# A semantic framework for enabling model integration for biorefining

Linsey Koo[1], Nikolaos Trokanas[2], Franjo Cecelja[1]

[1] *Centre for process & Information Systems Engineering, University of Surrey, Guildford GU2 7XH, UK*

[2] *Centre for International Manufacturing, Institute for Manufacturing (IfM), University of Cambridge, 17 Charles Babbage Road, Cambridge, CB3 0FS, UK*

**Abstract**

This paper introduces a new paradigm for establishing a framework that enables interoperability between process models and datasets using ontology engineering. Semantics are used to model the knowledge in the domain of biorefining including both tacit and explicit knowledge, which supports registration and instantiation of the models and datasets. Semantic algorithms allow the formation of model integration through input/output matching based on semantic relevance between the models and datasets. In addition, partial matching is employed to facilitate flexibility to broaden the horizon to find opportunities in identifying an appropriate model and/or dataset. The proposed algorithm is implemented as a web service and demonstrated using a case study.

**Keywords**

Ontology engineering, Model Integration, Computer Aided Process Engineering (CAPE), Biorefining

## 1. Introduction

In computer aided process engineering (CAPE) community, increased availability of mainstream commercial and free simulation software, as well as data from laboratory experiments or pilots to near commercial scale plants, has facilitated the development of a large number of custom-made models. As, historically, most of the models were developed to represent petrochemical processes, modelling and simulation for biorefining processes are still facing challenges due to lack of biochemical property data, complexity of feedstock characterisation, as well as a constant influx of new processes and technologies or adaptation to new environments. To develop an understanding of biochemical processes or to provide suitable design, development of a database system to support modelling and analysis of biochemical processes is vital. The development of these models, as practice has demonstrated, goes along the development of new models, integration and/or adaptation of existing models, or most commonly the combination of the two.

To increase reusability of existing models that are developed in disparate software tools and process simulators, CAPE-OPEN was initiated to conceptualise and develop a set of interface specifications as a method pertaining interoperability standard (Braunschweig et al. 2004; Morales-Rodríguez et al. 2008; Pons 2010). As such, CAPE-OPEN is a widely recognised standard which defines the interconnection representation of interfaces facilitated by a middleware service as a communication hub across heterogeneous software environments (Braunschweig et al. 2000; Bogusch et al. 2000). To take full advantage of reusability of existing models, the task of identifying the most sufficient model from the libraries is heavily dependent on the user's intuition and experience and remains as a manual process (Braunschweig et al. 2004). Yang et al. (2008) acknowledges that inadequate assessment for

the suitability of models may lead to potential misuse of the models, which has the risk of insufficient or even wrong solution to the engineering problems. To better address the shortcoming associated with user intervention in CAPE-OPEN, ontology engineering is recognised as a viable solution to reduce the chance of these error occurring and to minimise the impact of any errors that do occur. Ontology has an ability to address the problem of automated support for the configuration of process models and data in a structured and proactive manner (Yang & Marquardt 2004; Yang et al. 2008) by accounting for complex relations, such as systematic knowledge of model as well as tacit knowledge extracted from user intuition. A large scale ontology, the OntoCAPE, has as a result been introduced to support various process engineering applications, mainly addressing two aspects: i) characterisation of models stored in the libraries and ii) description of the specific requirements of the models to be identified as potential candidates. To address the reconciliation of interoperability between process modelling components, COGents was developed to perform the registration and integration of the models stored in the libraries. This method was the first attempt to integrate process modelling components from heterogeneous sources using ontologies as a tool in the field of process engineering. As indicated by Yang et al. (2008), the integration of the models they used was based on the full-scale matching. Partial matching which extends the search scope was first introduced by the eSymbiosis project to enable and hence to support processing technologies participation in Industrial Symbiosis (IS) and concomitant integration (Raafat et al. 2012; Raafat et al. 2013; Cecelja et al. 2015). The framework employed semantic technologies to automate widely used manual procedure of synergy identification of IS by finding the semantic relevance of participant's profile based on practical experience in the form of tacit knowledge and explicit knowledge acquired from users. The measure of semantic relevance requires obtaining appropriate description of processing technology to further use in the discovery process. To support these processes, the process of IS was semantically formulated in an IS domain ontology (Trokanas et al. 2012). Recently, a number of ontologies have been developed in the domain of biorefining, which focuses on the knowledge representation of biomass and bioprocessing technologies (Trokanas, Bussemaker, et al. 2015) and process systems design and optimisation of biorefining processes (Siougkrou & Kokossis 2016; Magioglou et al. 2015). These ontologies, however, although in the domain of biorefining do not address the process of model and data integration, and, to the best of our knowledge, they are not yet available in public domain for reuse.

Following on previous developments and use of ontology to address challenges of identifying most suitable model or data to achieve the best solution for a particular engineering problem, ontology engineering is employed to describe them in a comprehensive manner to distinguish between them (Koo & Cecelja 2015; Koo et al. 2016). It has been demonstrated that the differences of models and data can be addressed by explicit descriptions using defined terms to further improve consistency as well as understanding of the heterogeneity and concomitant consequences. The semantically enriched and reconciled process models and data are then applicable to facilitate semantic interoperability between them. The semantic interoperability is achieved by employing different matchmaking algorithms to benefit from partial matching to measure a meaningful similarity between models that are not identical. We argue that this approach allows to improve the decision making process and broaden the horizon to find opportunities in identifying appropriate models and/or datasets whilst increasing awareness of existing models.

This paper proposes a new paradigm for model and data integration with focus on biorefining and which is built around the ontology to i) model tacit knowledge in the domain of biorefining including the advances in biorefining process, biomaterial and technologies classifications, and ii) model explicit knowledge which includes a complete set of model input, output and auxiliary parameter properties, as well as known and otherwise identified potential model and data integration solutions. Tacit

knowledge is built in the ontology structure (Cecelja et al. 2015), i.e. subsumption and object properties with respective and domain dictated restrictions. Explicit knowledge is captured during the instantiation process from data collected on model/data entities presented as ontology instances and characterised by input, output and auxiliary parameter properties. The proposed ontology enables instance matching with the view of model integration, expanding knowledge base, generating new knowledge in the process of model integration for biorefining, and knowledge sharing. Designed ontology is open to further development in response to advances in the domain of biorefining. The proposed matching algorithm is tuned to match models and data based on i) tacit knowledge formulation to observe process synthesis logic by employing semantic distance measurements between the two or more instances of the ontology, and ii) explicit knowledge formulation by employing similarity calculation between input/output parameters of candidate models/data identified suitable for integration. In addition, matching process allows for recursive matching towards complex model/data integration solutions, matching for integration of models developed in heterogeneous software environments to generate a meaningful solution for particular engineering tasks, as well as for partial matching to broaden the search domain and to find comparable replacement model rather than focusing only on an exact match. This paper explicitly formulates theoretical concept of knowledge model and design of ontology and matching algorithm, as well as auxiliary conditions used in the process of model/data integration. The usefulness and operation of the proposed formalism is demonstrated by a case study to guide the user to make an informed decision by taking into consideration of users' intuition and their experience in modelling.

## 2. Theoretical Concepts of Model and Data Integration

### 2.1. Model and Data Representation

A process model represents a part of the actual system in which physical and chemical processes are taking place and describes the behaviour of a process system within well-defined boundaries together with inputs and outputs and under certain environmental conditions as a requirement (Hangos & Cameron 2001). The process models used to address process modelling, simulation and optimisation problems are arguably classified into two distinct types i) sequential modular models, and ii) equation based models. Sequential modular models represent individual units as a pre-configured block model where modelling equations are grouped to represent a particular process equipment. The sequence of calculation is initiated from one unit to the next in the process flowsheet through the process streams that connect the units using thermodynamics and physical property calculations. Equation based models are considered as custom modelling packages which have a set of equations from the various units in the process into a single large set to be solved.

Each model is semantically described by its type, i.e. its functionality in terms of the process and/or unit it represents. In addition, each model is (semantically) described by requirements and other characteristics that form a comprehensive knowledge model (Koo et al. (2016)) which includes model input(s), output(s), precondition(s), and the environment in which each process model operates (Trokanas et al. 2014; Trokanas, Cecelja, et al. 2015). The inputs and outputs are not limited to physical properties and can be extended to additional data or other properties. The number of output variables can be purposely adjusted or extended to include additional data or parameters to consider the dynamic nature of models. Contrary to the models, data is semantically annotated with regards to output(s), functionality, and precondition(s) required to process data (Koo et al. 2016).

## 2.2. Concept of Model Integration

The integration of model and data is a process of assembling heterogeneous tools and methods to generate new knowledge that is meaningful and useful for particular engineering tasks. The CAPE-OPEN interface specification (Belaud & Pons 2002) is developed as a standard requirement for the unit operation components (such as process unit operation, thermodynamics, and numerical solvers packages) to be compliant with any simulator without modification, compiling, or linking. The standard mainly provides the details for the interface specifications of sequential modular simulators and the granularity of the interface design was restricted to the unit operation level (CAPE-OPEN Project team 2000; van Baten & Pons 2014).

The structure of unit is configured by a coupling through the different inlet and outlet ports where a unit can be connected to another unit, which is separated from the functional behaviour of the unit model (Figure 1). To achieve consistency across simulation platforms, the unit operation components are represented as a template that allows access to the stream and provides unit operation data of a flowsheet in a conceptual manner. A set of data to be exchanged from one unit to another is distinguished by three different types: material, energy, and information streams. In addition, the stream properties are characterised by thermodynamics and physical properties, e.g. composition, temperature, pressure, flow, etc., with associated variables describing quantitative properties, e.g. physical dimension, value, unit, etc.

