



UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

IA03 - INFORMATIQUE POUR LA CAPITALISATION DES  
CONNAISSANCES

# Rapport - Ontologie sur les concepts HuTech

*Brizard Clément, Le Gauche Valentin*

supervisé par  
Prof. Marie-Hélène ABEL

27 novembre 2018



## Introduction

Nous sommes tous les deux étudiants en deuxième année de Génie Informatique, et passés par le cursus Humanités et Technologie (HuTech) de l'Université de Technologie de Compiègne. Cette formation forme les étudiants ingénieurs pendant leurs trois premières années, à analyser les rapports Homme-Technologie-Société, en développant un regard à la fois informé par les sciences humaines et les sciences de l'ingénieur.

Pour ce projet d'IA03, il nous était demandé de concevoir une ontologie, qui se définit comme un moyen de formaliser les connaissances sur un sujet en représentant ses concepts et les liens qui les unissent. Nous avons donc choisi de travailler sur les concepts de sciences humaines que nous avons étudiés au long de nos trois ans d'HuTech.

D'une part, ce travail vise à permettre à des étudiants HuTech de retrouver la définition et les connaissances liées à un concept qu'ils auraient oublié. D'autre part, nous souhaitons que l'ontologie permette à un utilisateur non-initié aux concepts vus en HuTech d'en tirer une utilité. Certains concepts théoriques issus de la philosophie ou de l'histoire des techniques ont en effet déjà été opérationnalisés par les principaux professeurs d'HuTech pour être utilisés en situation d'ingénierie.

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Choix du sujet</b>	<b>3</b>
1.1 Problèmes	3
1.2 Pourquoi une ontologie	3
1.2.1 Formaliser une connaissance abstraite	3
1.2.2 Valoriser une connaissance possédée ponctuellement	3
1.2.3 Un travail de sélection	4
1.3 Public cible	4
1.4 Questions pour l'ontologie	4
1.5 Périmètre de veille	5
<b>2 Démarche suivie</b>	<b>5</b>
2.1 Démarche prévue	5
2.2 Démarche suivie	6
2.2.1 Sélection, définition et recherche des liens	6
2.2.2 Une démarche en aller-retour entre conception et implémentation	6
2.2.3 Restriction du champ de recherche	6
<b>3 Structure de l'ontologie</b>	<b>7</b>
3.1 Hiérarchie des classes	7
3.1.1 La hiérarchie	7
3.1.2 Les choix de conception	8
3.2 Hiérarchie des <i>object properties</i>	9
3.2.1 La hiérarchie	9
3.2.2 Les choix de conception	9
3.3 Hiérarchie des <i>data properties</i>	10
3.3.1 La hiérarchie	10
3.3.2 Les choix de conception	11
<b>4 Mode d'emploi</b>	<b>11</b>
4.1 Nicolas, étudiant GI non-HuTech en TN10	11
<b>Conclusion</b>	<b>17</b>

# 1 Choix du sujet

## 1.1 Problèmes

Premièrement, **en tant qu'étudiants HuTech**, nous avons déjà éprouvé à plusieurs reprises des difficultés pour **nous rappeler le sens d'un concept** ou même son existence quand l'UV a été faite il y a longtemps. Ce problème est survenu en particulier lors de la rédaction du rapport de stage cet été. Toujours concernant les étudiants HuTech, nous avons aussi remarqué la **difficulté à saisir tous les liens** existant entre les différents concepts étudiés.

Concernant un **public non-initié** aux concepts étudiés en HuTech, ou même à la réflexion développée dans ce cursus, nous notons la **difficulté à accéder à une présentation claire et synthétique** de ces concepts, et de surcroît à **comprendre comment ces notions peuvent leur être utiles** dans leur (futur-)métier d'ingénieur.

## 1.2 Pourquoi une ontologie

### 1.2.1 Formaliser une connaissance abstraite

Les concepts issus des sciences humaines sont parfois **vus comme abstraits et non opérationnalisables**. L'ontologie se présente alors comme un modèle de définition de concepts et des liens qui les unissent qui convient bien à notre sujet. Les concepts HuTech sont pour la plupart liés entre eux. Certains liens paraissent déjà aisés à formaliser, comme les imbrications et les oppositions. En revanche, il sera moins facile de formaliser la nécessité de comprendre certains concepts avant les autres.

### 1.2.2 Valoriser une connaissance possédée ponctuellement

Un étudiant HuTech a une connaissance générale voire même assez fine des concepts étudiés à la fin d'une UV mais aimerait pouvoir les réutiliser plus tard.

En faire une ontologie permettrait de **retrouver leur sens rapidement**, sans avoir à se replonger dans ses notes. Le **renvoi vers des sources** permettrait d'approfondir les notions si on le souhaite.

La capitalisation serait utile à **plusieurs publics** :

- pour nous-mêmes ;
- pour nos professeurs ;
- pour les étudiants qui nous ont précédé et ceux qui nous suivent.

### 1.2.3 Un travail de sélection

Par ailleurs, comme pour tout sujet d'ontologie, notre sujet demande un travail de sélection entre tous les concepts étudiés au cours de nos trois ans de formation. Il s'agit de choisir à la fois **les concepts principaux pour comprendre la réflexion** développée en HuTech, et les concepts intéressants pour une utilisation en contexte d'ingénierie.

