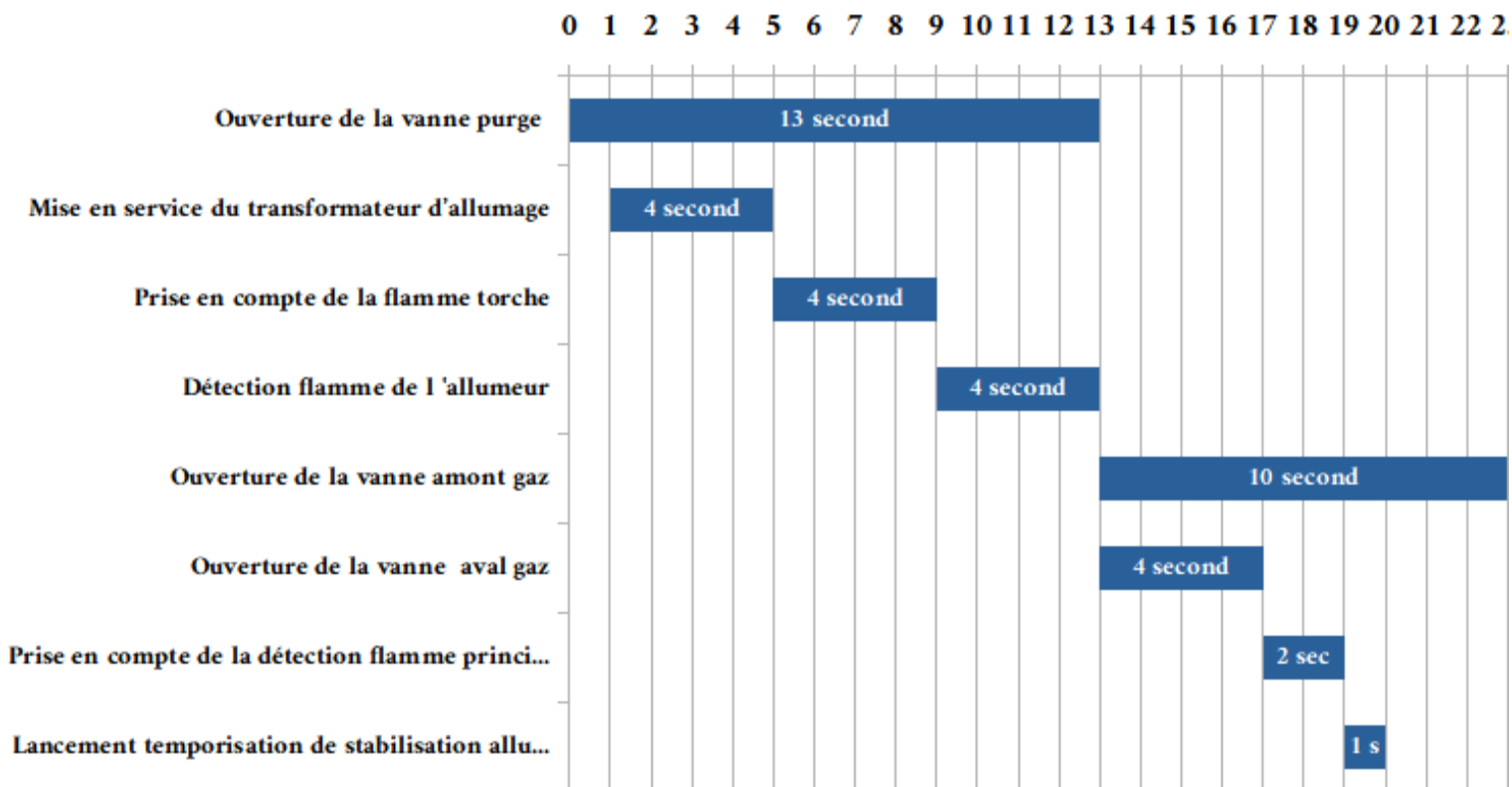


Figure 4.16: Diagramme de GANTT d'Allumage du premier brûleur.

2.2 Le mapping de GANTT vers le format IML

Les diagrammes de Gantt sont généralement utilisés pour décrire l'ordre, c'est-à-dire la séquence et la concurrence, et le temps d'exécution des activités, appelées barres dans ce qui suit, à un niveau d'abstraction élevé dans les premières étapes de l'ingénierie de l'usine. Les principales informations stockées sont : l'heure de début, l'heure de fin et la durée des barres, ainsi que les relations de prédécesseur et de successeur entre les barres. Cette section décrit les concepts de base de la représentation des diagrammes de Gantt à l'aide d'éléments IML. La structure du modèle IML résultant est dérivée du diagramme de Gantt selon les principes suivants :

1. Les diagrammes de Gantt sont mis en correspondance avec les systèmes IML pour prendre en charge les barres parallèles.
2. Les barres des diagrammes de Gantt doivent être associées à des activités.
3. Ces activités doivent être associées à des états.
4. Si aucune relation explicite n'est définie entre les barres de Gantt, c'est-à-dire qu'elles sont simultanées, l'ordre des états IML associés est parallèle, c'est-à-dire qu'ils n'ont aucune relation d'ordre basée sur les séquences de transition état-état.
5. Si les relations prédécesseur/successeur entre les barres de Gantt sont décrites, la structure résultante est une séquence d'états IML et de transitions d'état .
6. Pour chaque SFC représentant un diagramme de Gantt, un état initial doit être créé, suivi d'une transition d'état avec la condition « vrai » comme lien vers tous les autres éléments.

7. Chaque barre de Gantt est représentée par un état avec deux activités associées et une transition d'état successeur.

- a) La première activité représente directement le comportement temporel de la barre de Gantt correspondante et peut être retardée dans son exécution afin de démarrer de manière « synchrone » avec la barre de Gantt appropriée (voir la ligne rouge dans la figure 15). L'activité aura donc une définition de son point de départ au plus tôt. La convention d'appellation de cette action est la suivante : « DA_ » + nom de la barre de Gantt.
- b) La deuxième activité est chargée de permettre à la transition de désactiver l'état de manière « Synchrone » en fonction de la fin de la barre de Gantt. L'activité contiendra donc une définition de la durée afin de retarder le début de l'activité et de la stocker pour des raisons de synchronisation. La convention d'appellation de cette activité est la suivante « TA_ » + nom de la barre de Gantt.

8. Comme les diagrammes de Gantt ont généralement une heure globale, alors que les activités IML utilisent l'heure locale à partir de l'activation de l'état, une conversion entre les deux systèmes de temps est nécessaire. Pour les barres sans prédécesseurs, le point temporel de référence est le point de départ du diagramme de Gantt. Pour les barres avec prédécesseurs, le point temporel de référence est la fin de la dernière barre prédécesseur.

9. Si une barre de Gantt a deux ou plusieurs barres successeurs, elle est reliée à celles-ci par une divergence simultanée.

10. Si une barre de Gantt a deux prédécesseurs ou plus, ils sont reliés à eux par une convergence simultanée SFC. Entre cette convergence simultanée et les états des prédécesseurs, il y a des états de synchronisation sans activités et une transition de successeur avec la condition TRUE.

11. Chaque diagramme de Gantt a un état de départ unique et un état de fin unique.

2.2.1 Présentation des éléments du système IML

Chaque système IML est constitué des éléments de modèle suivants :

- **Header h**

Header h est constitué d'un identifiant, d'un nom et d'un ensemble de tous les éléments IML appartenant au système IML.

- **State**

Un état décrit une situation stable dans un système où un ensemble de propriétés dédié est valide. Ces propriétés peuvent être données en exécutant des actions, des événements et des valeurs de variables de processus. Les états sont atteints et laissés par des transitions d'état. Les états spéciaux sont les états initiaux, actuels et terminaux.

- **State Transition**

Une transition d'état est un changement d'un état à un ou plusieurs états suivants, en interprétant les conditions de transition d'état.

- **Activity**

Une activité décrit une ou plusieurs opérations liées à certains états. Elle est caractérisée par des variables requises et modifiées, des propriétés de temps locales et des événements possibles déclenchés après son exécution. Les activités spéciales sont les

activités initiales, actuelles et terminales.