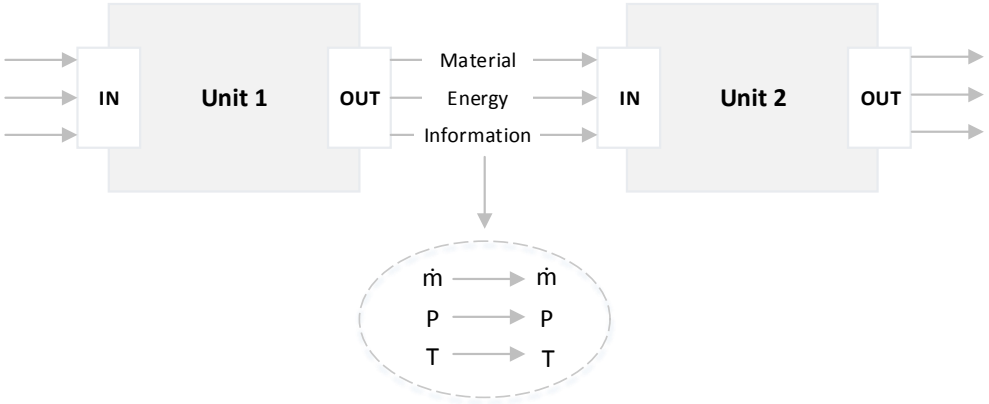
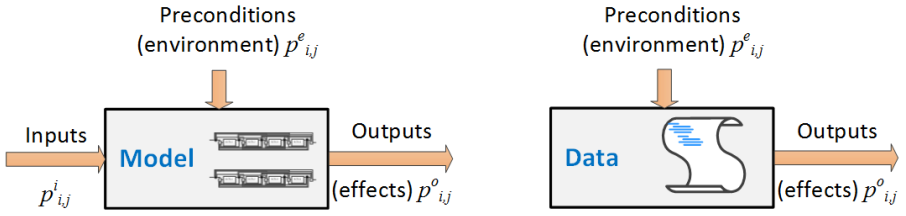



Figure 1 Conceptual representation of unit configuration via ports and internal connection of material stream

The internal connection between unit operations is further characterised by the association of physical and/or thermodynamic properties with streams. The CAPE-OPEN has established the specification for the thermodynamic and physical properties of materials that are processed in a unit operation to encapsulate interchangeable concepts. The physical objects are represented as abstract material properties for both mixture and pure components, together with state of the physical object, e.g. temperature, pressure, enthalpy, volume, vapour fraction etc., and phase for which the property calculation is required. The CAPE-OPEN additionally provides lists for constant and non-constant properties and lists of single and multi-phase properties including units and its conversion factor to SI units.

We argue that model interoperability could be assessed and established in more flexible manner and hence propose the conceptual representation of model interoperability established by CAPE-OPEN to be translated into ontology. In turn and with the focus on biorefining, such an approach enables the

development of a knowledge based platform for model and data integration through input/output matching. A higher degree of flexibility is achieved by introduction of a partial matching technique where the 'equality requirements' between input/output set of exchanged data is replaced by the 'similarity requirements'. In consequence, we propose a framework where models are considered at a superstructure level and hence assumed to be a piece of software representing a (biorefining) unit or a process and which, using mathematical or otherwise algorithms, converts its input parameter(s) into one or more output parameters, all within certain environment. The inputs to the model are not necessarily limited to physical parameters associated with the inputs to the process or unit they represent; the number and type of inputs to the model is normally extended to additional data and/or parameters the model needs to perform properly. By the same token, the number and type of model outputs could be deliberately or accidently extended to additional data or parameters the model provides and which could be useful to other models or purposes. Models are normally described and identified by their functionality of the processes or units they represent, but also further provide the association to respective synthesis problem. Also, to run a model, it requires certain preconditions, i.e. particular application such as MatLab, GAMS, MS Excel, or synchronisation with other models. Both of these two aspects form the environment in which a model runs and by which it is also distinguished from other models (Figure 2a).




**a) Models** **b) Data**

Figure 2 Representation of models and data

In contrast to models, data are contained in datasets external to models, and in relation to the execution of a model represent static values of process or unit parameters, characterised again as outputs. Dynamic aspects of data associated with its generation, modification and/or deletion will not be considered here. Data are also described by their 'functionality' and normally stored in datasets (databases) for which certain conditions should be provided to access and retrieve hence forming environmental conditions (Figure 2b).

For the reason of consistency and uniformity, the model inputs and the model and data outputs and preconditions are characterised by respective input properties $p^i_{i,j}$, output properties $p^o_{i,j}$, as well as precondition or environment properties $p^e_{i,j}$, as shown in Figure 2. For the reason of clarity and the demonstration of the framework, the structure of the properties $p^i_{i,j}$, $p^o_{i,j}$ and $p^e_{i,j}$ is assumed to be in the form of a single numerical or descriptive property. Later in this paper, this will be extended to composite format(s) with properties forming property-subproperty subsumption relation, as explained in Section 2.4.

The key to model (and data) integration is the model/data semantic annotation, discovery of candidate models and data which fully or partially satisfy matching conditions and ranking them by the level of match. To this end, the model and data matching process refers to the process of comparing requesting model inputs with other model or data outputs. Practice suggests that a full matching between models is rare and hence some adaptations and/or compromises are needed, a

process we term as *partial matching* process. In addition to matching the model and data functionality, which will be explained in details in Section 2.3, the model/data matching entails matching between input properties $p^{i}_{1,j}$ of requesting model $x_1$ and output properties $p^{o}_{i,j}$ of candidate model(s) or dataset(s) $y_2$, which we term as the i*nput/output* matching, as shown for a single matching between only two models in Figure 3. The input/output matching is performed for each of $n_I$ input properties $p^{i}_{i,j}$ separately and in turn as matched properties might be from different models/datasets. Also, it is assumed that requesting model is the last in the chain, hence *backward matching* applies. The properties used in matching are either descriptive, i.e. material type, numerical, i.e. flow rate, or even composite, i.e. range of flow rate with minimum and maximum values. Still, the level of match is expected to be quantified by a single value for easier comprehension by humans and further processing merely by decision support agents.

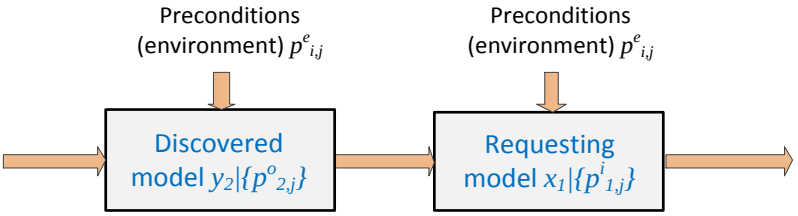



Figure 3 Principle of single model input/output matching

For more complex integration which involves more than two models and/or datasets, complex chains are formed by recursively repeating the single matching process with each of the candidate models, $y_2$ in Figure 3 switching the role to the requesting model, which would be replaced as $x_2$ in Figure 4, and which is then seeking for new matches. More complex chains, such as many-to-many chains, are also possible; this process involves either i) matching different input parameters of requesting model with outputs of different models or datasets, or ii) property decomposition, both of which is the subject of an on-going work and further publications. The combination of the two is also possible.

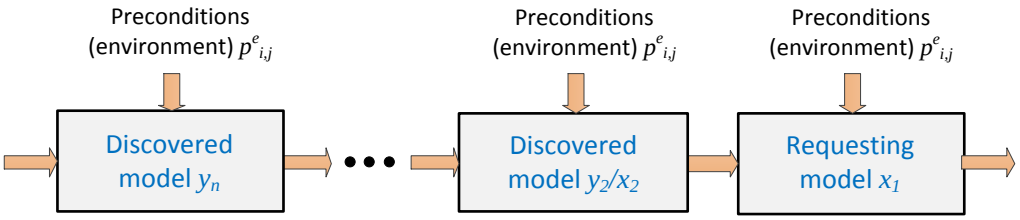



Figure 4 Principle of chained model matching

## 2.3. Definitions and Mathematical Formulations

Let the set $S = \{(x, y)|x \in X, y \in Y\}$=$\{s_i|i = 1, \dots, n_T\}$ be a set of all $n_I$ models and datasets $s_i$ available in the repository, where $x = \{x_1, x_2, \cdots, x_{n_M}\}$ is the set of $n_M$ models, $y = \{y_1, y_2, \cdots, y_{n_D}\}$ is the set of $n_D$ datasets, and hence $n_T = n_M + n_D$. Also, let $P_i^{n_I}$ be a set of $n_I$ properties characterising inputs to the models $s_i$

$$P_i^{n_I} = \{p_{i,j}|i = 1, \cdots, n_T\}_{j=1}^{n_I},\ p_{i,j} \overset{char}{\Longrightarrow} s_i \tag{1}$$

and $P_i^{no}$ be a set of $n_O$ properties characterising outputs of the models and datasets $s_i$

$$P_i^{n_O} = \{p_{i,j} | i = 1, \cdots, n_T\}_{j=1}^{n_O},\ p_{i,j} \overset{char}{\Longrightarrow} s_i \tag{2}$$

both with the subsets of $N_I$ numerical properties $P_i^{N_I}$ for inputs

$$P_i^{N_I} = \{p_{i,j} | p_{i,j} \in \mathbb{R},\ i = 1, \cdots, n_T\}_{j=1}^{N_I} \subseteq P_i^{n_I} \tag{3}$$

and with the subsets of $N_O$ numerical properties $P_i^{N_O}$ for outputs

$$P_i^{N_O} = \{p_{i,j} | p_{i,j} \in \mathbb{R},\ i = 1, \cdots, n_T\}_{j=1}^{N_O} \subseteq P_i^{n_o} \tag{4}$$