*A priori*, les concepts retenus vont donc être partagés en deux classes :

- **une classe théorique** : les concepts qui permettent d'appréhender la pensée de l'Homme, de la technique et de la société développée en HuTech ;
- **une classe pratique** : les concepts qui peuvent être utilisés dans des situations rencontrées dans un contexte d'ingénierie.

**Les concepts de la classe théorique sont nécessaires pour saisir complètement l'apport de ceux de la classe pratique.** Peut-être certains concepts appartiendront-ils aux deux classes.

Les concepts que nous avons retenus, pour chacune des classes, ont été **validés par nos professeurs**, soit en les ayant opérationnalisés et formalisés pour les concepts pratiques, soit *via* des compte-rendus de cours revus par les responsables des UV pour les concepts théorique.

## 1.3 Public cible

En principe, le public adressé est triple :

- étudiants HuTech ;
- étudiants et enseignants pas forcément familiers à HuTech ;
- public intéressé par les liens entre sciences humaines et technique.

En termes d'approche et d'utilisation de l'ontologie, nous considérons les deux derniers types d'utilisateurs comme identiques.

## 1.4 Questions pour l'ontologie

Il s'agit ici de donner les principales questions auxquelles l'ontologie devra pouvoir répondre :

- quels sont les concepts principaux étudiés en HuTech ?
- quelle est la définition des différents concepts ?
- quels sont les liens entre ces concepts ?
- quelles sont les références que je peux consulter pour approfondir ma première compréhension des concepts présentés dans l'ontologie ?

## 1.5 Périmètre de veille

En HuTech, nous avons étudié des concepts dans différentes UV : histoire des techniques, épistémologie et histoire des sciences, philosophie de la connaissance, sciences du travail, analyse fonctionnelle et de la valeur, entre autres. Nous sommes certains de passer en revue ceux vus en histoire des techniques, discipline qui nous a suivis pendant nos deux premières années, avec une UV différente à chaque semestre. Les compte-rendus rédigés à chaque cours par un binôme d'élèves seront notre première ressource. Nous déciderons d'étendre ou pas notre recherche aux concepts vus dans les autres UV.

Cette première recherche devrait nous permettre d'identifier les **concepts théoriques**. Pour les **concepts opérationnels**, notre première ressource sera les fiches rédigées par les trois principaux professeurs d'HuTech. Baptisées fiches SUSHI (Sorbonne Universités Sciences Humaines pour l'Ingénieur), elles s'inscrivent dans le projet SUSHI. Lancé par ces responsables pédagogiques d'HuTech, son principal objectif est de faire connaître l'intérêt des sciences humaines telles qu'elles sont appréhendées en HuTech pour l'ingénieur, mais aussi et surtout de produire des outils de terrain pour l'ingénieur.

## 2 Démarche suivie

Nous comptons implémenter une **ontologie d'application**, dans le sens où elle sera spécifique à notre domaine. Nous commencerons par **une semi-informelle pour aller vers une formelle**.

### 2.1 Démarche prévue

La démarche prévue était la suivante :

1. lister les UV HuTech où des concepts ont été étudiés
2. lister les concepts étudiés dans ces UV
3. sélectionner les concepts importants et les placer par rapport aux autres (construction progressive de la structure de l'ontologie) :
  - (a) critère de choix :
    - soit le concept rentre dans la classe théorique et complète les concepts déjà retenus
    - soit le concept rentre dans la classe "pratique"
  - (b) trouver les liens entre les concepts retenus
4. si le concept rentre dans la classe "pratique", lui associer un type de situation concrète
5. trouver les références à associer à chaque concept

## 2.2 Démarche suivie

### 2.2.1 Sélection, définition et recherche des liens

Nous avons commencé par travailler sur les concepts vus dans les quatre UV d'Histoire des techniques. Nous notions déjà les quelques liens existant entre les différents concepts sélectionnés.

Puis, parmi les concepts que nous avons retenus, nous en avons extrait les principaux pour commencer l'ontologie, en prévoyant ajouter les autres après. L'identification et le nommage des liens ne furent pas aisés. En effet, nos concepts ne représentent pas des choses du monde réel, ce qui *a priori* implique moins de liens de subsomption et davantage de liens d'explication, des liens de sens. Pour trouver ces liens, nous avons tracé un graphe, qui mettait en évidence les relations *a priori* identifiées entre les concepts. Pour ce faire, une étape importante fut d'abord de bien définir chaque concept, pour faire apparaître les liens d'explication avec les autres. Pour les représenter, nous avons commencé par inscrire le terme qui était le moins lié aux autres, à savoir "Technique", c'est-à-dire, dont la compréhension ne nécessitait pas d'avoir compris beaucoup d'autres concepts. Nous avons construit le graphe autour de ce terme "centrifuge", qui était nécessaire pour comprendre plusieurs des autres concepts principaux. Ce travail de représentation nous a obligé à nommer les liens entre les concepts mais aussi à nommer certaines de leurs propriétés.