Remarque : Les activités ne peuvent pas être associées à des transitions dans IML.

- **Event**

Un événement est déclenché par une activité à laquelle il est associé. Les événements sont principalement utilisés comme déclencheurs pour les transitions d'état.

- **Variable**

Une variable est un attribut de modélisation qui caractérise les états et les activités. Sa valeur peut être modifiée par les activités. Il peut être utilisé comme conditions de déclenchement pour les transitions d'état ou comme entrée et sortie du système.

3 Additional Data

Des données supplémentaires permettent de stocker des informations au-delà des définitions de SFC dans les deux IML et CEI 61131-3. Exemples : Informations de synchronisation complexes, Informations d'exécution et de données descriptives.

3.1.1 Propriétés et relations de l'élément

Les éléments IML possèdent des propriétés et des relations avec d'autres éléments tel que :

- **Propriétés de l'en-tête et relations**

Header h est caractérisé par les propriétés suivantes :

- h.ID représente l'identification unique du système IML.
- h.Name représente le nom unique du système IML. Un Header h a les relations suivantes :

- **Propriétés et relations de l'État**

State s est caractérisé par les propriétés suivantes :

- s.ID représente l'identification unique de s dans un système IML.
- s.Name est le nom unique de s dans un système IML.
- s.Init est une valeur booléenne indiquant que s'est un état initial.
- s.Current est une valeur booléenne indiquant que s'est un état activé courant.
- s.Terminal est une valeur booléenne indiquant que s'est un état terminal. Chaque état peut avoir les relations suivantes :
- s.Pre est l'ensemble des prédécesseurs de s couvrant les références aux convergences de sélection, aux divergences simultanées et aux transitions d'états.

- **Propriétés et relations de transition d'état**

Un state transition est caractérisé par les propriétés suivantes :

- st.ID représente l'identification unique de st dans un système IML.

- st.Name représente le nom unique de st dans un système IML.
- st.Guard.Boolean nomme la variable booléenne unique donnant le tir condition.
- st.Guard.Value nomme une variable unique et un intervalle fermé de valeurs valides de la variable en tant que condition de déclenchement nécessaire.

- **Propriétés de l'activité et relations**

Une activité a est caractérisée par les propriétés suivantes :

- a.ID représente l'identification unique d'un dans un système IML.
- a.Name représente le nom unique d'un dans un système IML.
- a.Init est une valeur booléenne indiquant que a est une activité initiale.
- a.Current est une valeur booléenne indiquant que a est une activité en cours d'exécution.
- a.Terminal est une valeur booléenne indiquant que a est une activité terminale.
- a.Formula nomme une formule de texte structuré.
- a.Time représente la condition temporelle de a, utilisant toutes les secondes (s)
- a.Time.Duration est une valeur réelle non négative représentant la durée de a.
- a.Time.Start est une plage (plus ancienne, plus récente) avec des valeurs réelles non négatives comme limites pour le point de départ de a.
- a.Time.End est une plage (la plus ancienne, la plus récente) avec des valeurs réelles non négatives comme limites pour le point final de a.
- a.Time.Delay est une valeur réelle non négative représentant le délai de démarrage a. Une activité a peut avoir les relations suivantes :
 - a.FiredEvent nomme tous les événements déclenchés à la fin de a.
 - a.Pre représente l'ensemble des prédécesseurs de a couvrant la relation avec les états prédécesseurs

- **Propriétés et relations variables**

Une variable var est caractérisé par les propriétés suivantes :

- var.ID est l'identifiant unique de var dans un système IML.
- var.Name est le nom unique de var dans un système IML.
- var.Type est le type de données de var suivant les types de données définis par la CEI 61131-3.
- var.SIUnit est le type de mesure de var suivant le système SI des unités de mesure.
- var.InitialValue est la valeur initiale de var.
- var.Address est l'adresse physique de var définie par la norme CEI 61131-3.

- var.Content décrit l'utilisation de var comme variable locale, entrée, sortie. Valeur par défaut est locale.

```

page1.java | GanttToIMLConverter.java | page2.java
// Méthode pour transformer les données du tableau en IML
private void transformToIML() {
    StringBuilder imlContent = new StringBuilder();

    // État et transition initiaux
    imlContent.append("state s with\n");
    imlContent.append("s.ID = 1\n");
    imlContent.append("s.Name = \"InitialStep\"\n");
    imlContent.append("s.Init = true\n");

    imlContent.append("state transition st with\n");
    imlContent.append("st.ID = 1\n");
    imlContent.append("st.Name = \"InitialTransition\"\n");
    imlContent.append("st.Guard.Formular = True\n");
    imlContent.append("st.Pre = 1\n");

    imlContent.append("addData ad with\n");
    imlContent.append("ad.ID = 1\n");
    imlContent.append("ad.Type = ChartType\n");
    imlContent.append("ad.Value = \"Gantt\"\n");
    imlContent.append("ad.Pre = 1\n");

    // Boucle à travers les données du tableau
    for (int i = 0; i < tableModel.getRowCount(); i++) {
        String id = tableModel.getValueAt(i, 0).toString();
        String name = tableModel.getValueAt(i, 1).toString();
    }
}

```

Figure 4.17: Le mapping de GANTT vers le format IML.

3.2 Le mapping de l'IML vers le format XML de PLCopen

La traduction d'IML en PLCopen XML est basée sur la similarité entre IML et SFC suivant la norme IEC 61131-3 et la capacité de PLCopen XML à stocker les SFC. Dans le cadre de ce processus de traduction, les différents éléments IML et leurs paramètres et relations sont, si possible, traduits un par un vers des éléments SFC et stockés en tant que tels avec PLCopen XML. Les informations requises par PLCopen XML mais non données dans IML sont générées de manière unique. Les informations fournies dans IML mais non directement exprimables par PLCopen XML sont mappées sur des données supplémentaires.

Le mapping général est basé sur les principes suivants :

- Chaque système IML, c'est-à-dire son en-tête et ses éléments, est mappé à un module et à des sous-éléments correspondants.
- Chaque élément d'état d'IML est mappé à un élément en une étape.
- Chaque élément de transition d'état d'IML est mappé à un élément de transition.
- Chaque élément d'activité d'IML est mappé à un élément d'action dans un élément de bloc d'action.
- Chaque élément de saut d'IML est mappé à un élément de saut.

- Chaque élément d'événement d'IML est mappé à un élément variable.
- Chaque élément variable d'IML est mappé à un élément variable.