To provide more logical arrangements, let $S_i^I$ be an ordered subset of finite number of elements in $S$ as

$$S_i^I = \{s_j\}_{j=0}^{n_C},\ p_j := p_k\ \wedge\ \forall\, j > 0 \tag{5}$$

where $n_C$ is the total number of instances sharing common properties. If $S_i^I$ observes (5) with all instances having intentionally equal[1] properties $p_j := p_k$ and $S_i^I \subseteq S$, then $S_i^I$ is a class of instances $\{s_j\}_{j=1}^{n_C} = s_1, s_2, \cdots, s_{n_C}$ characterised by set of $n_P$ properties $P_i^{n_P} = \{p_{i,j} | i = 1, \cdots, n_T\}_{j=1}^{n_P}$. As all instances of a class $S_i^I$ share common properties, then $P_i^{n_P}$ semantically describes the class $S_i^I$. For $n_C = 0$ in eq. (5), $S_i^I \subseteq S$ is an empty class and still having properties $p_j$. Again, out of all considered $n_T$ properties, the set of $n_I$ properties for inputs is normally different from the set of $n_O$ properties for outputs.

Let $N_i^I$ be a distinct name of the class $S_i^I$, then intension $I_i^I$ of the class $S_i^I$ is defined as 3-tuple (Junli et al. 2006);

$$I_i^I := \langle N_i^I, P_i^{n_P}, S_i^I \rangle \tag{6}$$

Also, let $S_k^I$ be a superset of $S_i^I$ such that

$$S_i^I \subseteq S_k^I \subseteq S,\ \forall\, P_k^{n_P} \subseteq P_i^{n_P}\ \wedge\ i \neq k \tag{7}$$

In ontological sense, the set $S_k^I$ is the superclass of $S_i^I$, if $S_k^I$ observes (7) by following subsumption condition $S_i^I \subseteq S_k^I$ and inheritance condition $P_k^{n_P} \subseteq P_i^{n_P}$.

Let $H_C$ be a superset of $S_i^I$ such that

$$H_C = \bigcup_i S_i^I \tag{8}$$

If $H_C$ observes eq. (8) and, if $S_i^I$ follows subsumption and inheritance conditions given by eq. (7), then $H_C$ could be considered as a graph $H_C = (S_i^I, is - a)$ forming a subsumption hierarchy in ontology sense, called the *subsumption*, were $is - a$ indicates the edge between the nodes of the graph representing classes, and hence representing class-subclass participation. In the subsumption, a superclass contains all the instances of all its subclasses, but it can also have instances on its own. Also, all the properties $P_i^{n_P}$ characterising superclass are inherited by all subclasses.

Two non-empty subclasses $S_i^I$ and $S_j^I$ are disjoint classes if $S_i^I \cap S_j^I = 0, \forall i \neq j$. In practical terms, disjoint classes cannot share instances.

---

[1] Two instances are intentionally equal if they have the same structure of the properties, not necessarily the same property values.

Let $r_{i,j}$ be a relationship between instances other than by class-subclass participation between domain instance $s_{k,i}$ and range instance $s_{k,j}$, then the class relationship $R_i^C$ is a set of bijective relationships between all elements of domain class $S_i^I$ and range class $S_j^I$ defined as

$$R_i^C = \left\{ r_{i,j}\left(S_i^I, S_j^I\right) \middle| \forall \left( \left(S_i^I, S_j^I\right) \in S, i \neq j \right) \right\} \tag{9}$$

Note in eq. (9) that the term $r_{i,j}\left(S_i^I, S_j^I\right)$ refers to a predicate calculus form. The relationships can also be organised in a $n_R$-dimensional subsumption $R^C$ as

$$R^C = \left\{ r_{i.j}\left(S_i^I, S_j^I\right) \middle| \forall i \neq j \right\}_{i,j=1}^{n_R} \tag{10}$$

Although the inclusion mapping $i = j$ in eq. (9) and (10) is generally possible, we exclude such a reflexive relationship for the purpose of simplifying the process without limiting practical aspect of the application in mind. For $r_{i,j}^{-1}$ being inverse instant relationship of $r_{i,j}$, then $R_i^{C^{-1}}$ $\left(= \left\{ r_{i,j}^{-1}\left(S_j^I, S_i^I\right) \middle| \forall \left( \left(S_j^I, S_i^I\right) \in S, i \neq j \right) \right\}\right)$ is the inverse class relationship of $R_i^C$.

Extension of a class $S_i^I$ is defined by the relationship $R_i^C$ which profiles the structural properties of the class by its relations with other classes (Junli et al. 2006).

Let $S_i^D$ be a subset of relationship domain $S_i^I$ and $S_i^R$ be a subset of relationship range $S_j^I$, then the restriction of $S_i^I = \mathrm{dom}\left(R_i^C\right)$ to $S_i^D$ is a partial function $f_D = \mathrm{dom}R_i^C|_{S_i^R}$ providing inclusion map $S \xrightarrow{f_D} S$ as

$$f_D : S_i^I \xrightarrow{f_D} S_i^D \tag{11}$$

and the restriction of $S_j^I = \mathrm{rang}(R_i^C)$ to $S_j^R$ is a partial function $f_R = \mathrm{rang}R_i^C|_{S_i^D}$ providing inclusion map $S \xrightarrow{f_R} S$ as

$$f_R : S_j^I \xrightarrow{f_R} S_j^R \tag{12}$$

In consequence, $f_D$ (and $f_R$) establishes the binary relationship between:

- $S_i^D$ and $S_j^R$ based on universal and existential quantifiers over properties $R_i^C$ of $S_i^I$,
- $S_i^D$ and $n, n \in \mathbb{N}$, based on cardinality quantifiers over properties $P_i^{n_P}$ of $S_i^I$,
- $S_i^D$ and $v,\ v \in s_i \vee N$, based on equality quantifiers over properties $P_i^{n_P}$ of $S_i^I$.

Let $R_i^C\ and\ R_j^C$ be the extensions of classes $S_i^I$ and $S_j^I$ respectively, then $S_i^I$ and $S_j^I$ are equivalent classes, if $R_i^C = R_j^C$ and if $S_i^I \cap S_j^I = S_i^I \cup S_j^I$.

A set of classes $H^I$, subsumption hierarchy $H_C$, set of relationships $R_i^C$, relationship hierarchy $R^C$ and the set of instances $S_i^I$ form an ontology $O$ expressed as 5-tuple

$$O = \langle H^I, H_C, R_i^C, R^C, S_i^I \rangle \tag{13}$$

If the ontology given by eq. (13) is used to provide hierarchically structured set of causes and effects for understanding the (knowledge) domain, which is an effective means to explicitly describe knowledge in knowledge base, then eq. (13) refers to the domain ontology. In practical terms, domain ontology refers to a collection of interlinked concepts, or names $N_i^I$ as suggested by eq. (6), the concept attributes or properties $P_i^{n_C}$ and functions or logical statements $R_i^C$ expressing the constraints

existing in the domain and restricting the interpretation of vocabulary (Qi et al. 2009), all arranged in respective hierarchies $H_C$ and $R^C$ and supplemented by class-attached instances $S_i^I$. The terms `class` and `concept` are then interchangeable.

Let a h-metric $h_i$ be defined over set of properties $P_i^n$ characterising model inputs as well as model and data outputs as

$$h_i: P_i^n \xrightarrow{h} \mathbb{R} \tag{14}$$

then the object $(P_i^n, h_i)$ forms a metric space over $S_m$. By observing numerical properties $P_i^N$ (which includes $N_I$ numerical properties of inputs and $N_O$ of outputs) as $N$-dimensional vector $\mathbf{p}_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,N})$, objects $(P_i^N, \mathbf{p}_i)$ form the vector space[2] $Q^N$ of $n$ vectors.

For metric $h_i$ observing eq. (14) and respective metric and vector spaces, every pair of vectors $(\mathbf{p}_i, \mathbf{p}_j)$ can be mapped as $S_m^2 \to \mathbb{R}$:

$$h: S_m \times S_m \xrightarrow{h} h(p_i, p_j) \equiv h_{i,j} \in \mathbb{R} \tag{15}$$

Let a h-metric $h_k^V$ be defined over the vector space $Q^n$ as mapping from $\mathbb{R}^n \to \mathbb{R}$ so that

$$h_k^V: Q^n \xrightarrow{h} h\left((\mathbf{p}_i, \mathbf{p}_j)\right) \tag{16}$$

then we can define similarity measure of the object $\left(Q^n, (\mathbf{p}_i, \mathbf{p}_j)\right)$ as

$$h_k^V = \left\{\frac{\mathbf{p}_i \cdot \mathbf{p}_j}{\|\mathbf{p}_i\|\|\mathbf{p}_j\|}\right\}_{i,j=1}^{\mathrm{n}}, \; k = 1,2,\cdots,2^{\mathrm{n}} \tag{17}$$

representing the measure known as the vector similarity.