### 2.2.2 Une démarche en aller-retour entre conception et implémentation

À ce stade, nous aurions pu ajouter les autres concepts au graphe existant. Mais nous avons le sentiment de ne pas vraiment savoir ce que nous allions pouvoir faire avec Protégé. Nous avons donc décidé d'implémenter dans Protégé le début d'ontologie dont nous disposions. À partir de ce moment, nous avons continué à procéder par aller-retour entre Protégé et représentations des liens sur tableau blanc. Le fait de ne pas attendre d'avoir terminé l'ontologie sur papier avant de commencer à travailler dans Protégé nous a aidé à construire la suite de notre ontologie, dans le sens où les possibilités offertes par Protégé nous ont donné un cadre, qui nous a orienté, guidé. Nous savions ce que nous pouvions faire et réalisons la formalisation sur papier en conséquence. L'ontologie s'est ensuite construite chemin faisant, par aller-retour plutôt qu'en deux phases bien distinctes de formalisation puis implémentation. Le risque aurait été d'implémenter une ontologie dont la forme aurait fini par être limitante pour l'intégration du reste des connaissances.

### 2.2.3 Restriction du champ de recherche

C'est pour garder la cohérence de l'ontologie naissante, et voyant le nombre déjà important de concepts retenus en Histoire des Techniques que nous avons décidé de ne pas étendre



notre recherche à d'autres UV. Ce choix restait en accord avec l'objectif de restituer la réflexion développée en HuTech, étant donné que cette discipline a constitué le socle constant de notre formation, s'épaississant chaque semestre au fil des UV.

## 3 Structure de l'ontologie

Nous décrivons ici la structure globale de notre ontologie, c'est-à-dire la hiérarchie des classes, des *object properties* et des *data properties*.

### 3.1 Hiérarchie des classes

#### 3.1.1 La hiérarchie

**Auteur**, **ConceptHutech** et **Ressource** sont nos trois classes principales et représentent donc les concepts centraux de notre ontologie. Elles sont ensuite spécialisées en un certain nombre de sous-classes pour les besoins du sens et des objectifs de la modélisation. Nous invitons le lecteur ou la lectrice à se reporter à l'ontologie elle-même pour plus de précisions sur chacune d'entre elles ainsi que pour une navigation hypertexte entre les concepts. La hiérarchie des classes est la suivante :

- owl : thing
  - Auteur
    - ConstructCOP (constructeur de concepts opérationnels)
    - Penseur
      - Anthropologue
      - Économiste
      - Ethnologue
      - Historien
      - Informaticien
      - IngenieurAnalyseFct (ingénieur en analyse fonctionnelle)
      - Linguiste
      - Philosophe
      - Psychologue
      - Sociologue
  - ConceptHutech
    - ConceptArrierePlan (concept d'arrière-plan)
    - ConceptOp (concept opérationnel)
      - ConceptOPAnalyse (concept opérationnel en situation d'analyse)
      - ConceptOpInvent (concept opérationnel en situation d'invention)
      - ConceptOpPrise (concept opérationnel en situation de prise en main)
      - ConceptOpPrbl (concept opérationnel en situation de problématisation)

- ConceptOpResti (concept opérationnel en situation de restitution)
- Ressource
  - CompteRenduCours (compte-rendu de cours)
  - FicheSUSHI
  - Livre

Nous précisons que pour les sous-classes de `ConceptOp`, nous avons repris la classification vue en DI05, UV d'analyse fonctionnelle et de la valeur. Cette classification a été reprise par nos professeurs HuTech dans les fiches SUSHI, qui opérationnalisent certains concepts.

### 3.1.2 Les choix de conception

**Disjoindre les sous-classes de `Penseur`** Plusieurs penseurs appartenait à plusieurs sous-classes de `Penseur`. Il n'est en effet pas surprenant de voir un intellectuel être par exemple à la fois un économiste et un sociologue.

Si nous avons décidé de laisser les sous-classes non-disjointes, alors nous aurions dû placer les penseurs "multi-disciplinaires" dans chacune des sous-classes auxquelles ils appartiennent. Imaginons maintenant qu'on veuille faire pointer un concept vers un penseur, en utilisant la propriété `estPensePar` qu'on verra dans la section suivante. La question qui se pose est la suivante : vers quelle instance de du penseur doit-on faire pointer la relation ? Vers celle qui est dans telle sous-classe plutôt que dans telle autre ? Cela signifierait qu'un auteur a pensé tel concept en tant qu'économiste par exemple, et tel autre concept plutôt en tant que sociologue, ce qui semble éloigné de la réalité.

Par conséquent, nous avons choisi de placer les penseurs dans leur discipline principale, et de disjoindre les classes. Notre ontologie permet d'avoir davantage d'informations sur les penseurs (et les auteurs *a fortiori*), avec la *data property* `pourAllerPlusLoin` qu'on verra par la suite, qui fournit l'URL d'une page web complémentaire.

**Disjoindre `ConceptArrierePlan` et `ConceptOp`** Les concepts opérationnels ont tous d'abord été des concepts théoriques, c'est-à-dire ce que nous avons nommé "concepts d'arrière-plan". Nous avons hésité à placer les concepts devenus concepts opérationnels également dans la classe des concepts théoriques. Leur définition dans la classe théorique aurait ainsi été dépourvue de l'aspect opérationnel, et inversement pour leur définition dans la classe `ConceptOp`.

