

УДК 33

DOI: 10.34670/AR.2022.77.66.016

Экономические аспекты модели симулятора технологического процесса первичной переработки какао-бобов

Камдум Рафаэль

Магистр,

Московский государственный университет пищевых производств,
125080, Российская Федерация, Москва, Волоколамское шоссе, 11;
e-mail: kamdoumraphael5@gmail.com

Максимов Алексей Сергеевич

Профессор,

Кандидат технических наук

кафедра информатики и вычислительной техники пищевых производств,
Московский государственный университет пищевых производств,
125080, Российская Федерация, Москва, Волоколамское шоссе, 11;
e-mail: maksimov@mgupp.ru

Аннотация

В статье поднимается вопрос эксплуатации семантического поля как инструмента бизнес-моделирования. Подробно описаны концепция и принцип действия данного нового подхода, который называется онтологией. Онтологии представляют собой модели в семантической форме, для их реализации разработано несколько программ. Среди этих программ авторы выбрали платформу Protégé из-за ее многочисленных преимуществ. После ее относительно простой установки в рабочей среде ее эксплуатация позволила создать онтологию. Это было сделано путем создания сначала метаданных онтологии, затем класса, подкласса и непересекающихся классов, затем отношений, подотношений и обратных отношений, после этого осуществлялись установление характеристик и определение областей и объема онтологии, наконец, был получен результат онтологии и последовало обсуждение.

Для цитирования в научных исследованиях

Камдум Р., Максимов А.С. Экономические аспекты модели симулятора технологического процесса первичной переработки какао-бобов // Экономика: вчера, сегодня, завтра. 2022. Том 12. № 7А. С. 145-152. DOI: 10.34670/AR.2022.77.66.016

Ключевые слова

Онтология, модель, индустрия семантического моделирования, переработка какао-бобов.

Введение

Во все более конкурентной рыночной среде производственные компании постоянно ищут пути и средства, чтобы иметь продукцию, которая всегда актуальна по количеству и качеству. Настоящее исследование тем более важно, что компании находятся в контексте постоянно меняющихся потребностей клиентов. Таким компаниям, особенно производителям пищевых продуктов, необходимо постоянно активировать новые инновационные механизмы, чтобы адаптироваться к динамике рынка, а также быстро и эффективно реагировать на потребности клиентов.

Для этого необходимо срочно внедрить правильную политику планирования производства. На этом этапе может быть эффективным создание цифровых моделей с использованием информационных структур, таких как онтологии.

Таким образом, наши эксперименты являются частью контекста содействия поиску решения проблемы моделирования в отраслях по переработке какао. Чтобы лучше решить эту проблему, мы рассмотрим оборудование и методы, которые будут задействованы для реализации нашей онтологической модели, а затем представим результаты и краткое обсуждение.

Основное содержание

Для построения онтологии могут использоваться две большие группы инструментов: средства построения онтологий, зависящие от формализма представления (Ontolingua, OntoSaurus, Интернет, OilEd); средства построения онтологий, независимые от формализма представления (защищенные, ODE и WebOde, OntoEdit).

Исходя из нашего опыта, мы решили использовать инструмент для построения онтологии, не зависящий от формализма представления: программу платформы Protégé (редактор онтологий), очень удобную для пользователя структуру базы знаний на основе Java. Ее преимущество в том, что она находится в свободном доступе. Ранее мы видели, что онтология представляет собой структурированный набор понятий в определенной области знаний. Итак, Protégé предлагает три метода разработки онтологии: редактор Protégé-OWL для построения онтологий для семантической сети (Protégé 3.4.2 и Защищенный 4.0.2); редактор Protégé-Frames для создания и заполнения баз знаний на основе фреймов (Protégé 3.4.2); редактор Protégé также позволяет редактировать онтологии через веб-интерфейс (WebProtégé 0.5 alpha).

Protégé позволяет использовать язык OWL – компьютерный язык, используемый для моделирования онтологий. OWL – это диалект XML, расширяющий RDFS. Protege позволяет работать в режиме клиент/сервер с регулируемым управлением правами в зависимости от желаемых профилей и объединяет инструменты для совместной работы.

Построение конструкции онтологии требует соблюдения определенного ряда критериев: полнота; ясность и объективность; использование экземпляров для представления элементов.