Equivalently, for metric $h_i$ observing eq. (14) and respective metric and vector spaces, every pair of classes $(S_i^I, S_j^I)$ can also be mapped as $S^2 \to \mathbb{R}$:

$$h: S_m \times S_m \xrightarrow{h} h(s_i, s_j) \equiv h_{i,j} \in \mathbb{R} \tag{18}$$

Let a h-metric $h_k^C$ be defined over $H^I$ as mapping from $S^n \to \mathbb{R}$ so that

$$h_k^C: H^I \xrightarrow{h} h\left((S_i^I, S_j^I)\right) \tag{19}$$

then we can define similarity measure of the object $\left(H^I, (S_i^I, S_j^I)\right)$ as

$$h_k^C = \min_{S_C^I \in H^I} \left[\delta(S_i^I, S_C^I) + \delta(S_j^I, S_C^I)\right] \tag{20}$$

where $\delta(S_i^I, S_C^I)$ ($\delta(S_j^I, S_C^I)$) is the distance between classes $S_i^I$ ($S_j^I$) and another class $S_C^I$ measured in number of intermediate edges[3] in graph sense along subsumption $H_C$ and $R_i^C$ relationships.

Let the aggregated similarity measure between two instances in respective classes $S_i^I$ and $S_j^I$ be

---

[2] In linear algebra, a vector space is a set $V$ of vectors together with the operations of addition and scalar multiplication (and also with some natural constraints such as closure, associativity, and so on).

[3] The term edge represents the links or relationships between the two classes.

$$h_k = \frac{\alpha h_k^V + \beta h_k^C}{\alpha + \beta} \quad (21)$$

where $\alpha$ and $\beta$ are weighting factors deepening the semantics of the ontology similarity and their values are dictated by the application.

## 2.4. Implementation of Ontology for Model Integration of Biorefining

Ontologies are used to represent knowledge, as described in Section 2.3, in terms of classes $S_i^I$ with unique names $N_i^I$ employing subsumption hierarchies $H_C$, so called taxonomy, which are merely used as classification schemes. The instances $S_i^I$ are organised by common properties $p_{i,j}$, which characterise classes through the relationships $R_i^C$ to specify how they are related. The ontology in the domain of biorefining reflects the knowledge of a conceptual representation of the models and data representing biorefining processes and its inputs and outputs in order to facilitate i) the consistent and explicit description of models and data through common vocabulary for biorefining domain, ii) registration process by parsing the taxonomy of the ontology and instantiation of model in the web-based repository, iii) input/output matching for automated search of models and data based on the request for input of the model, and iv) integration of such models or data. The top level of the ontology developed to evaluate the proposed concept consists of a concept `Model`.

### 2.4.1. Semantic Description of Models in Biorefining

The `Model` concept provides a common reference of existential process models that represent biorefining technologies. The `Model` is classified using the following five main classifications i) `ModelByFunctionality`, ii) `ModelByBiorefiningPlatform`, iii) `ModelByCharacteristics`, iv) `ModelByInputType`, and v) `ModelByOutputType`. The name of each classification represents the name of respective concepts in ontology and the names are self-explanatory. The `ModelByFunctionality` classification describes the functionality of the process models at four different scales, which include individual operating unit level, functional process unit level, process plant level, and supply chain level. Each model that performs a specific functionality is further specified in the domain of biorefining. For example, the model for reaction that represents the biorefining technology applied to convert biomass feedstock into intermediate/final products is categorised into three subgroups of processes: biochemical-, chemical-, and thermochemical processes. This classification is closely linked with intrinsic properties of feedstock as process has certain feedstock requirements as well as process requirements based on biomass characteristics. The `ModelByBiorefiningPlatform` classification represents the intermediates that link between biomass feedstocks and final products where feedstock is fractionated into a number of intermediates that are further processed into final material and energy products. The main intermediates are known as sugar, oil, lignin, gas, syngas, hydrogen, organic juice, pyrolytic liquid, and electricity and heat. The last two classifications reflect the level of detail considered in a model, which are also known as granularity of the model. The `ModelByCharacteristics` classification characterises process models by key aspects, such as scope, complexity, nature, equation form, scale, and type of model. The `ModelByInputType` and `ModelByOutputType` classifications are the structured knowledge representation of internal connection between models, which is mainly used for calculation of semantic similarity measure by input/output matching. The `ModelByInputType` and `ModelByOutputType` classifications describe different types of flows that were identified by the CAPE-OPEN and follows two different categorisations: i) Material and ii) Energy. The Material category typically represents the physical flow from one process unit to the other through inlets and

outlets, and defines the chemical compositions of biomass feedstock and intermediate/final products. It is the most frequently occurring stream type, yet most complex streams to model. Similarly, the Energy category is used to represent energy flows, such as heat transfer. This classification is developed such that it considers the inheritance and the common features of concepts represented through the structure of ontology to evaluate concept in order to obtain a more accurate similarity. Top three levels of classifications are listed in Table 1 where indentation indicates the respective level in the ontology.

Table 1 Classification of the biorefining related process models

| ModelByFunctionality | ModelByBiorefiningPlatform | ModelByCharacteristics | ModelByInputType | ModelByOutputType |
|---|---|---|---|---|
| FunctionalityForEquipmentLevel | SugarPlatform | ModellingScope | MaterialInput | MaterialOutput |
| Reaction | C5SugarPlatform | ModellingAndSimulation | FeedstockByType | ProductType |
| BiochemicalReaction | C6SugarPlatform | ProcessSynthesisAndDesign | VirginResource | BiochemicalProduct |
| ThermochemicalReaction | Bio-OilPlatform | PlanningAndScheduling | WasteResource | Biofuel |
| ChemicalReaction | BiogasPlatform | ProcessMonitoringAndControl | FeedstockBySource | Biomaterial |
| HeatExchange | SyngasPlatform | IntegratedApproach | EnergyCrop | ProductByIndustrySector |
| Heating | HydrogenIPlatform | ComplexityOfModel | PrimaryResidue | CommunicationSector |
| Cooling | OrganicJuicePlatform | Rigorous | Wastes | EnvironmentSector |
| PressureChanger | PyrolyticLiquidPlatform | Shortcut | ChemicalComponent | HealthAndHygieneSector |
| IncreaseInPressure | LigninPlatform | Conceptual | EnergyInput | HousingSector |
| DecreaseInPressure | ElectricityAndHeatPlatform | NatureOfModel | Steam | IndustrialSector |
| Mixing | | Mechanistic | Heat | RecreationSector |
| Splitting | | Empirical | Electricity | SafeFoodSupplySector |
| Separation | | EquationFormOfModel | | TextileSector |
| HomogeneousSeparation | | Dynamic | | TransportationSector |
| HeterogeneousSeparation | | SteadyState | | ChemicalComponent |
| FunctionalityForProcessLevel | | ScaleOfModel | | EnergyOutput |
| PretreatmentProcess | | IndividualOperatingUnit | | Steam |
| SizeReduction | | FunctionalProcess | | Heat |
| Densification | | ProcessPlant | | Electricity |
| Physico-chemicalProcess | | SupplyChain | | |
| ChemicalProcess | | ModellingType | | |
| BiologicalProcess | | SequentialModularApproach | | |
| Densification | | EquationOrientedApproach | | |
| ConversionProcess | | StatisticalModelling | | |
| BiochemicalConversion | | BlockDiagramOriented (ForControl) | | |
| ThermochemicalConversion | | ComputationalFluidDynamics | | |
| ChemicalConversion | | | | |
| SeparationProcess | | | | |
| EquilibriumSeparation | | | | |
| AffinityBasedSeparation | | | | |
| MembraneBasedSeparation | | | | |
| HybridReaction-Separation | | | | |

### 2.4.2. Relation and Attributes using Properties

In order to support integration of the models in biorefining domain, the properties are used to characterise inputs and outputs of the `Model` concept. Each property represents a connection between models as part of an internal representation as CAPE-OPEN defined streams that connect flowsheet blocks in sequential modular simulation. The properties are developed to follow the process of developing a process flow diagram, which consists of the flowsheet blocks and streams that connect the blocks. The construction of the semantic model representing process system begins with identifying the direction of flow of each stream using the properties *`hasInput`* and *`hasOutput`*, the feed streams are denoted as inputs to the model and the outlet streams are denoted as outputs to the model. The properties are organised in property subsumption $R^C$ (eq.(10)) and hence super-properties *`hasInput`* and *`hasOutput`*, have two sub-properties to further specify number of input and output streams to the model and parameters that are associated with inputs and outputs of the model, as shown in Table 2. The relevant parameters for physical characterisation and chemical composition of materials in process streams, in addition to operating conditions for the model representing a particular biorefining process are classified by *`hasInputParameter`* and *`hasOutputParameter`* property. The set of properties that characterise each input and output stream is denoted as a vector and the values of each properties in numerical format are used to calculate the similarity. This classification is adopted for consistency to enable strong encapsulation of any models representing biorefining processes. Again, indentation shown in Table 2 indicates the property level in the property subsumption, as implemented in the domain ontology.