Un problème similaire à celui de la situation précédente se serait alors posé : si une propriété, par exemple `estComprisGraceA` qui sera vue dans la section suivante, fait référence à un concept présent dans les deux classes, quelle instance choisir ?

Nous avons donc choisi de considérer qu'un concept opérationnel apparaîtrait uniquement dans `ConceptOp`, et donc de disjoindre les deux classes.

**Pour aller plus loin** Nous n'avons pas instancié les comptes-rendus de cours, ce qui aurait nécessité de scanner une cinquantaine de pages, puis de les intégrer sur le Drive déjà créé pour héberger les fiches SUSHI.

## 3.2 Hiérarchie des *object properties*

### 3.2.1 La hiérarchie

Nous avons classé les *object properties* selon les liens qu'elles représentent. Ainsi, `caracterise` modélise le lien pouvant exister entre une ressource et un `ConceptHutech`, elle est aussi l'inverse de la propriété `estCaracterisePar`. Ses sous-propriétés décrivent les relations pouvant exister entre certaines sous-classes de ressources et certaines sous-classes de `ConceptHutech`. De nouveau, nous renvoyons vers l'ontologie elle-même pour de plus amples indications, notamment concernant les *domains* et les *ranges* des propriétés non détaillées ici, comme `aEcrit`. La hiérarchie est la suivante :

- owl : topObjectProperty
  - aEcrit
  - caracterise
    - contient
    - presente
    - resume
  - concoit
    - construit
    - pense
  - estCaracterisePar
    - estContenuDans
    - estPresenteDans
    - estResumeDans
  - estComprisGraceA
  - estConcuPar
    - estConstruitPar
    - estPensePar
  - estEcritPar
  - estNecessPourCompre (est nécessaire pour comprendre)
  - estOpposeA

### 3.2.2 Les choix de conception

**Formalisation des liens d'explication avec `estComprisGraceA` et `estNecessPourCompre`**

Une des premières décisions a été de statuer sur la manière de formaliser les liens d'explication entre les concepts, autrement dit le fait qu'un concept en explique un autre. Ce qui

implique que ce concept doit avoir été vu avant de voir l'autre.

Nous avons opté pour deux propriétés inverses l'une de l'autre : `estComprisGraceA` et `estNecessPourCompre`. Ils permettent une circulation à double-sens entre les concepts :

- depuis un concept que je n'ai pas compris, j'ai accès à ceux que je dois avoir compris avant
- depuis un concept que j'ai compris, j'ai accès à ceux que je peux maintenant aborder

**Instanciation de toutes les propriétés inverses** Nous avons été surpris de constater que protégé n'instanciait pas les *object properties* à partir des instances connues de leurs propriétés inverses. Par exemple, si nous avons déclaré qu'un livre a été `ecritPar` un `Auteur`, nous nous attendions à ce que Protégé infère et ajoute que l'auteur en question `aEcrit` le livre en question.

Constatant que ce n'était pas le cas, nous avons tout de même pu vérifier que le raisonneur réussissait à inférer les propriétés inverses, la seule déception étant que ces inférences n'étaient pas conservées une fois le raisonneur stoppé. Par ailleurs, en testant quelques requêtes dans les DL *queries*, nous avons pu voir que nous pouvions interroger les propriétés inverses non instanciées.

Donc *a priori*, nous pouvions nous passer de les instancier. Nous avons néanmoins finalement choisi de le faire. Car si le fait de ne pas les instancier n'est pas handicapant en situation de requête, il l'est lorsqu'on navigue entre les concepts, sans avoir lancé le raisonneur. Et si quand on visualise par exemple les informations d'un livre, on peut savoir par qui il a été écrit, nous ne voyions pas de raison qu'on ne puisse pas réciproquement savoir quel livre a écrit un auteur. Nous avons donc systématiquement instancié les propriétés inverses.

### 3.3 Hiérarchie des *data properties*

#### 3.3.1 La hiérarchie

Nous avons rassemblé les *data properties* selon leur sens et les liens qu'elles modélisent. Notamment, nous avons regroupé en tant que sous-classes de `estUtilisePour` les propriétés liant un concept opérationnel utilisable dans un certain contexte d'ingénierie – contexte d'invention, d'analyse, de restitution, etc. – à un but précis dans ce contexte. La hiérarchie est la suivante :

- owl : topDataProperty
  - aPourObjectif
  - aPourTitre
  - estDefiniPar
  - estLieA
    - estAccessibleA

- pourAllerPlusLoin
- estUtilisePour
- estUtilisePourAnalyse
- estUtilisePourInvention
- estUtilisePourPrise
- estUtilisePourProbl
- estUtilisePourResti
- paruEn

### 3.3.2 Les choix de conception

**Ne pas regrouper les propriétés propres à un titre** Pour une meilleure lisibilité des hiérarchies de propriétés, nous avons, autant que possible, essayer de créer de regrouper les propriétés dans des super-propriétés. Nous avons été tentés de le faire avec `aPourTitre` et `paruEn` dont le *domain* est uniquement un livre, mais nous sommes finalement ravisés. En effet, nous aurions dû appeler cette super-propriété "propriétés livre", ce qui aurait plutôt été un agrégat logique et non une super-propriété au sens strict, qui généralise ses sous-propriétés.