Процедура создания онтологии становится относительно простой, когда у имеется лучшее знание работы инструмента, используемого для моделирования. В платформе Protégé создание онтологии всегда начинается со скачивания и установки программы, совместимой со средой имеющейся операционной системы.

Несколько шагов составляют процесс построения нашей онтологии:

1. Настройка рабочей среды.

Будучи бесплатным редактором онтологий с открытым исходным кодом, Protégé 5 можно

загрузить по следующей ссылке: <http://protege.stanford.edu/>, затем «Загрузить сейчас» позволяет загрузить программное обеспечение. Следует обратить внимание, что «Загрузить для...» автоматически предлагает версию для подключенного компьютера. После установки программы нужно перейти на вкладку меню «Файл», выбрать «Создать» для создания онтологии. Также важно не забыть сохранить проект. В разделе «Сохранить как...» необходимо назвать файл «Simulation.owl», формат сохранения должен быть в RDF/XML.

2. Создание метаданных онтологии.

Здесь речь идет о добавлении комментария к онтологии, для этого необходимо убедиться, что вкладка «Active ontology» активна. Затем в заголовке онтологии «Ontology header» справа от «Annotations» есть знак «+», следует щелкнуть по нему. Появится редактируемое диалоговое окно. После нужно ввести комментарий «ТП от 07.06.2022» в поле «owl:versioninfo» и нажать «ОК».

Тот же процесс применяется для создания комментария с той лишь разницей, что нужно выбрать средний значок, чтобы добавить аннотацию. Назвать его следует «Описание», далее нажать «ОК». После этого отредактировать свое описание следующим описанием «Моя онтология моделирования», проверив, какой язык редактируется в нижней части окна.

3. Создание классов.

Для создания каждого нового класса необходимо возвращаться к URI онтологии. Это можно сделать двумя способами: введя пространство имен непосредственно в поле создания классов (вкладка «Classes») или посредством меню «Edit», затем «Create new». При этом необходимо создать иерархию классов, состоящую из «Сценария», «Учебного материала», «Контента» и «Движка». Для этого надо выделить курсором узел «Owl Thing» и нажать на кнопку «Add subclass», чтобы добавить класс. Новый класс появляется как подкласс «Owl Thing». Откроется диалоговое окно, в котором можно ввести имя вашего нового класса. Далее нужно ввести «Сценарий», повторить операцию для классов «Учебный материал», «Контент», «Движок». Важно всегда следить за тем, чтобы они отображались как подклассы «Owl Thing».

Следует также отметить, что другой метод заключался в использовании кнопки «Создать родственный класс» («Add sibling class») для создания «Учебных материалов», «Контента» и «Содержания». В этом случае необходимо предварительно выбрать класс «Сценарий».

Непересекающийся класс

Теперь речь идет о том, чтобы сделать разные классы непересекающимися. Для этого в разделе «Description» класса «Сценарий» следует нажать кнопку + справа от «Disjoint With». Откроется окно, в котором можно выбрать классы, которые необходимо разделить. Сделав это для одного из классов, например «Контента», действие будет автоматически распространено на другие выбранные классы.

Создание подклассов

Как мы видели ранее, каждый раз, когда создается новый класс, необходимо всегда возвращаться к URI онтологии. Следует убедиться, что выбрана вкладка «Classes». Далее нужно выбрать концепцию «Сценарий». Затем посредством команд меню «Tools» > «Create class hierarchy...» открыть окно «Enter hierarchy». В открывшемся окне построчно ввести перечень подклассов и нажать кнопку «Continue». Инструмент проверяет совместимость введенных имен с предустановленным стилем (без пробелов, без диакритических знаков). Затем устанавливается переключатель «Сделать одноуровневые классы непересекающимися» (Make sibling classes disjoint) и нажимается кнопка «Finish».

Процесс создания подмножеств других классов будет таким же. В итоге получаем дерево

классов онтологии симулятора технологического процесса первичной переработки какао-бобов.

4. Создание отношений.

Отношения между объектами онтологии выражают то, как объекты связаны друг с другом. Как правило, отношения и подотношения будут очень важны в нашей модели, для их создания следует перейти на вкладку «Свойства объекта» (Object properties). Далее нужно выделить топовый класс «owl:topObjectProperty» и воспользоваться возможностями главного меню «Tools» > «Create object property hierarchy...». В открывшемся диалоговом окне необходимо ввести список следующих свойств: «создает», «является продуктом», «предшествует» и «выполняется после» и завершить операцию.