Table 2 Relationships reflecting process of developing a process flow diagram

| Input | Output | Description |
|---|---|---|
| HasInput | hasOutput | Define direction of flow |
| hasNumberofInputs | hasNumberofOutputs | Define number of ports required for the model by type of inputs and outputs |
| hasNumberofMaterialInputs | hasNumberofMaterialOutputs | |
| hasNumberofEnergyInputs | hasNumberofEnergyOutputs | |
| hasMaterialInputs | hasMaterialOutputs | Define value of material composition for each stream |
| hasMaterialInput1 | hasMaterialOutput1 | |
| hasMaterialInput2 | hasMaterialOutput2 | |
| hasMaterialInput3 | hasMaterialOutput3 | |
| : | : | |
| hasEnergyInputs | hasEnergyOutputs | Define value of energy composition for each stream |
| hasEnergyInput1 | hasEnergyOutput1 | |
| hasEnergyInput2 | hasEnergyOutput2 | |
| hasEnergyInput3 | hasEnergyOutput3 | |
| : | : | |
| hasInputParameters | hasOutputParameters | Define parameters of input/output and set values for each parameter in SI units |
| hasInputFlowrate | hasOutputFlowrate | |
| hasMassFlowrate | hasMassFlowrate | |
| hasMolarFlowrate | hasMolarFlowrate | |
| hasVolumetricFlowrate | hasVolumetricFlowrate | |
| hasPhaseFraction | hasPhaseFraction | |
| hasTemperature | hasTemperature | |
| hasPressure | hasPressure | |
| : | : | |

In addition, properties are further used to describe the attributes characterising sub-concepts of the `Model` concept and to enhance inference. There are two main aspects that support semantic matching for the purpose of model integration based on technical compatibility and functional feasibility. The technical compatibility aspect assesses how well models can work together with given conditions without having to be altered properties, such as maximum capacity, main compositions, key parameters etc. The functional feasibility aspect considers the ability of a process that model present to remain operable and satisfy output specifications through functionality of the model, modelling methods, modelling type etc. All properties mentioned in this paper, in order to characterise the attributes and relationship between concepts, are in format of datatype property for simplification.

### 2.4.3. Property Restrictions

The restrictions $f_D$ and $f_R$ on properties, or axioms, as defined by eq. (11) and (12), are introduced to further enrich the knowledge in the domain of biorefining. Value restriction on properties allows to support ontology reclassification using inference engine. As an example, restriction on property *`hasSugarInput`* and *`hasEthanolOutput`*, which are subproperties of *`hasMaterialInput`* and *`hasMaterialOutput`*, respectively, relates to the concepts representing the model `Fermentation` and its quantity. The `Fermentation` concept is defined using the equivalent class stating necessary and sufficient conditions, as illustrated in Figure 5, which semantically interprets as every model that represents fermentation process has quantity of sugar input and ethanol output that are greater than zero. In this particular example, a datatype property links the `Fermentation` concept to the data literal '0.0', which has a type of an xsd:float in order to collect information.

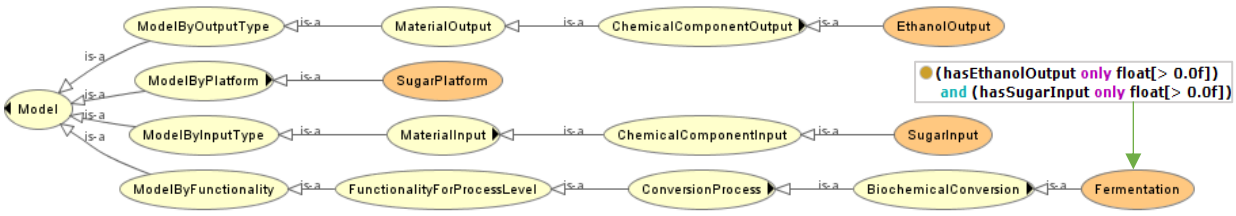



Figure 5 Restriction Example

The composition of inputs and outputs, as well as other characteristics of the process models are defined by restrictions and axioms, which can be used in virtue of input and output validation that enables input/output matching for the model integration process. Along the same line, Figure 6 illustrates an example of reclassification for leveraging the semantic content of ontologies to discover a new form of knowledge.The model `Fermentation` is reclassified as model that has `SugarInputType` and `SugarPlatform`, which is inferred by the restriction "`SugarPlatform` *`hasSugarInput`* allValuesFrom greater than zero." The `SugarPlatform` model is defined as an equivalent class by restriction relating the quantity of sugar through either *`hasSugarInput`* or *`hasSugarOutput`* Properties. As a result, the `Fermentation` model is automatically inferred as a sub-concept of `SugarInputType` and `SugarPlatform`, which is a new form of knowledge generated by inference engine.

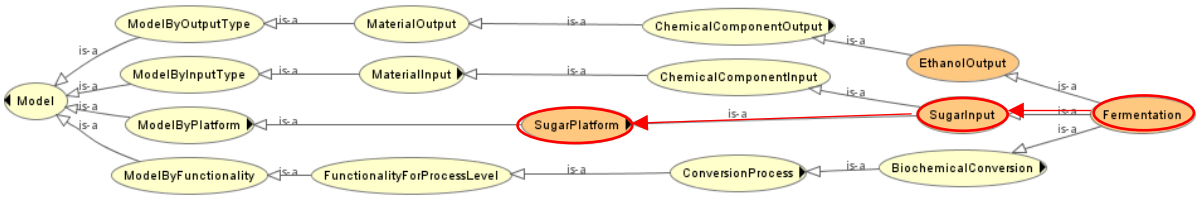



Figure 6 Reclassified Fermentation Concept

# 3. Implementation of Semantic Integration of Process Models in Biorefining

## 3.1. Model Registration

The process of model registration is guided and presented to the user in the form of a questionnaire generated on-the-fly by parsing the domain ontology. The direction of parsing is formulated by previous answers and hence exploits the full potential of the ontology towards providing the best description of the model. The datatype properties associated with most recently parsed concept are then enumerated, the process known as acquisition of explicit knowledge and by which the model becomes an instance of the domain ontology. An example of the registration path of fermentation model is shown in Figure 7. The user initiates the process by selecting the model using one of the classification `ModelByFunctionality`, `ModelByBiorefiningPlatform`, `ModelByCharacteristics`, `ModelByInputType`, and `ModelByOutputType`. The user selects the `ModelByFunctionality` classification and then identifies the scale of the model as a "Process Unit Level" and continues to navigate through the path until the `Fermentation` model is selected. The semantic profile of the model is then created by collecting explicit knowledge of the model and data that are required during the matching process. At the ontology instantiation process as shown in Figure 8, the information contains characteristics of the model as well as its inputs and outputs. The inputs and outputs matching supports the model discovery process based on the semantic relevance between the profile of the models and data, which further facilitates the model integration process.