**Pour aller plus loin** Il nous resterait à instancier complètement les sous-propriétés de `estUtilisePour`. De même avec `aPourObjectif`. Ces informations sont certes présentes dans les fiches SUSHI, que nous avons rendues accessibles avec le lien Drive associé à `estAccessibleA`, mais nous jugeons intéressants pour l'utilisateur d'avoir déjà un aperçu dans l'ontologie de ce à quoi peut lui servir un concept selon le concept (`estUtilisePour`) et en général (`aPourObjectif`).

Pareillement, nous n'avons pas eu le temps d'instancier la propriété `estIllustrePar`, qui devait associer à chaque `ConceptHutech` un exemple l'illustrant.

## 4 Mode d'emploi

Pour illustrer l'utilisation de notre ontologie nous allons décrire un cas d'application concrète qui répond aux objectifs que nous avons fixés au début de ce document.

### 4.1 Nicolas, étudiant GI non-HuTech en TN10

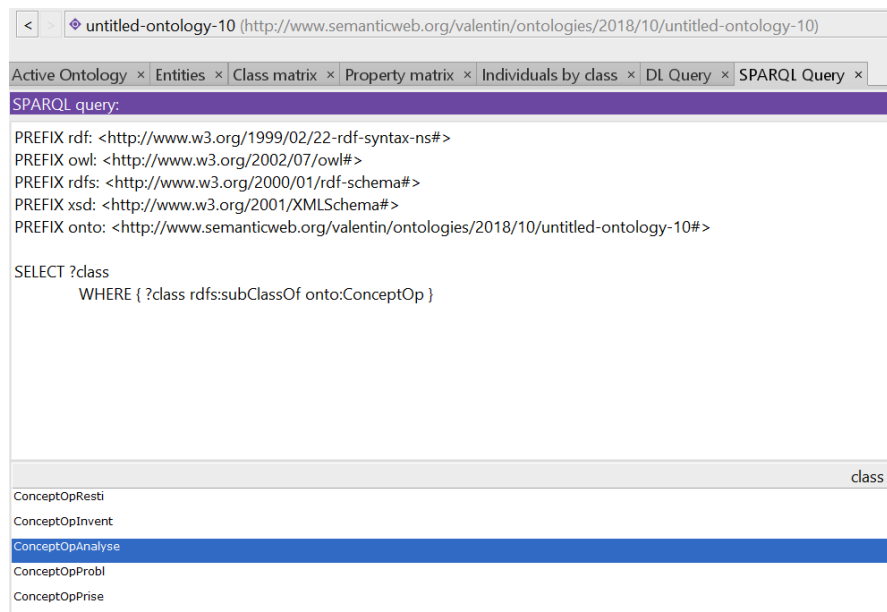
Nicolas est un étudiant en TN10 dans un entreprise d'informatique fabriquant et vendant des applications web en *business-to-business*. Après s'être familiarisé avec l'entreprise, ses manières de travailler et les technologies utilisées, on lui demande de participer à la mise en place d'un nouveau *framework* destiné à standardiser et simplifier la production des

applications pour plus de compétitivité. Il sait que l'introduction de ce nouvel outil de travail changera grandement la manière de travailler des développeurs.

Aujourd'hui, ce sont eux qui possèdent les savoir-faire légitimes, à la fois individuellement et collectivement. Ils les ont acquis sur le long terme en se confrontant à une matière qui résiste, seul ou en s'entraînant : ils se sont construits grâce et par eux. On peut dire qu'ils ont créé une technicité d'équipe "par le bas" : elle a émergé des développeurs et de leurs pratiques individuelles et collectives. Changer leurs outils de travail changera l'ensemble de ces données.

Nicolas sait qu'une ontologie des concepts HuTech est disponible et qu'elle contient des concepts opérationnels et mobilisables directement en contexte d'ingénierie.

Grâce à l'onglet SPARQL de Protégé, Nicolas peut déterminer les classes regroupant les concepts opérationnels utilisables dans les différentes phases d'un projet d'ingénierie.



**FIGURE 1:** Requête SPARQL pour déterminer les différents types de situation pour mobiliser les concepts opérationnels

Après avoir obtenu le résultat de sa requête, il se rend sur la vue détaillant les instances des concepts opérationnels utilisables en analyse, car il est au début de son projet et tente de *voir ce qui est*.

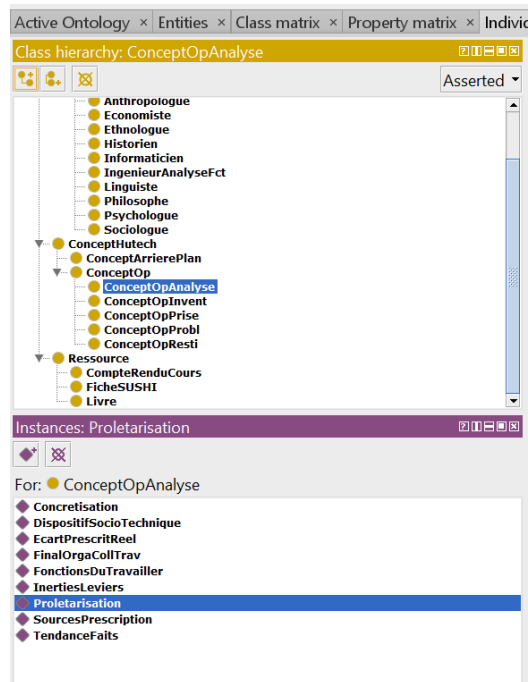


FIGURE 2: La classe `ConceptOpAnalyse` et ses instances

Après quelques tâtonnements sur certains concepts, à regarder leur définition afin de voir s'ils peuvent convenir à son projet, il s'arrête sur celui de "Prolétarisation", qui est adéquat pour anticiper les dépossessions de capacités lors de changements dans l'organisation de travail. Il peut voir dans quel but précis il peut être utilisé grâce à la propriété `estUtilisePourAnalyse`.