Selon la structure du schéma XML PLCopen, toutes les informations sont stockées dans les parties d'interface, d'action, de transition ou de SFC d'un POU à l'intérieur du document PLCopen. Une partie du résultat de mapping d'IML vers le PLCopen XML de notre travail est présenté dans la figure suivante :

```

// Méthode pour transformer IML vers PLCopenXML
private void transformToPLCFromIMLText() {
    String imlText = imlTextArea.getText();
    StringBuilder plcContent = new StringBuilder();

    // Header PLCopenXML
    plcContent.append("<pou name=\"Program\" pouType=\"program\">\n");
    plcContent.append("<interface>\n");
    plcContent.append("<localVars>\n");
    plcContent.append("</localVars>\n");
    plcContent.append("</interface>\n");
    plcContent.append("<body>\n");
    plcContent.append("<SFC>\n");

    // Transformation des données IML en PLCopenXML
    String[] lines = imlText.split("\n");
    for (int i = 0; i < lines.length; i++) {
        String line = lines[i];
        if (line.startsWith("state s with")) {
            String id = lines[++i].split(" ")[1];
            String name = lines[++i].split(" ")[1].replace("\\", "");

            plcContent.append("<step localId=\"\" .append(id) .append(\"\" name=\"\" .append(name) .append(\">\n");

            if (name.equals("InitialStep")) {
                plcContent.append("<initialStep/>\n");
            }
        }
    }
}

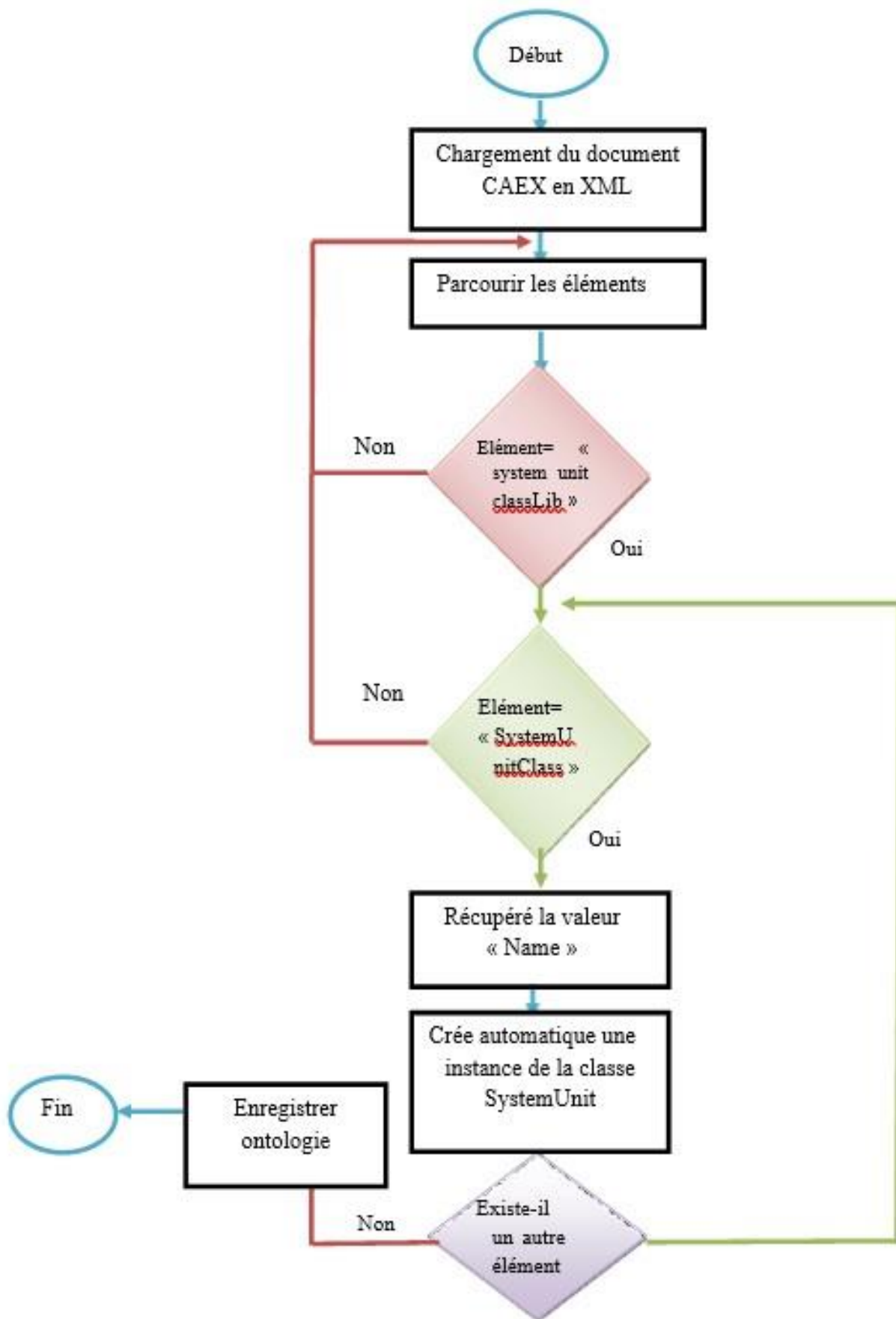
```

Figure 4.18: Le mapping de GANTT vers le format IML

3.3 Mapping automatique CAEX-OWL

Une ontologie de base AML qui reflète les principales décisions de conception pour représenter les structures CAEX dans OWL, c'est-à-dire les structures CAEX qui se trouvent dans n'importe quel fichier AML et qui ne sont pas spécifiques à la discipline ou à l'outil. L'ontologie principale de l'AML, ainsi qu'un fichier AML spécifique, servent de contribution à la transformation CAEX à OWL, ce qui produit l'ontologie AML étendue comme sortie (Amira & Samiya, n.d.).

Une ontologie AML étendue représente la sémantique complète et les données d'un fichier AML dans OWL. Il comprend tous les Rôles, les unités du système et les interfaces définies dans un fichier AML ainsi que toutes les données hiérarchiques d'instance. Les algorithmes que nous avons proposés pour la création de l'ontologie finale (Peuplement de l'ontologie AML de base) sont illustré sous forme d'organigramme dans les figures suivantes :



4 Conclusion

Dans ce chapitre nous avons présenté une architecture générale d'un système de création automatique d'ontologie à partir du CAEX pour la surveillance industrielle. Certains composants de cette architecture ont été implémentés pour illustrer non seulement son fonctionnement mais aussi les possibilités offertes par l'intégration des aspects sémantiques dans le domaine industriel. Les détails d'implantation feront l'objet du prochain chapitre

Chapitre 05

Implémentation

1 Introduction

Une fois que nous avons terminé l'analyse théorique de notre système, notre prochaine étape consiste à mettre nos idées en action et à développer l'application décrite dans le chapitre précédent. Il s'agit de transformer notre concept dans un format facilement compréhensible par la machine. Pour commencer, nous donnerons un aperçu des différents outils, plates-formes et éditeurs que nous avons utilisés, ainsi que des langages de programmation utilisés pour créer notre produit final. Suite à cela, nous présenterons des captures d'écran de notre logiciel pour fournir une représentation visuelle de ses fonctionnalités.

2 Programmation XML

Le XML ou **eXtensible Markup Language** est un langage informatique de balisage générique. Le XML 1.0 est devenu une recommandation du W3C (le "World Wide Web Consortium") le 10 février 1998. L'objectif de XML est de faciliter les échanges de données entre les machines. Le XML se veut également compatible avec le web afin que les échanges de données puissent se faire facilement à travers le réseau Internet. Il est standardisé, simple, mais surtout extensible et configurable afin que n'importe quel type de données puisse être décrit.

2.1 Les parseurs XML

Le parseur est l'élément de programmation le plus important, puisque c'est lui qui réalise le travail d'analyse du document XML. Son rôle est de vérifier la cohérence du document XML (en termes syntaxique et/ou par rapport à un schéma ou une DTD) et de transmettre à l'application les informations utiles au traitement du document. Tous les parseurs n'offrent pas la même puissance, certains étant non validant et d'autres validant. De plus, certains supportent une API SAX ou bien DOM, qui permettent au parseur d'entrer en communication avec l'application. Pour une application, un parseur fait partie d'une API ou d'une bibliothèque : ce n'est pas un organe indépendant mais un bloc de traitement que l'on peut contrôler, comme n'importe quelle méthode externe

2.2 Sax

L'API XML simple (SAX) établit un moyen de communication entre le parseur et l'application, en lien avec un mécanisme événementiel. Il s'agit d'un projet Open Source [90] avec deux versions d'une API, dont la deuxième version est la seule capable de prendre en compte les espaces de noms.

Le parseur divise le document en petites parties et envoie ces parties à l'application au fur et à mesure de l'étude. Une unité pourra symboliser, par exemple, l'ouverture d'un élément ou sa fermeture, la rencontre d'un discours... Dans ce système, on comprendra que l'application ne représente pas l'ensemble du document, mais plutôt que le parseur ne fournit qu'un cheminement dans le document, que l'application puisse le stocker ou non.

2.3 API DOM

DOM (Document Object Model) est une représentation objet d'un document XML. Chaque classe (ou type) d'objets provient des types de noeuds dans un document XML (élément, texte...). DOM est une API disponible dans la plupart des langages et maintenue par le W3C. Elle impose un modèle mémoire d'un document XML. Cela signifie qu'il est mal adapté à de gros documents XML. Dans tous les autres cas, cette représentation mémoire (par objet) offre beaucoup de souplesse en termes de navigation dans un document ou bien en termes de modification.