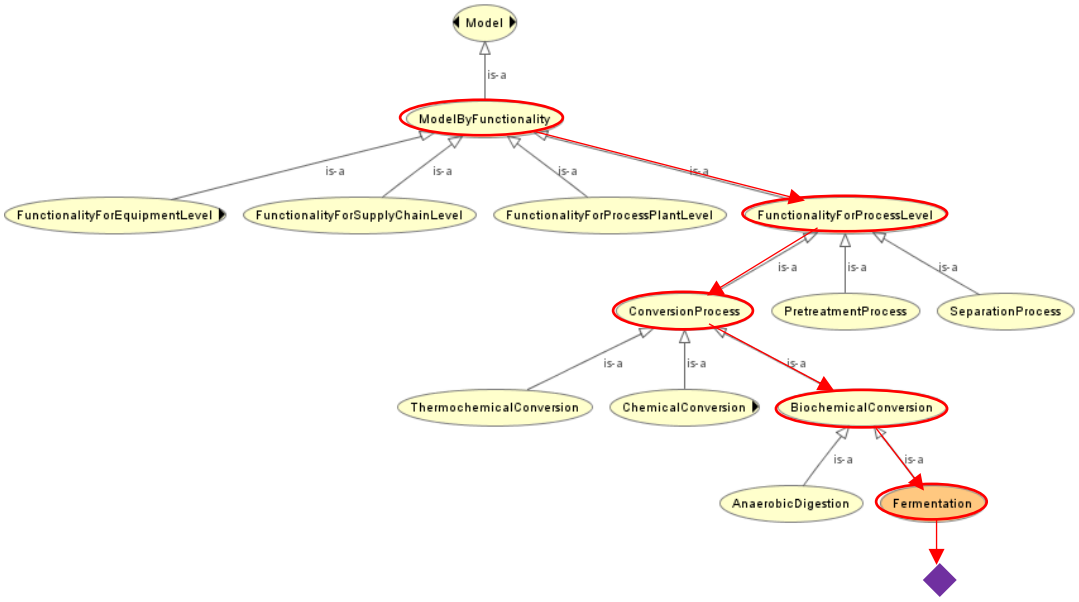



Figure 7 Example of Fermentation Model Registration

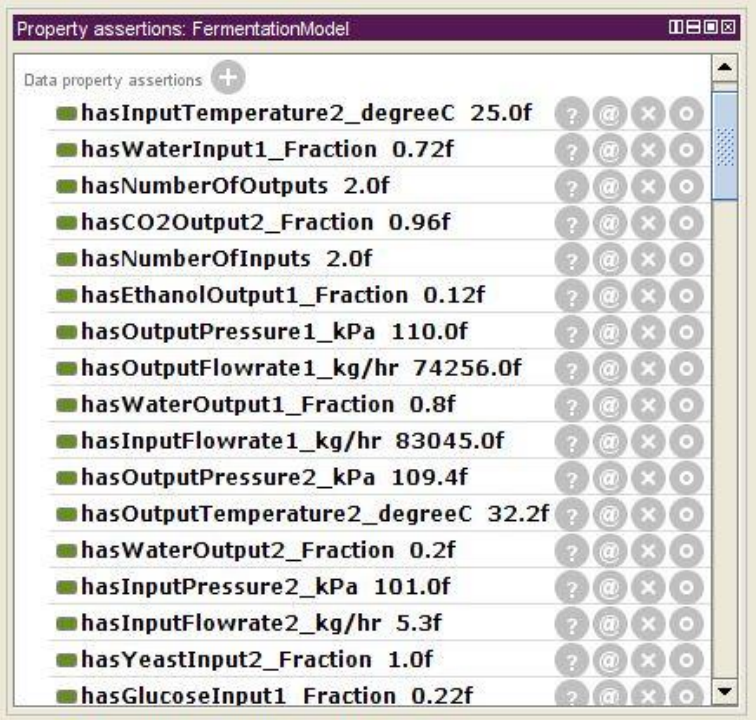



Figure 8 Instantiation of Fermentation Model

## 3.2. Semantic Integration by Input/Output Matching

The formation of semantic integration is performed by the process of matching, which is supported by a domain ontology representing process system models and datasets. The input/output matching facilitates interoperability between models and allows for the automated discovery of candidate models and datasets and hence support model integration (Raafat et al. 2013; Koo et al. 2016). The semantic relevance between the models is measured by the similarity measure $h_k$, as defined by eq. (21) using tacit knowledge of the model (modelling scope, complexity in modelling methods, nature of model, equation form of model, scale of model, modelling type, etc.), as well as explicit knowledge of its inputs and outputs (number of inputs and outputs, type, associated properties, etc.). The tacit knowledge is embedded in the ontology structure which includes subsumption $H_C$, relationships $R_i^C$ and respective relationship subsumptions $R^C$ and restrictions $f_D$ and $f_R$, as defined by eq. (8), (9), (10), (11) and (12), respectively. Similarly, explicit knowledge is quantified by enumerated properties $P_i^{n_I}$, $P_i^{n_O}$ and $P_i^{N_I}$, as defined by eq. (1), (2) and (3), respectively, and formulated in the form of vectors used for input/output matching. The input/output matching is capable to incorporate not only full matching but also considers partial matching to facilitate a wider search capability. Semantic partial matching is considered to suggest alternative options for the model that partially satisfies the matching criteria.

The process of matching undergoes three phases: i) elimination, ii) semantic matching by calculating similarity measures $h_k^V$ and $h_k^C$, and iii) performance ranking. The process of elimination is used to reduce redundant matching without changing the functionality of search and hence to avoid performance deficiency. As at present, the key components required for the input of the requesting model is considered as elimination criterion in the process of elimination. The instances that do not belong to the requested categories are eliminated from matching. The model profiles which are not eliminated from the process of elimination are qualified for the second phase of semantic matching.

Semantic matching is a process of quantifying the semantic relevance between the requesting model and models residing within the repository to determine candidate models and which is based on two methods: i) distance measure (eq. 20) between respective concepts representing tacit knowledge which is measured along the hierarchical relationships and object relationship in the domain ontology; and ii) property similarity (eq. 17) that calculates values of properties that characterise explicit knowledge in the form of vectors and measure by a mean average of cosine and Euclidean similarity (Cecelja et al. 2015).

The distance measure $h_k^C$ is a graph based method for matching, which is a process of calculating similarity between the concepts (Conte et al. 2004) to exploit tacit knowledge embedded in the ontology. Graphs are made of vertices and edges, where the vertices represent the concepts and the edges represent relationships such as subsumption $H_C$ and relationship hierarchy $R^C$. The similarity measure calculates the shortest distance $\delta(S_i^I, S_C^I)$ between two classes where stronger links in ontology graphs are given lower weights. The maximum similarity is given to the class itself, which is defined as an equivalent class with the distance zero. The subsumption relationship, *is-a*, is calculated by counting the number of vertices in a graph model with a weight of 1. The knowledge about the simultaneous processes where two processes occur at the same time in order to increase yield and efficiency was considered on the matching process using object property *hasSimultaneousProcess* and its inverse property *isSimultaneousProcessOf* has the weight of 2. These values are selected to represent experiential side of model integration in practice. The similarity is then normalized by the longest logical path between the vertices in the ontology graph.

In property similarity $h_k^V$, each property that characterises the concept representing explicit knowledge is presented as a vector, which has direction and magnitude. The magnitude of each vector is determined by property value with an assumption that the vectors that are close in space are similar. In the current implementation, four datatype properties are used as criteria of calculating property similarity, which are converted into a four-dimensional vector {Total flowrate, Temperature, Pressure, Fraction of main component}, and which are prepared to expand to more dimensions, as required by practice. Cosine similarity $h_k^{V,C}$ approach calculates the degree of similarity of two vectors expressed as the cosine of the angle between them and Euclidean similarity $h_k^{V,E}$ is considered as it is the most commonly used distance function (Wilson & Martinez 1997). As the Euclidean distance is dealing with parameters of different scales, the normalization, which scales all numeric variables in the range [0,1], is required in order to have the same scale for a fair comparison between two vectors. The shortcoming of cosine similarity in dealing with magnitude of vectors is addressed with the inclusion of Euclidean distance. As a result, the property similarity is a mean average between cosine similarity and normalized Euclidean distance, which is converted into similarity (mentioned hereafter as Euclidean similarity).

To better capture intuition of relevance between the requesting model and existing models in the library, weighting factors $\alpha$ and $\beta$ (eq. (21)) are introduced to allow users to determine the level of interoperability. In the current implementation, the weight of the individual property as well as fuzzy weight, $\alpha$ and $\beta$, for the aggregated similarity are treated as equal, unless user defines otherwise.

## 3.3. Experimental Verification

A real-life biorefining modelling scenario is used to demonstrate the performance of the proposed approach to coordinate model interoperability with regards to technical compatibility and functional feasibility. Here, a reduced number of properties are used to simplify yet purposely illustrate the performance of the designed ontology and matching algorithm, which are the scale, scope, functionality, equation form, modelling type, complexity of models, flowrate, temperature, pressure,

and the fraction of main component. As previously mentioned in Section 2, properties used in characterising inputs and outputs of the model are employed during the matching process. In practice, the number and type of matching criteria are determined based on the input requirements of the requesting model in order for the particular model to run. The matching results are then presented to the user(s) to assist in decision making process hence to fully reflect respective synthesis aspect, which is supported by their expertise in modelling.

The input/output matching as a mean of establishing interoperability between the model(s) and/or dataset(s) is demonstrated by investigating the scenario of discovering models from the repository that potentially satisfies the requirement of the requesting model. MODEL 1 is an Excel-based model registered by the user as a functional process unit representing separation process which purifies bioethanol as a product at 80%. This unit consists of two individual pieces of equipment, which are a flash separator and a distillation column. Water and ethanol are separated from gas and other impurities by flash separator and go through to the distillation column, which further separates water from ethanol. During the process of registration, the user has registered the MODEL 1 as an instance in the repository and identified it as a requesting model, as illustrated in Figure 9. In turn, the requesting model then searches for a potential candidate model(s) that matches according to the requirement in S3. The full set of input requirements of MODEL 1 are given in Table 3, which was provided by the owner of the model during the registration process and, concomitantly, used for matching based on the functionality of the model. As a result of the matching process, the process and simulation model(s) and/or dataset(s) at functional process unit level representing conversion process that produce ethanol as an output are expected to be discovered. The established interoperability, shown in Figure 9, does not aim to create a new pathway, is nevertheless possible to form a biorefining pathway as a result of matching process. Based on the information that user provided, the requesting model, MODEL 1, initiates backward matching process as it becomes the last in the chain. A list of 10 models, residing in the repository, is presented in Table 4 for the demonstration purposes.

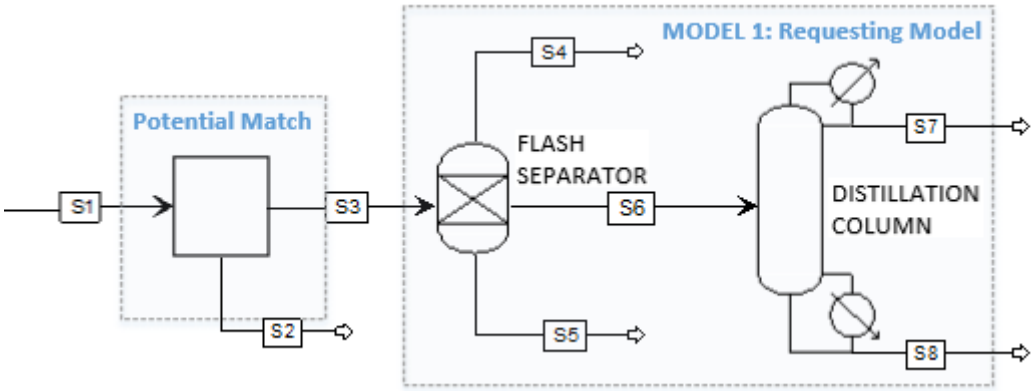



Figure 9 Illustration of Demonstration Scenario

The process of matching undergoes the three matching phases i) elimination, ii) semantic matching by calculating similarity measures $h_k^V$ and $h_k^C$, and iii) performance ranking (Section 3.2). To reduce redundant matching, the key components required for the input of the requesting model is defined as a critical criterion in the process of elimination. As at present, the key component that MODEL 1 requires in input/output matching for the purpose of model integration is identified as ethanol. All the individuals, models and data, registered as an instance in the repository which do not satisfy the

requirements are eliminated during this phase. As a result, MODEL 4 and MODEL 10 (Table 4), which do not have ethanol presence in their output, are eliminated.