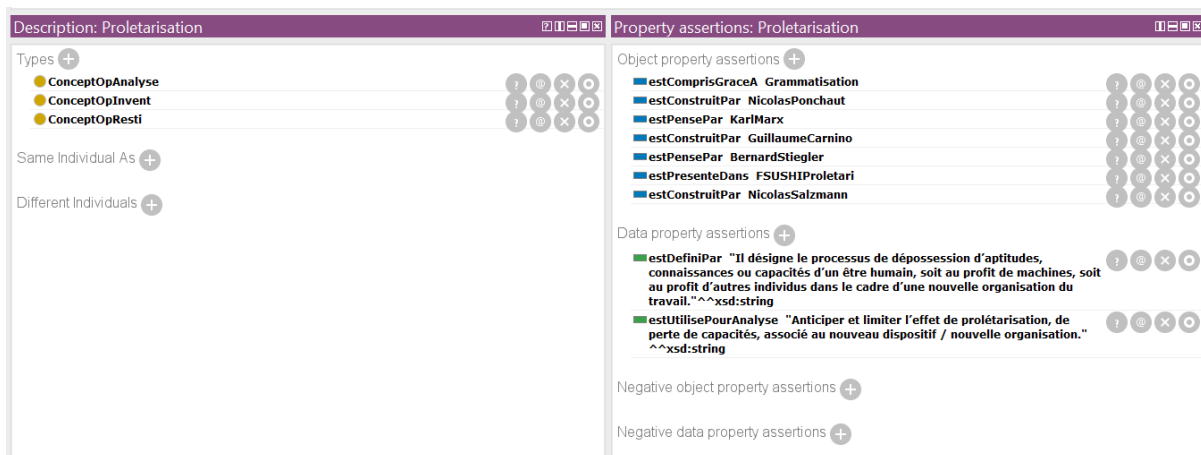


FIGURE 3: Le concept de prolétarisation

Pour en savoir plus, il clique sur `FSUSHIProletari` qui le conduit vers la vue de la fiche SUSHI qui présente le concept de prolétarisation – grâce à la propriété `estPresentePar`.

De là, il pourra obtenir un lien URL pour accéder à une version PDF de la fiche et trouver des informations plus détaillées sur l'objectif du concept opérationnel, ses types précis d'utilisation et les formalismes à adopter.



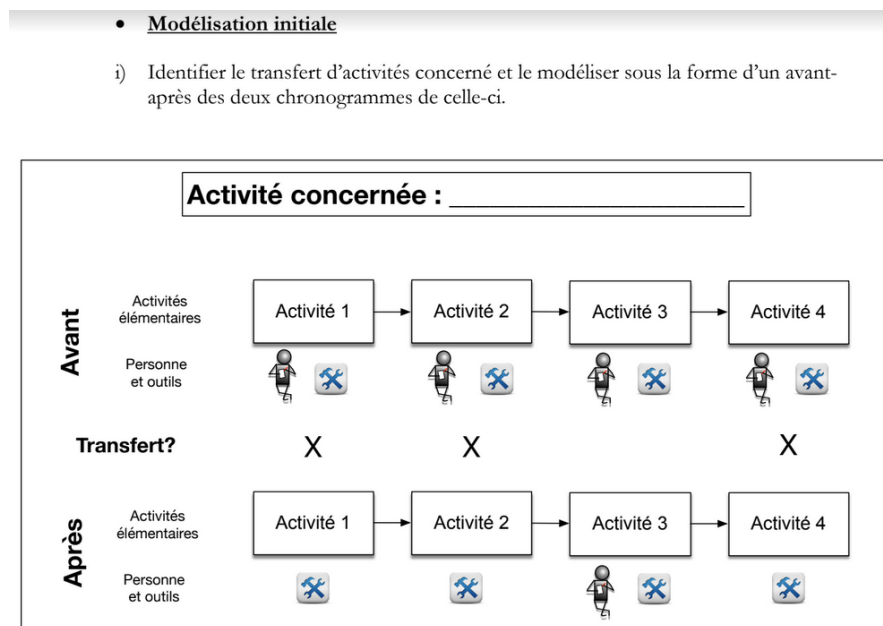
Projet SUSHI

## Prolétarisation

### 1. Origine et définition du concept

Le concept de prolétarisation a été formulé par K. Marx, dans le cas précis de la prolétarisation des ouvriers dans l'organisation capitaliste. Il a ensuite été repris et développé par B. Stiegler, dans une définition plus élargie, parlant ainsi par exemple de la prolétarisation des ingénieurs. Il désigne le processus de dépossession d'aptitudes, connaissances ou capacités (savoirs, savoir-faire, savoir-être, etc.) d'un être humain, soit au profit de machines, au sein desquelles elles seront dès lors intégrées, engrammées, soit au profit d'autres individus dans le cadre d'une nouvelle organisation du travail (perte de la maîtrise globale d'une activité et de son sens). Ce processus concerne potentiellement toutes les activités, manuelles ou intellectuelles.