Создание подчиненных отношений

Чтобы создать подчиненные отношения, нужно перейти на вкладку «Свойства объекта» (Object properties), выделить свойство «предшествует» и нажать кнопку «Добавить подсвойство (Add sub property)», чтобы создать новую связь. Назвать новое свойство следует как «Выполняется перед», заполнив появившееся диалоговое окно и завершив операцию.

Созданные свойства будут учитывать отношения между экземплярами классов.

Создание экземпляров классов

На этом этапе требуется создать экземпляры классов «Технологические операции» и «Продукты». Для этого в окне «Classes» нужно выделить класс «Технологические операции» и в разделе «Instances» панели «Description» создать экземпляры класса в соответствии с названиями отдельных операций процесса производства какао продуктов.

Аналогично создаются экземпляры класса «Продукты».

5. Настройка свойств.

После создания свойств и экземпляров классов можно приступить к настройке свойств, которая заключается в определении характеристики свойства и заполнении обязательных разделов панели описания свойства.

Определение предметной области и области действия онтологии

Здесь речь идет об определении источника (Domains) и области действия (Ranges) каждого отношения. Для этого в иерархии следует выбрать отношение «Создает».

Необходимо установить источник, нажав значок «+» справа от (Domains), и выбрать класс «Технологические операции». Затем указать область действия, нажав на значок «+» справа от «Ranges», открыть иерархию классов и выбрать класс «Продукт».

Создание обратных отношений

Полная онтология должна требовать создания прямых и обратных отношений. Пусть отношение «Является продуктом» обратно отношению «Создает». Чтобы создать эту инверсию, нужно выделить свойство «Создает» и в панели «Description» нажать значок «+» справа от «Inverse Of». Это действие открывает диалоговое окно, в котором следует выбрать обратное отношение «Является продуктом». В этой же панели для свойства «Создает» надо указать характеристику «Inverse functional». Это значит, что каждая операция может создать несколько продуктов.

Теперь необходимо указать, что конкретно создает каждая технологическая операция. Для этого надо перейти на вкладку «Individuals», для каждой операции в разделе «Object property assertions» окна «Property assertions» нажать кнопку «+» и в открывшемся окне вручную ввести наименование свойства («создает») и наименование соответствующего экземпляра класса «Продукт».

Выполненная настройка свойств объектов онтологии позволяет интеллектуальной части

редактора Protege самостоятельно сгенерировать действие обратного свойства «Является продуктом». Для этого достаточно включить плагин Reasoner.

Аналогичным образом настраиваем свойства «выполняется после», «выполняется перед» и «предшествует».

Для всех перечисленных свойств в разделах Domain и Ranges указываем класс «Технологические операции». Отмечаем, что свойство «выполняется после» является инверсным по отношению к свойству «выполняется перед». В панели «Characteristics» отмечаем, что свойство «выполняется после» является функциональным (Functional), свойство «выполняется перед» является обратным функциональным (Inverse functional) и свойство «предшествует» является транзитивным (Transitive).

После этого возвращаемся на вкладку Individuals и с помощью свойства «выполняется после» указываем последовательность технологических операций. Остальную работу выполняет плагин Reasoner.

Результаты и обсуждение

Чтобы иметь законченную и хорошо представленную онтологическую модель, важно сначала сформировать древовидную структуру предметной области симулятора (рис. 7), а затем с помощью набора инструментов, содержащихся в программе Protégé, представить структуру предметной области в виде ориентированного графа. Этот инструмент называется OntoGraf.

Наша экспериментальная онтология содержит около 40 понятий моделирования переработки какао бобов.

Древовидная структура показывает, что она содержит:

Классы предметной области симулятора (движок, контент, tutoriales и сценарий), не пересекающиеся друг с другом.

Подклассы, такие как «Режим работы тренажера», «Проверка знаний» и «Интерфейс пользователя», которые являются подклассами экземпляра «Сценарий». Следует обратить внимание, что подклассы онтологии имеют три степени вложенности, например, подклассы «Экзамен» и «Тест» являются подклассами класса «Контроль знаний», а «Контроль знаний», в свою очередь, является подклассом класса «Сценарий». Таким образом, все подклассы, являющиеся членами класса «Контроль знаний», также являются членами класса «Сценарий».