3 Langage de programmation java

Pour le choix de programmation de notre système nous avons opté pour le langage JAVA et cela pour de nombreuses raisons : c'est un langage orienté objet simple, qui réduit le risque des erreurs d'incohérences. Il est indépendant de toute plateforme et il est possible d'exécuter des programmes JAVA sur tous les environnements qui possèdent une Java Virtual Machine (JVM). Il est doté d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, menus, graphismes, boîtes de dialogue, contrôles), la programmation multi-threads (multitâches) et la gestion des exceptions, afin de faciliter le développement des applications notamment des parsers pour XML et OWL. Il permet un accès aux bases de données simplifié soit à travers la passerelle JDBC-ODBC ou à travers un pilote JDBC spécifique au SGBD. Il est caractérisé aussi par la réutilisation de son code ainsi que la simplicité de sa mise en œuvre.

4 Environnement de développement

4.1 Protégé

Protégé OWL (The Protégé Ontology Editor and Knowledge Acquisition System, 2006) est un éditeur qui permet de construire une ontologie pour un domaine donné. Protégé est un éditeur de base de connaissance indépendant de la plateforme. C'est un environnement à base de frames pour le développement des systèmes à base de connaissances. Protégé est également une librairie Java qui peut être étendue pour créer de véritables applications à base de connaissances en utilisant un moteur d'inférence pour raisonner. Le modèle de connaissances de Protégé OWL contient des classes (concepts), des slots (propriétés) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Il autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. Dans le contexte du web sémantique des « plugin » pour les langages RDF et OWL ont été développés pour Protégé. Ces « plugin » permettent d'utiliser Protégé comme éditeur d'ontologies pour ces différents langages et de créer des instances.

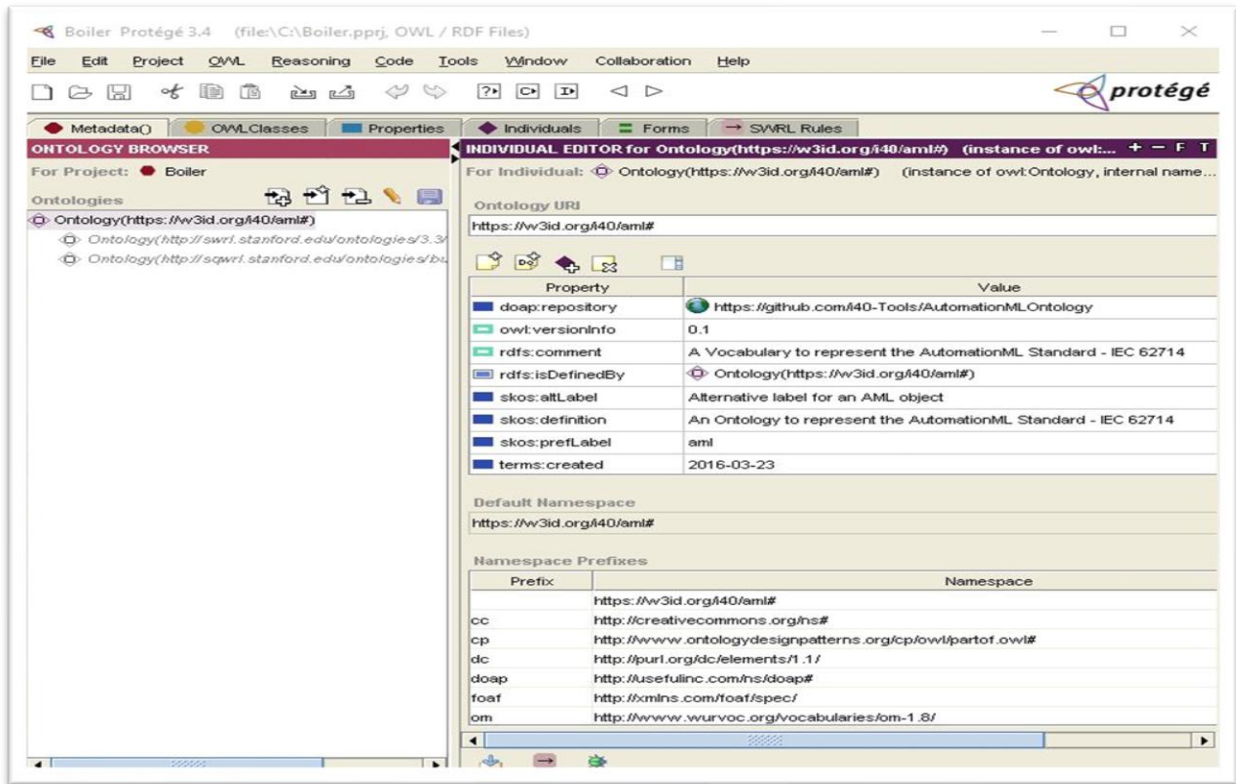


Figure 5.1: Présentation de la fenêtre Protégé.

4.2 JCreator

C'est un environnement de développement intégré pour Java. Permet, en plus de gérer des fichiers source individuels, de créer des projets. Que ce soit une application en console, une application avec interface graphique, un applet, ou autre, cela permet de nombreuses fonctionnalités additionnelles pour simplifier la gestion de projets.

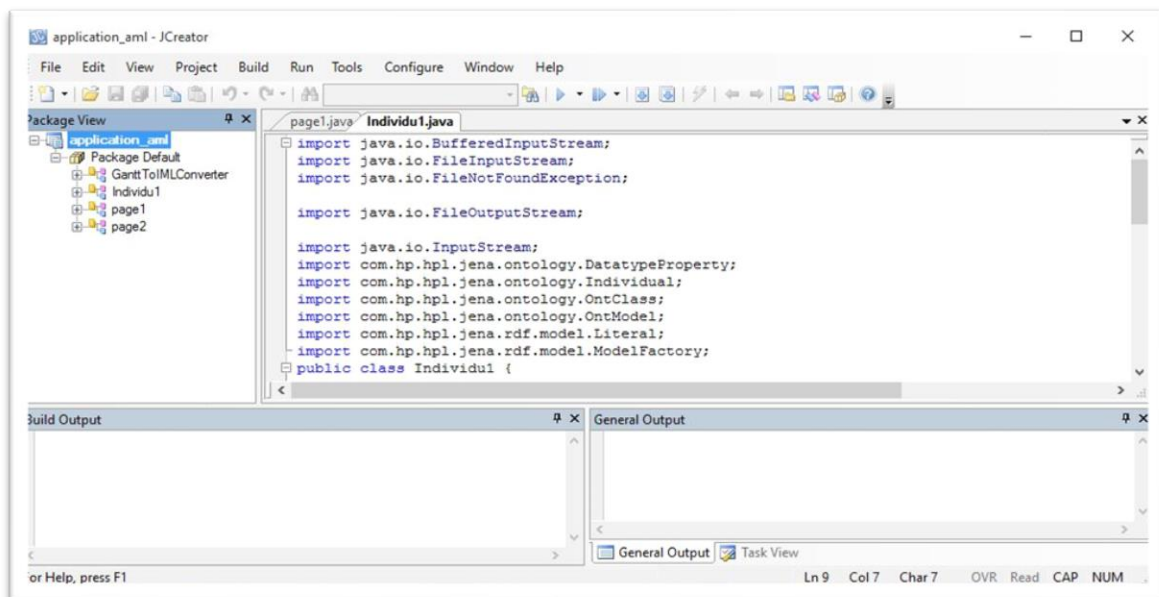


Figure 5.2 : Présentation de la fenêtre JCreator.

4.3 L'API JENA

JENA est un environnement de travail open source en Java, pour la construction d'application web sémantique. JENA permet de manipuler des documents RDF, RDFS, OWL et SPARQL. Il fournit un moteur d'inférences permettant des raisonnements sur les ontologies.

5 AML

AutomationML (Automation Markup Language) est un format d'échange de données basé sur un schéma XML conçu pour le stockage des informations d'ingénierie de l'installation industrielle indépendamment du fournisseur. AML a pour objectif de gérer l'hétérogénéité des outils d'ingénierie dans leurs différentes disciplines (Ingénierie d'installations mécaniques, conception électrique, développement d'IHM, programmation d'automates programmables et de commandes de robots.). AML a été bien présenté dans le chapitre 3 partie 1.