**FIGURE 4:** Introduction de la fiche SUSHI du concept de prolétarisation



**FIGURE 5:** Formalisme d'utilisation du concept de prolétarisation, disponible dans la fiche SUSHI

En revenant à l'éditeur Protégé, sur la vue du concept de Prolétarisation, plusieurs choix s'offrent à lui. Il peut voir qui l'a construit – Guillaume Carnino, par exemple, grâce à



la propriété `estContruitPar`. Il peut aussi accéder à la description de Bernard Stiegler, penseur du concept de Prolétarisation – grâce à la propriété `estPensePar` – dont il vient de voir le nom sur la fiche SUSHI. De là, il pourra observer que cet auteur est le penseur d'autres concepts, notamment celui de Grammatisation.

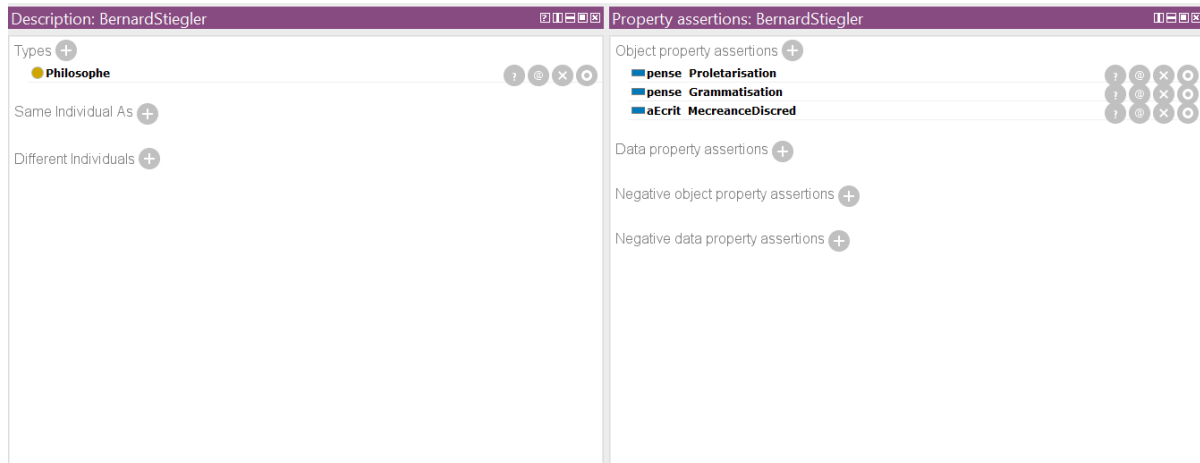


FIGURE 6: Vue du concept représentant le philosophe Bernard Stiegler

Il décide de retourner sur la vue du concept de Prolétarisation pour se recentrer sur ses recherches initiales. Il aimerait en connaître davantage sur ce concept et choisit d'aller sur la vue du livre dont il est extrait, *Mécreance et Discrédit 1*, écrit par Bernard Stiegler – grâce à la propriété `estContenuPar`. De là, il pourra avoir accès un lien, grâce à la propriété `estAccessibleA`, à la page web de la maison d'édition dédiée au livre.

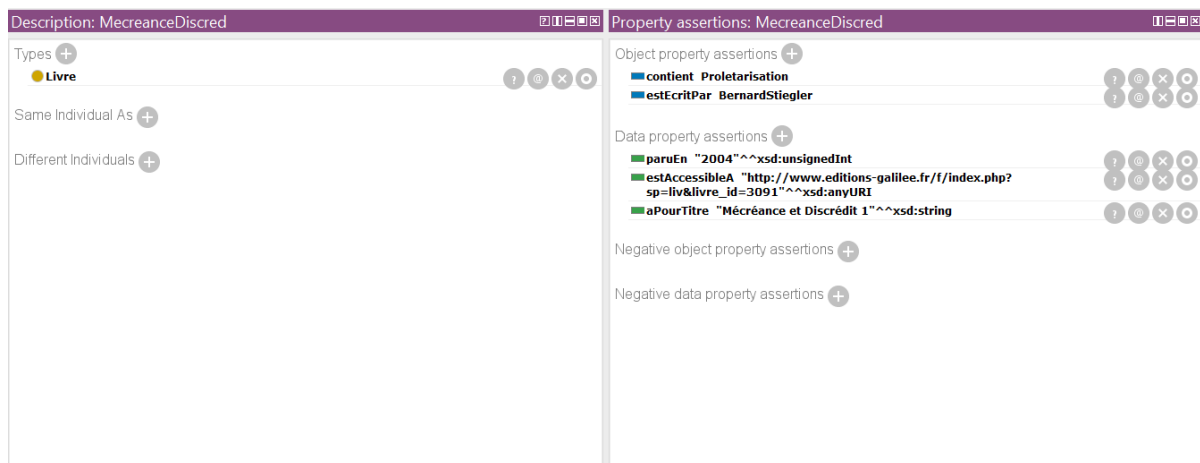


FIGURE 7: Vue du concept représentant le livre *Mécreance et discrédit 1*

Suite à cette précision obtenue, il décide de chercher d'autres informations car il perçoit que quelque chose lui manque pour avoir une pleine compréhension du concept de Prolétarisation. C'est alors qu'il aperçoit la propriété `estComprisGraceA` de Prolétarisation