In the second phase of matching, quantification of semantic relevance is performed by distance measurement in the ontology, accounting for tacit knowledge. The tacit knowledge about the model, such as semantic descriptions of the models including model functionality, equation form of the model, modelling type, as well as complexity of modelling methods are referred by the classes in the ontology where the instances are attached to. The distance measure $h_k^C$ is used to calculate semantic similarity between the instance of requesting model with other instances of candidate models using graph methods, and which reflect synthesis problem and hence helps the user to make even more informed decision to choose the most appropriate model. To demonstrate the process of matching, Figure 10 illustrates a part of ontology that represents the model by functionality at process level, which is used to calculate semantic relevance between MODEL 1 and MODEL 3. The distance between the two concepts, `Co-Fermentation` and `SSF` (Simultaneous Saccharification Fermentation) are measured along the *is-a* subsumption relationship, as well as an object property *isSimultaneousProcessOf*. As mentioned in Section 3.2, the weights on each property, *is-a* and *isSimultaneousProcessOf*, are 1 and 2 respectively. Therefore, the values of the shortest distance between the concepts `Co-Fermentation` and `SSF` are 3. The similarity is then normalized by the longest logical path that exists between any two concepts in the ontology graph. For the purpose of demonstration, the part of ontology in Figure 10 is the only part that was taken into account in measuring longest path. Note that the concept of MODEL 10 `IndirectGasification` has the maximum distance in terms of number of edges to the concept `Co-Fermentation` and therefore these two concepts are used to determine the longest path in the graph, which is 10 as illustrated in Figure 11. As a result, the similarity for this particular criteria is 0.700. The distance measurement to calculate the semantic relevance based on equation form, modelling type, and complexity of the model is repeated and gives a vector of {0.700, 0.800, 0.800, 0.800}. Finally, similarity measure of MODEL 1 and MODEL 3 for the functional feasibility is calculated to be 0.775, which means the match between requesting model, co-fermentation model, and comparing model, SSF model have the similarity of 77.5%.

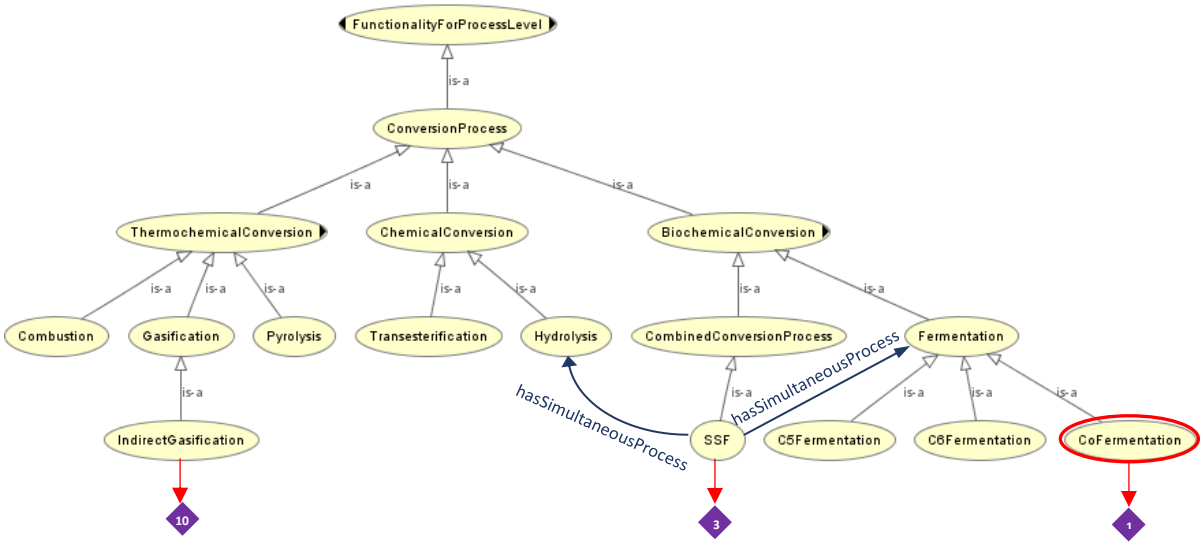



Figure 10 Domain Ontology used for Semantic Matching

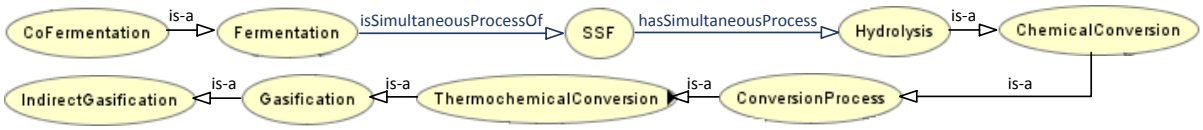



Figure 11 Demonstration of Longest Logical Path

Table 3 Requirement of Requesting Model

| | Distance Matching Requirements | | | | | | | Requirements of Input Parameter* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Model Scale | Model Scope | Model Functionality | Model Functionality for Process | Equation Form | Modelling Type | Complexity | Total Flow (kg/hr) | Temp. (C) | Pressure (kPa) | Ethanol Fraction |
| **MODEL 1** | Process Unit | Modelling & Simulation | Conversion Process | Co Fermentation | Dynamic | Equation Oriented | Detailed | 50,000 | 20-35 | 100-200 | 0.075 |

* Input parameters to the model

Table 4 List of Model Profile in Repository

| | Elimination Criteria | Distance Matching Requirements | | | | Requirements of Input Parameter* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Criteria | Ethanol | Model Functionality for Process | Equation Form | Modelling Type | Complexity | Total Flow (kg/hr) | Temp. (C) | Pressure (kPa) | Ethanol Fraction | Software | Reference |
| **MODEL 2** | Yes | C6 Fermentation | Dynamic | Equation Oriented | Detailed | 117,233 | 34 | 101 | 0.116 | gProms | (Siougkrou et al. 2016) |
| **MODEL 3** | Yes | SSF* | Steady State | Sequential Modular | Shortcut | 449,353 | 40 | 91 | 0.055 | AspenPlus | (Humbird et al. 2011) |
| **MODEL 4** | No | Transesterification | Steady State | Sequential Modular | Detailed | 1,004 | 60 | 395 | 0 | AspenPlus | (Zhang et al. 2003) |
| **MODEL 5** | Yes | Gasification | Steady State | Equation Oriented | Conceptual | 3,967 | 700-1000 | n/a | 0.066 | Data | (Wei et al. 2009) |
| **MODEL 6** | Yes | C6 Fermentation | Steady State | Sequential Modular | Detailed | 74,256 | 32 | 111 | 0.121 | AspenPlus | (AspenPlus 2007) |
| **MODEL 7** | Yes | Gasification | Steady State | Equation Oriented | Conceptual | 1,653 | 200-350 | 6000-7000 | 0.114 | Data | (Wei et al. 2009) |
| **MODEL 8** | Yes | SSF* | Steady State | Equation Oriented | Conceptual | 10,722 | 30 | 101 | 0.016 | Data | (Wei et al. 2009) |
| **MODEL 9** | Yes | C6 Fermentation | Steady State | Sequential Modular | Conceptual | 47,191 | 25 | 101 | 0.075 | AspenPlus | (Siougkrou et al. 2016) |
| **MODEL 10** | No | Indirect Gasification | Steady State | Sequential Modular | Detailed | 6,507 | 870 | 158 | 0 | AspenPlus | (Spath et al. 2005) |

* Simultaneous Saccharification and Fermentation

Four criteria are considered to calculate input/output matching using property similarities and are based on the set of physical properties characterising the inputs of the model they required for the property matching. Table 3 identifies the requirements of input parameter of the requesting model, including total flowrate, temperature, pressure, and fraction of input components. The properties are represented in the form of 4-dimentional vector, $P_i^4 = (total\ flowrate, temperature, pressure, fraction\ of\ input\ component)$. Total flowrate is used as a measure of capacity in the system that requesting model represents, temperature and pressure are operating condition that provides upper and lower limit to set boundaries of the operating condition based on the type of biological strain employed in modelling, and the required input component of the requesting model is ethanol, hence, fraction of the main component is incorporated to ensure the presence of this component. The value of total flow of the requesting model is 50,000 kg/hr, the ranges of temperature and pressure are represented to reflect optimal operating condition for employing Saccharomyces cerevisiae (Galanakis et al. 2012; Deesuth et al. 2016) for the fermentation process, which are 20-35 degree C and 100-200 kPa respectively, and the fraction of ethanol component that is processed by the requesting model is 0.075. To accommodate the range in values of input parameters (temperature and pressure) that the requesting model provided, the closest values of the parameter of MODEL 1 to MODEL 3 are selected. The values of these properties are then converted into a vector and subsequently compared using cosine and Euclidean similarity. MODEL 1 and MODEL 3 are presented in the form of vectors $\mathbf{p}_1 = (50000, 35, 100, 0.075)$ and $\mathbf{p}_3 = (449353, 40, 91, 0.055)$. Cosine similarity $h_k^{V,C}$ and Euclidean similarity $h_k^{V,E}$ are 1.000 and 0.000, respectively, which then combined together as a semantic similarity $h_k^C$ using the mean average gives a result of 0.637. In the case of missing values of temperature and pressure, the default values are atmospheric temperature, 25 degrees C, and atmospheric pressure 100kPa. The final summary of the matches with other models are shown in Table 5.