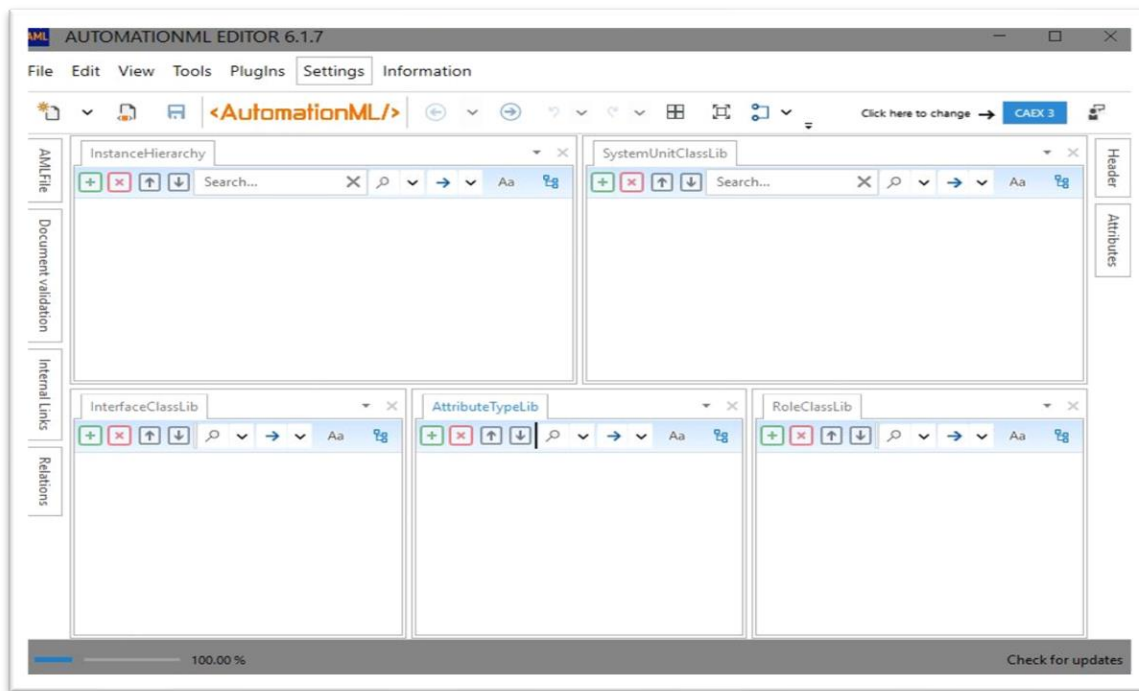


Figure 5.3 : Présentation de la fenêtre AutomationML

6 Editeur SWRL

L'éditeur SWRL (Bouaroudj Samiya & Boufaïda, 2009) est un plugin de Protégé OWL permettant l'édition interactive des règles SWRL. Les utilisateurs peuvent créer, modifier, lire et écrire les règles SWRL.

L'éditeur SWRL permet aux utilisateurs de saisir des règles complètement sous forme de texte. Toutefois, il permet également aux utilisateurs de sélectionner des classes OWL, propriété et individus de la base de connaissances (Ontologie) chargée et les insérer dans la règle en cours de création.

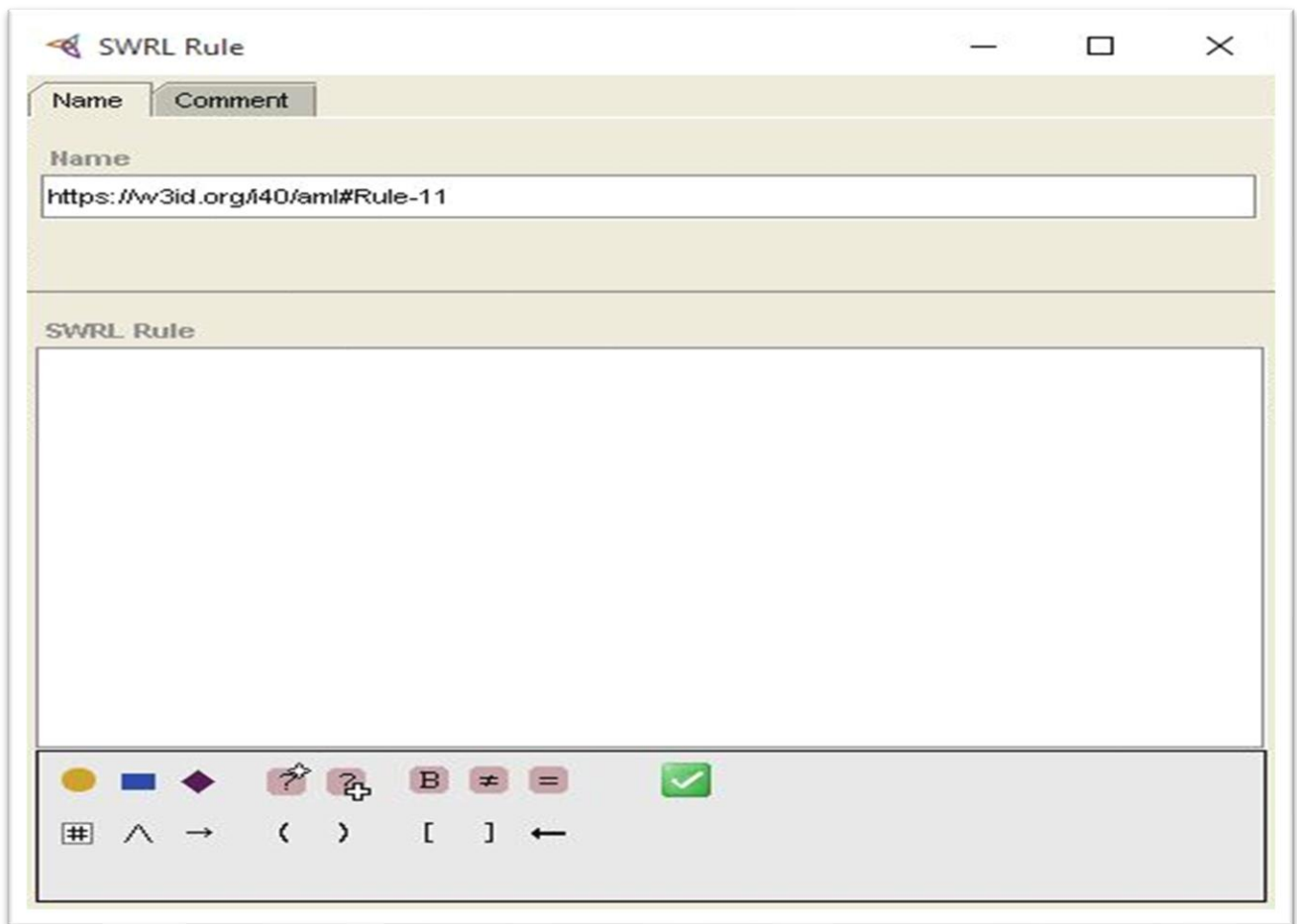


Figure5.4 : Présentation de l'éditeur SWRL

7 Quelques interfaces de notre simulateur

Dans cette partie nous allons visualiser quelques aspects relatifs à l'interface de l'application.



Figure 5.5: Interface de l'application.