qui pointe vers Grammatisation, qu'il a déjà vu chez Stiegler. De cette vue, il pourra aller vers le concept d'industrialisation *via* la propriété `estComprisGraceA`.

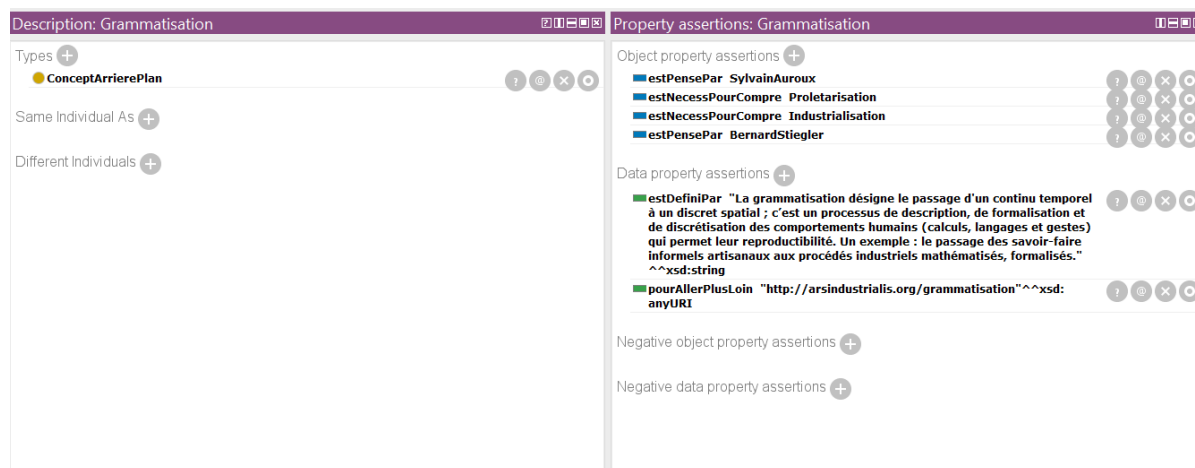


FIGURE 8: Vue du concept de grammatisation

Arrivé à la fin – provisoire – de son périple, Nicolas e acquis un certain nombre d'informations contenues et référencées dans l'ontologie qui lui permettront de bâtir des connaissances qu'il sera à même d'utiliser, de mettre à profit lors de son projet.

On peut aussi voir qu'il est logique qu'il arrive sur le concept d'industrialisation puisque celle-ci désigne historiquement le passage de savoir-faire artisanaux informels à des procédés scientifiques rationalisés et mathématisés. Ils ont été peu à peu discrétisés, systématisés, mis en écriture, mis en gramme, grammatisés puis incorporés dans des machines qui forment le royaume des ingénieurs. Il y a bien eu dépossession et transfert de savoir-faire – des artisans au profit des ingénieurs – et donc prolétarianisation. C'est un raisonnement, une connaissance de ce type que nous avons implémentée dans notre ontologie. Il est possible que ce même processus se rejoue aujourd'hui dans des cas comme celui de Nicolas. C'est pour cela qu'une ontologie telle que la nôtre peut être un bon support à la capitalisation et la gestion de la connaissance pour une entreprise. Espérons qu'elle puisse être un outil au service de l'ingénieur.

## Conclusion

Nous avons éprouvé quelques difficultés pour modéliser les liens de subsomption qui peuplent maintenant notre ontologie. Nous y sommes finalement parvenus en même temps que d'avoir réussi à implémenter des liens d'explication et de raisonnement, comme le montre notre étude de cas. Notre ontologie est cohérente, l'utilisateur peut y naviguer de nœud en nœud selon ce qu'il souhaite trouver ou explorer. Elle est aussi adaptée aux différents publics que nous visions. Une personne ne connaissant pas HuTech et souhaitant connaître la pensée qu'on y enseigne et l'action qui lui est inhérente pourra circuler dans l'ontologie pour comprendre les liens entre les concepts et avoir accès à des ressources pour aller plus loin. Un ancien étudiant du cursus pour facilement retrouver ce qui lui échappe grâce à une ou deux requêtes SPARQL. Un ingénieur en activité sera à même d'identifier et sélectionner les concepts opérationnels adéquats à son travail et à ses projets.

Pour finir, il nous semble essentiel de souligner que l'ontologie ne se suffit pas à elle-même, mais qu'elle est avant tout un support à la connaissance et au raisonnement, qu'elle ne saurait se substituer à un apprentissage et une acclimatation sur un temps long. Elle est un pont vers d'autres ressources tout en constituant une base, un point de départ.