Table 5 Similarity Results

| | Semantic Similarity $h_k^C$ | Cosine Similarity $h_k^{V,C}$ | Euclidean Similarity $h_k^{V,E}$ | Property Similarity $h_k^V$ | Aggregated Similarity $h_k$ |
|---|---|---|---|---|---|
| **MODEL 2** | 0.917 | 1.000 | 0.832 | 0.916 | 0.916 |
| **MODEL 3** | 0.775 | 1.000 | 0.000 | 0.500 | 0.637 |
| **MODEL 5** | 0.775 | 0.978 | 0.885 | 0.931 | 0.853 |
| **MODEL 6** | 0.850 | 1.000 | 0.939 | 0.970 | 0.910 |
| **MODEL 7** | 0.775 | 0.250 | 0.878 | 0.564 | 0.670 |
| **MODEL 8** | 0.825 | 1.000 | 0.902 | 0.951 | 0.888 |
| **MODEL 9** | 0.800 | 1.000 | 0.993 | 0.996 | 0.898 |

Based on the results the suggested models to establish interoperability with the requesting models have been ranked in Table 6. Following models, MODEL 2, MODEL 6, MODEL 8 and MODEL 9, are suggested to the user as potential candidates, where MODEL 2 being most suitable model. In addition, the MODEL 3, MODEL 5, and MODEL 7 will be flagged to inform the user with their similarity measure that the operating conditions of these models did not meet the requirement range of input parameters and model characteristics that requesting model initially provided. In this stage, the system allows user intervention for the user to make informed decision in choosing a model for user's particular needs.

Table 6 Suggested Models

| | Semantic Similarity $h_k^C$ | Cosine Similarity $h_k^{V,C}$ | Euclidean Similarity $h_k^{V,E}$ | Property Similarity $h_k^V$ | Aggregated Similarity $h_k$ |
|---|---|---|---|---|---|
| MODEL 2 | 0.917 | 1.000 | 0.832 | 0.916 | 0.916 |
| MODEL 6 | 0.850 | 1.000 | 0.939 | 0.970 | 0.910 |
| MODEL 9 | 0.800 | 1.000 | 0.993 | 0.996 | 0.898 |
| MODEL 8 | 0.825 | 1.000 | 0.902 | 0.951 | 0.888 |

In the case of selected model requiring further information for it to run, role of the model becomes a requesting model and the process of matching is then repeated. As previously mentioned, there is a potential to form biorefining pathways as a result of chain matching process, however, it is not a goal of the proposed approach.

## 4. Conclusion And Future Work

The concept of using ontology in model and data integration was introduced to improve upon previous research with particular focus on flexibility (partial matching) and reusability (reuse of existing models and data). The semantic algorithm for establishing interoperability between the models and data is presented to reflect the knowledge based on technical compatibility and functional feasibility. The domain ontology with a particular view to coordinate model integration embeds both tacit and explicit knowledge in the domain of biorefining modelling. Process models and data are semantically annotated in terms of input(s), output(s), precondition(s), the software environment in which they operate, as well as the functionality they perform. It demonstrates the process of registration and instantiation of the model to form model profile, which further supports the input/output matching process. Semantic relevance is measured in terms of semantic similarity by employing a graph matching method and vector similarity; in addition, the semantic partial matching is performed to facilitate the flexibility of model integration. To this end, the suitability of using ontology for model and data integration in process modelling in the domain of biorefining has been successfully verified. Following upon our current implementation, another extension of the framework that we wish to explore is to generalise the concept for all processes in the domain of chemical process systems engineering. Therefore, the results of this paper can be considered as a fundamental step towards the challenging task of defining and implementing extended framework.

## Acknowledgements

The authors wish to acknowledge the financial support by the Marie Curie Initial Training Networks Program, under the RENESENG project (FP7-607415). We would also like to thank School of Chemical Engineering at National Technical University of Athens for their contribution of process models.

## 5. References

AspenPlus, 2007. Aspen Plus Bioethanol from Corn Model.

Belaud, J.P. & Pons, M., 2002. Open software architecture for process simulation: The current status of CAPE-OPEN standard. *Computer Aided Chemical Engineering*, 10(C), pp.847–852.

Bogusch, R. et al., 2000. *CAPE Path Recommandations*,

Braunschweig, B. et al., 2004. CAPE web services: The COGents way. *Computer Aided Chemical Engineering*, 18(C), pp.1021–1026.

Braunschweig, B.L. et al., 2000. Process modeling : The promise of open software architectures. *Chemical engineering progress*, 96(9), pp.65–76.

CAPE-OPEN Project team, 2000. *Conceptual Desing Document (CDD2) for CAPE-OPEN Project*,

Cecelja, F. et al., 2015. Semantic algorithm for Industrial Symbiosis network synthesis. *Computers & Chemical Engineering*, Online.

Conte, D. et al., 2004. Thirty Years Of Graph Matching In Pattern Recognition. *Ijprai*, 18(3), pp.265–298.

Deesuth, O., Laopaiboon, P. & Laopaiboon, L., 2016. High ethanol production under optimal aeration conditions and yeast composition in a very high gravity fermentation from sweet sorghum juice by Saccharomyces cerevisiae. *Industrial Crops and Products*, 92, pp.263–270.

Galanakis, C.M. et al., 2012. Effect of pressure and temperature on alcoholic fermentation by Saccharomyces cerevisiae immobilized on γ-alumina pellets. *Bioresource Technology*, 114, pp.492–498.

Hangos, K.M. & Cameron, I.T., 2001. Process Modelling and Model Analysis. *Process Systems Engineering*, 4, pp.19–40.

Humbird, D. et al., 2011. Process Design and Economics for Biochemical Conversion of Lignocellulosic Biomass to Ethanol. *Renewable Energy*, 303(May), p.147. Available at: http://www.nrel.gov/biomass/pdfs/47764.pdf.

Koo, L. et al., 2016. *A Holistic Approach to Model Discovery Using A Domain Ontology*, Elsevier Masson SAS. Available at: http://dx.doi.org/10.1016/B978-0-444-63428-3.50127-2.

Koo, L. & Cecelja, F., 2015. Model Integration Using Ontology Input-Output Matching. *Computer Aided Chemical Engineering*, 37, pp.2567–2572.

Magioglou, V. et al., 2015. Model-based Analysis of Waste Management Systems through a Natural Language Approach. In *Computer Aided Chemical Engineering*. pp. 977–982.

Morales-Rodríguez, R. et al., 2008. Use of CAPE-OPEN standards in the interoperability between modelling tools (MoT) and process simulators (Simulis® Thermodynamics and ProSimPlus). *Chemical Engineering Research and Design*, 86(7), pp.823–833.

Pons, M., 2010. How to make use of CAPE-OPEN? In *2010 AIChE Annual Meeting, 10AIChE*.

Raafat, T. et al., 2013. An ontological approach towards enabling processing technologies participation in industrial symbiosis. *Computers & Chemical Engineering*, 59, pp.33–46.

Raafat, T. et al., 2012. Semantic Support for Industrial Symbiosis Process. *Computer Aided Chemical Engineering*, 30, pp.452–456.

Siougkrou, E. & Kokossis, A., 2016. Development of semantically-enabled community hubs in biorefineries and biorenewables. In *Computer Aided Chemical Engineering*. pp. 2013–2018.

Siougkrou, E., Mountraki, A. & Kokossis, A., 2016. *Model Integration and Interoperability*, National Technical University of Athens: Workshop Presentation.

Spath, P. et al., 2005. *Biomass to Hydrogen Production Detailed Design and Economics Utilizing the*

*Battele Columbus Laboratory Indirectly-Heated Gasifier.*, Available at: http://www.nrel.gov/docs/fy05osti/37408.pdf.

Trokanas, N., Bussemaker, M., et al., 2015. BiOnto: An Ontology for Biomass and Biorefining Technologies. In *Computer Aided Chemical Engineering*. pp. 959–964.

Trokanas, N. et al., 2012. Semantic formalism for waste and processing technology classifications using ontology models. *Computer Aided Chemical Engineering*, 30, pp.167–171.

Trokanas, N., Cecelja, F. & Raafat, T., 2015. Semantic approach for pre-assessment of environmental indicators in Industrial Symbiosis. *Journal of Cleaner Production*, 96, pp.349–361.

Trokanas, N., Cecelja, F. & Raafat, T., 2014. Semantic input/output matching for waste processing in industrial symbiosis. *Computers & Chemical Engineering*, 66, pp.259–268.

van Baten, J. & Pons, M., 2014. CAPE-OPEN: Interoperability in Industrial Flowsheet Simulation Software. *Chemie Ingenieur Technik*, 86(7), pp.1052–1064. Available at: http://doi.wiley.com/10.1002/cite.201400009.

Wei, L. et al., 2009. Process engineering evaluation of ethanol production from wood through bioprocessing and chemical catalysis. *Biomass and Bioenergy*, 33(2), pp.255–266.

Wilson, D.R. & Martinez, T.R., 1997. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6, pp.1–34.

Yang, A. et al., 2008. A multi-agent system to facilitate component-based process modeling and design. *Computers and Chemical Engineering*, 32(10), pp.2290–2305.

Yang, A. & Marquardt, W., 2004. An Ontology-based Approach to Conceptual Process Modelling. *Computer Aided Chemical Engineering*, 18, pp.1159–1164.

Zhang, Y. et al., 2003. Biodiesel production from waste cooking oil: 1. Process design and technological assessment. *Bioresource Technology*, 89(1), pp.1–16.