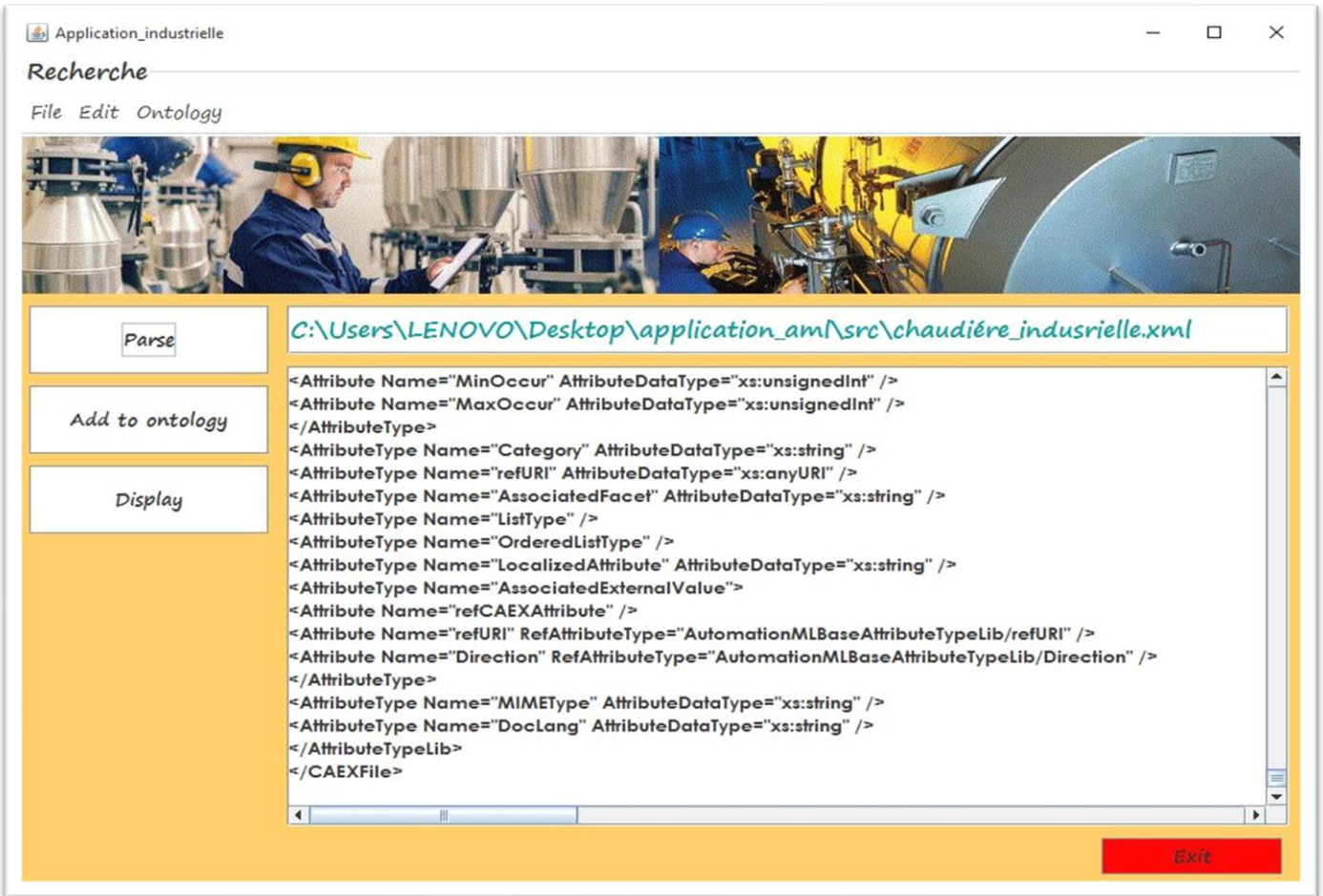


Figure 5.6: Interface Principale de la Modélisation Statique

| ID | Name | Pre | Start Time | Duration | End Time |
|----|------------------------|-------|------------|----------|----------|
| 1 | Ouverture de la va... | | 0 | 13 | 13 |
| 2 | Mise en service du ... | | 1 | 4 | 5 |
| 3 | Prise en compte de... | 2 | 5 | 4 | 9 |
| 4 | Détection flamme d... | 3 | 9 | 4 | 13 |
| 5 | Ouverture de la va... | 1,4 | 13 | 10 | 23 |
| 6 | Ouverture de la va... | 1,4,5 | 13 | 4 | 17 |
| 7 | Prise en compte de... | 6 | 17 | 2 | 19 |
| 8 | Lancement tempori... | 7 | 19 | 1 | 20 |

Figure 5.7: Interface de Création d'un diagramme de GANTT.

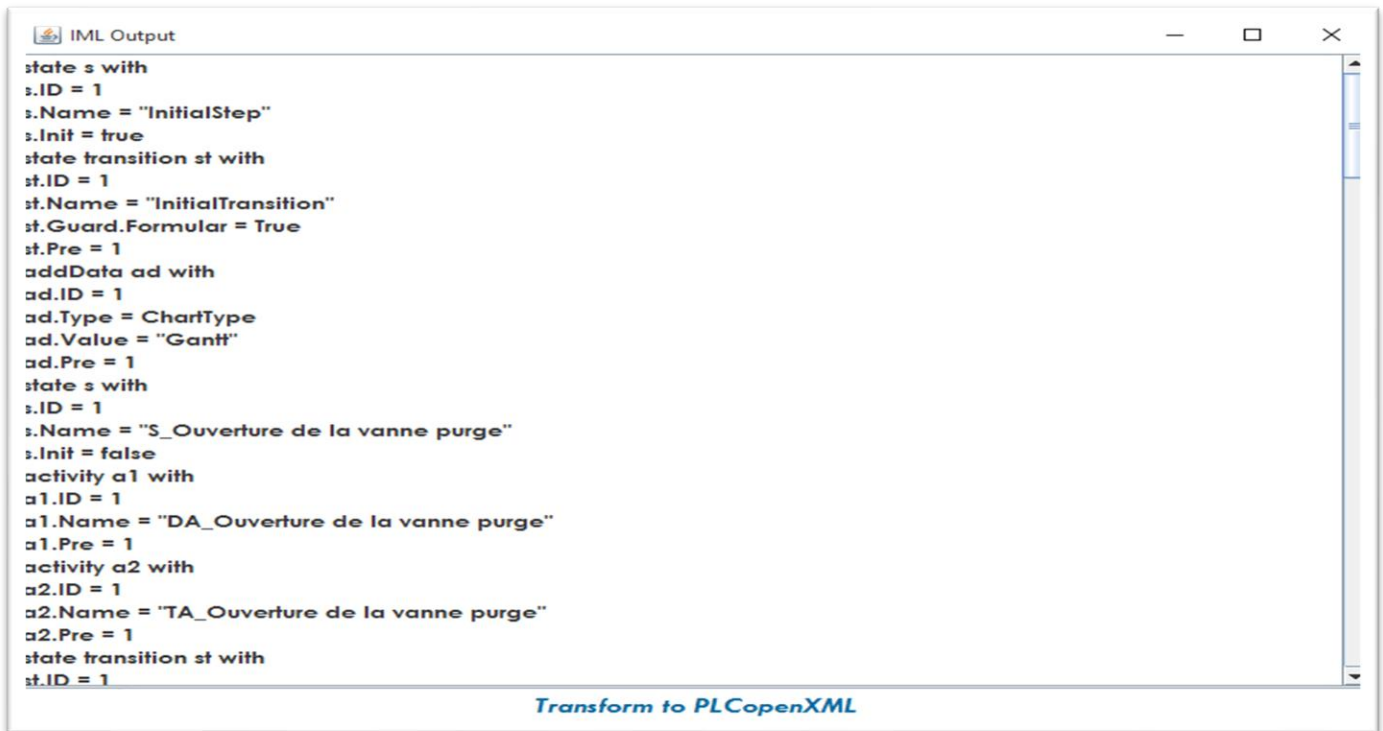


Figure 5.8: Interface de Conversation le tableau GANTT au format IML.

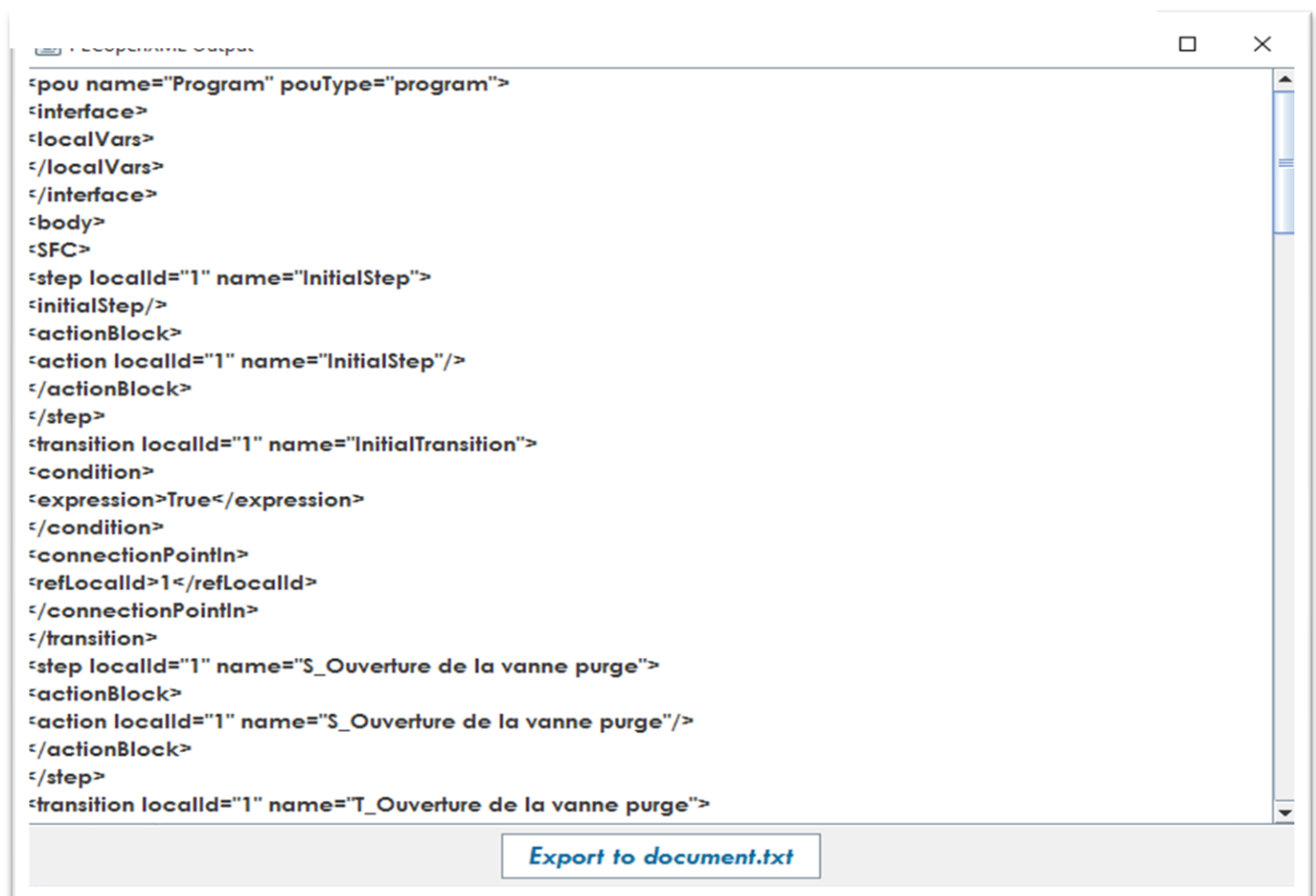


Figure 5.9: Interface de Conversation l'IML vers le format XML de PLCopen.

8 Conclusion

Ce chapitre a fourni un aperçu des outils utilisés pour mettre en œuvre notre système, ainsi que la création automatique de cette dernière à partir du document PLCOpen XML. Nous avons aussi présenté les différentes captures d'écrans montrant les différentes fenêtres de notre système. Afin de réaliser l'implémentation de notre logiciel, nous avons utilisé la plate-forme JCreator.

Conclusion et perspectives

L'Industrie 4.0, une initiative lancée par le gouvernement allemand, vise à restructurer les usines afin d'améliorer la satisfaction client, accroître la souplesse de la production et rationaliser l'utilisation des ressources. Cette évolution est justifiée par l'émergence des systèmes cyber-physiques, des données numériques et de l'interconnectivité, tout en incluant également des technologies telles que l'intelligence artificielle, l'informatique en nuage et l'Internet des objets.

La transition en cours de l'Internet traditionnel vers l'Internet des objets entraîne de profonds bouleversements dans les modèles de communication sous-jacents. L'accroissement de la complexité des installations industrielles et la diversité des parties prenantes impliquées dans le processus de construction des usines compliquent de manière croissante la communication et la vérification des modèles d'ingénierie des usines.

Afin de permettre une exploitation plus efficace, tant par les utilisateurs que par les systèmes informatiques, des vastes quantités d'informations et de services divers, le web traditionnel a évolué vers le web sémantique, offrant ainsi une représentation sémantique du contenu en ligne. L'objectif du web sémantique est d'incorporer des annotations sémantiques aux données existantes afin de faciliter l'automatisation des services, la découverte, la liaison et l'inférence des connaissances. Son architecture repose sur une hiérarchie de langages afin de modéliser des informations sur internet tout en respectant les normes de standardisation, de souplesse et d'interopérabilité.

Les ontologies jouent un rôle essentiel dans le contexte de la perspective sémantique de l'Internet des objets. En réalité, les ontologies facilitent la description des objets, de leurs attributs, de leurs capteurs et de leurs actionneurs en utilisant des terminologies standardisées, ce qui contribue à atténuer les effets de l'hétérogénéité. L'objectif de cette recherche est de développer une ontologie OWL riche et expressive en utilisant les données extraites des documents CAEX, PLCopen XML et des diagrammes de Gantt. Des algorithmes détaillés ont été élaborés afin d'assurer la correspondance entre ces catégories de documents et les ontologies OWL. L'ontologie supérieure de l'industrie 4.0 AML a été élargie afin d'intégrer des concepts spécifiques liés à la chaudière industrielle.

Dans une optique prospective, nous considérons que ce genre d'architecture se révèle très adaptable, étant donné que les technologies de l'IoT sont en constante évolution, tandis que les méthodes de représentation des connaissances et de raisonnement deviennent de plus en plus sophistiquées et efficaces. Diverses améliorations peuvent être envisagées pour le système, notamment :

- La mise en œuvre des blocs fonctionnels conformément à la norme IEC 61131-3 est nécessaire pour gérer les données de diagnostic du système de régulation de la chaudière. Cette tâche nécessite une collaboration avec des spécialistes des domaines de l'automatique et de l'électronique.
- Automatisation de la génération du code de diagnostic pour les architectures de contrôle décentralisées basées sur des PLC.

Bibliographie

- AMOKRANE Ahlem, & MERSEL Melissa. (2019). Mise en place d'une solution d'industrie 4.0 Cas d'étude : Gestion de stock et RFID. Université A/Mira de Béjaia.
- Baghdadli Chahrazed, & Ferouani Zineb. (2015). Développement d'une application déployée sur un réseau de capteurs sans fils supportant 6Lowpan. Université Abou Bakr Belkaid–Tlemcen.
- Giusto, O., Iera, A., Morabito, G., & Atzori. (2010). The internet of things: 20th Tyrrhenian workshop on digital communications. New York, NY : Springer Science & Business Media.
- Gorecky, O., Schmitt, M., Loskyll, M., & Zühlke, D. (2014). Human-machine-interaction in the industry 4.0 era 12th IEEE international conference on industrial informatics (INDIN (pp. 289–294). Porto Alegre, Brésil.
- Grangel-González, I. (2018). A Knowledge Graph Based Integration Approach for Industry 4.0. Retrieved from <http://hss.ulb.uni-bonn.de/diss>
- Hermann, M., Pentek, T., & Otto, B. (2016). Design principles for industrie 4.0 scenarios. Communication présentée 2016 49th Hawaii international conference on system sciences (HICSS).
- Kagermann, H. (2014). Chancen von Industrie 4.0 nutzen. Bauernhansl T., ten Hompel M., Vogel-Heuser B : Industrie 0 in Produktion, Automatisierung und logistik: Springer Fachmedien, Wiesbaden.
- Malone, T. W. (1999). Is 'empowerment' just a Control, decision-making, and information technology. . . BT Technology Journal, 141–144.
- N.Jazdi. (2014). Cyber physical systems in the context of Industry 4.0, Automation, Quality and Testing, Robotics, IEEE International Conference on Industrial Informatics (ICII).
- Ragavan, V., Khamis, A., Fiorini, S., Carbonera, J. L., Alarcos, A. O., Habib, M., Goncalves, P., Li, H., & Olszewska, J. I. Ontologies for Industry 4.0. doi: 10.1017/S0000000000000000
- X. Lu, Z. Qu, Q. Li, & P. Hui. (2014). Privacy Information Security Classification Study in Internet of Things", International Conference on Identification, Information and Knowledge in the Internet of Things.
- Zuehlke, D. (2010). SmartFactory-Towards a factory-of-things. Annual Reviews in Control. [Http://Dx.Doi.Org/L0.L016/j.Arcontrol.2010.02.008](http://Dx.Doi.Org/L0.L016/j.Arcontrol.2010.02.008).
- Barry Smith, Farhad Ameri, Hyunmin Cheong, Dimitris Kiritsis, Dusan Sormaz, Chris Will, & J. Neil Otte. (2019). A first-order logic formalization of the industrial ontology foundry signature using basic formal ontology.
- Damiano Arena, Farhad Amer, & Dimitris Kiritsis. (2018). Skill modelling for digital factories (pp. 318–326).

- eClass e.V., eClass Standardized Material and Service Classification. (2016). Url: <Http://Www.Eclass.Eu/> .
- Farhad Ameri, & Debasish Dutta .An upper ontology for manufacturing service description. En 2006 (Vol. 42578, pp. 651–661).
- G. Van Heijst, Ath. Schreiber, & BJ. Wielinga. (n.d.). Using explicit ontologies in KBS.
- Gasmi Mohamed. (2017). Raisonement pour les logiques de description appliqué au web sémantique.
- Grangel-González, I. (2018a). A Knowledge Graph Based Integration Approach for Industry 4.0. <http://hss.ulb.uni-bonn.de/diss>
- Grangel-González, I. (2018b). A Knowledge Graph Based Integration Approach for Industry 4.0. <http://hss.ulb.uni-bonn.de/diss>
- I. Horrocks, PF. Patel-Schneider, H. Boley, S. Tabet, & B. Grosz. (2004). SWRL : A Semantic Web Rule Language Combining OWL and RuleML. In W3C.
- Irlán Grangel-González, Lavdim Halilaj, Steffen Lohmann, Christoph Lange, & Diego Collarana. (2016). An rdf-based approach for implementing industry 4.0 components with administration shells.
- KoreckyMasterThesisOntologiesinIndustry4.0Public. (n.d.).
- L. Amir. (2017). La Génération Automatique des Ontologies à partir des Diagrammes de classes UML.
- La Recherche Scientifique, E. DE, & Boufaïda, M. (2009).
- M. Hepp. (2008). GoodRelations: An Ontology for Describing Products and Services Offers on the Web.
- M. Uschold, & M. King. (1995). Towards a methodology for building ontologies, in Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing.
- N. Zouggar, et, B. Vallespir, & D. Chen. (2006). Enrichissement de la modélisation d'entreprise par les ontologies.
- O. Kovalenko, M. Sabou, A. Lüder, S. Bif, S. Auer, & M.-E. Vidal. (2018). AutomationML Ontology: Modeling Cyber-Physical Systems for Industry 4.0 (pp. 68–86).
- S. Lemaignan, A. Siadat, & A. Semenenko. (2006). Mason : A proposal for an ontology of manufacturing domain (pp. 195–200).
- S. Lemaignan, A. Siadat, J.-Y. Dantan, & A. Semenenko. (2006). Mason : A proposal for an ontology of manufacturing domain (pp. 195–200).
- T. R. Gruber. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition (pp. 199–220).
- Tim Berners-Lee, James Hendler, & Ora Lassila. (2001, May). The Semantic Web. [Www.Scientificamerican. Com/Article.Cfm?ArticleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2](http://Www.Scientificamerican.Com/Article.Cfm?ArticleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2).

- X.H. Wang, D.Q. Zhang, T. Gu, & H.K. Pung. (2004). Ontology based context modeling and reasoning using owl. 18–22.
- AutomationML e.V.: AutomationML . (2010, April). [Www.Automationml. Org.](http://www.automationml.org)
- Boukadoum, H., & Samia, B. (2018). Option : Génie logiciel et application avancés (GLAA) Raisonnement sémantique pour le diagnostic des systèmes de contrôle à base de .PLC
- Grangel-González, I. (2018). A Knowledge Graph Based Integration Approach for Industry 4.0. <http://hss.ulb.uni-bonn.de/diss>
- M. Schleipen, & O. Sauer. (2009). “Flexibility and interoperability face Manufacturing Execution Systems (MES). In Proceedings of 3rd International Swedish Production Symposium (SPS), Day 2 Book (pp. 139–146).
- M. Schleipen, R. Drath, & O. Sauer. (2008). The systemindependent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system. In IEEE International Symposium on Industrial Electronics – ISIE 2008 (pp. 1786–1791).
- Mit Beitragen Von Rainer Drath, Dirk Weidemann, Steffen Lips, Arndt Luder, Lorenz Hundt, Andreas Keibel, Bjorn Grimm, & Miriam Schleipen. (2010). Datenaustausch in der Anlagenplanung mit AutomationML Integration von CAEX, PLCopen XML und COLLADA, Springer Heidelberg Dordrecht (Issue ISBN 978-3-642-04673-5).
- PLCopen association : PLCopen XML. (2012). [Www.Plcopen. Org.](http://www.plcopen.org)
- R. Drath, D. Weidemann, S. Lips, L. Hundt, A. Luder, & M. Schleipen. (2010). Datenaustausch in der Anlagenplanung mit AutomationML, R. Drath, Ed. Springer.
- Schmidt, N., & Lüder, A. (2015). AutomationML in a Nutshell AutomationML eV Office.
- Sebastian Faltinski, Oliver Niggemann, Natalia Moriz, & Mankowski. (2012). AutomationML: From Data Exchange to System Planning and Simulation.
- W. Terkaj, T. Tolio, & A. Valente. (2009). Focused Flexibility in Production Systems, in Changeable and Reconfigurable Manufacturing Systems, Springer Series in Advanced Manufacturing (pp. 47–66).
- Wahlberg, D., & Yixian, Z. (2010). “Automatic generation of control code for robot function packages. Automation and Mechatronics CHALMERS UNIVERSITY OF TECHNOLOGY.
- Whitepaper AutomationML Part 4-AutomationML Logic Description. (2010). www.automationml.org
- Amira, B., & Samiya, B. (2017). Réalisé par : Encadré par : Construction d’ontologie à partir du document CAEX pour la surveillance industrielle.
- BOUNEB lobna. (2015). ’INFLUENCE DES PARAMETRES OPERATOIRES SUR LES PROPRIETES DU PEHD ET RECUPERATION DE L’ETHYLENE NON REAGI AU COMPLEXE CP2K SKIKDA.

- El. Mechhoud, & M. Rouainia. (2014). Analyse de la sûreté de fonctionnement d'un réacteur de PEHD.
- Ferrag Sihèm, & Korzet Yassine. (2013). 'Suivi de procédé de dessalement de l'eau de mer au niveau du complexe CP1K –Skikda''.
- Khadiche Manel, & Amiar Bouchra. (2018). Construction d'une ontologie dynamique à partir du document PLCopen XML dans le cadre du web sémantique des objets'.
- Abele, L., Kleinstüber, M., & Hansen, T. (2011). Resource monitoring in industrial production with knowledge-based models and rules''. In ACM.
- Amira, B., & Samiya, B. (2017). Réalisé par : Encadré par : Construction d'ontologie à partir du document CAEX pour la surveillance industrielle.
- La Recherche Scientifique, E. DE, & Boufaïda, M. (2009).
- Lisa Abele, Martin Kleinstüber, & Thorbjørn Hansen. (2011). Resource Monitoring in Industrial Production with Knowledge-Based Models and Rules.
- Zander, S., & Hua, Y. (2016). Utilizing ontological classification systems and reasoning for cyber-physical systems. In In Karlsruhe Service Summit Research Workshop.
- The Protégé Ontology Editor and Knowledge Acquisition System. (2006).
[Http://Protege.Stanford.Edu](http://Protege.Stanford.Edu)
- Bouaroudj Samiya , E. DE, & Boufaïda, M. (2009).