Свойства объектов (отношения). Несколько типов отношений составляют нашу онтологическую архитектуру: функциональные, обратные функциональные и транзитивные. В частности, транзитивным является свойство «предшествует», которое относится к классу технологических операций. Оно имеет подчиненное свойство «выполняется перед». В связи с этим любая технологическая операция, кроме последней, предшествует по крайней мере одной из них. А операция, выполняемая первой, предшествует всем остальным.

Мы также отмечаем, что некоторые классы и подклассы существуют отдельно друг от друга. Теперь мы поймем важность непересекающихся классов в онтологии. На самом деле в языке OWL классы по умолчанию пересекаются. Таким образом, было бы трудно утверждать, что экземпляр не является членом определенного класса только потому, что он не был определен как член этого класса, поэтому для разделения группы классов они должны быть объявлены как непересекающиеся (Disjoint With). Это предотвращает принадлежность объекта, созданного как член класса в группе, к другому классу в той же группе. Пример в нашей модели выше: «Сценарий», «Учебный материал» и «Контент» не пересекаются друг с другом. Это означает,

что ни один подкласс не может находиться в этих трех классах одновременно.

Reasoner (встроенный плагин Protege) подтверждает соответствие созданной онтологии базе знаний. На основе заложенных свойств объектов онтологии интеллектуальная часть редактора Protégé позволяет получить новые знания. На рисунке 8 показан результат работы плагина Reasoner для технологической операции «Дробление какао-бобов». В панели «Property assertions» на желтом фоне показана полученная резонером информация. На основании информации о том, что данная операция выполняется после обжарки какао-бобов, и на основании сформулированных выше свойств и их характеристик получены новые знания о том, что операция выполняется перед измельчением какао-крупки и предшествует прессованию какао тертого и размолу и просеиванию жмыха какао

Заключение

Таким образом, нами представлено построение онтологической модели тренажера-симулятора технологического процесса первичной переработки плодов какао в рамках поиска оптимальных решений для потребностей компаний с точки зрения производительности и конкурентоспособности. На экспериментальном примере показано, что семантическое поле на онтологическом языке представляет собой ценность для создания симуляторов технологических процессов, подобных процессу первичной переработки какао-бобов, на этапе их моделирования.

Очевидно, что результаты обнадеживают. Однако было бы более оптимально использовать онтологии в функциональных анализах, таких как методы анализа SADT этого процесса обработка какао.

Библиография

1. Абади А., Секкат С., Земмури Э.М., Бен-Азза Х. Использование онтологий для поддержки одновременного проектирования продукта и цепочки его поставок. URL: <http://ceurws.org/Vol-600/paper3.pdf>.
2. Бертран Ф. и др. Онтологии для моделирования процессов ухода в медицинских учреждениях. URL: <http://ceurws.org/Vol-600/paper3.pdf>.
3. Гандон Ф. Компьютерные онтологии. URL: <https://interstices.info/ontologies-informatiques>.
4. Джон Шарлет, Одри Бэникс, Ксавье Эме, Домбр Ф. Создание онтология в OWL с Protégé 4/5. URL: https://www.researchgate.net/publication/318208041_Creation_d'une_ontologie_en_OWL_avec_Protege_45
https://www.researchgate.net/publication/318208041_Creation_d'une_ontologie_en_OWL_avec_Protege_45.
5. Фортино В. Вклад в онтологическое моделирование информации на протяжении всего жизненного цикла товара. URL: https://www.researchgate.net/publication/261026131_Contribution_a_une_modelisation_ontologique_des_information_tout_au_long_du_cycle_de_vie_du_product.
6. Шарлет Д., Стейхен О., Бэникс О., Алеку Ю. Используйте и стройте онтологии в медицине. Примат терминологии. URL: researchgate.net.
7. Xiaojun Y.E. Моделирование и симуляция производственных систем: объектно-ориентированный подход. URL: <https://tel.archives-ouvertes.fr/tel-00821121>.
8. Harrison W. S., Tilbury D. M., Yuan C. From hardware-in-the-loop to hybrid process simulation: An ontology for the implementation phase of a manufacturing system //IEEE Transactions on Automation Science and Engineering. – 2011. – Т. 9. – №. 1. – С. 96-109.
9. Luukkainen M., Karhela T. Ontology approach for co-use of 3D plant modelling and large scale process simulation //The 48th Scandinavian Conference on Simulation and Modeling (SIMS 2007); 30-31 October; 2007; Göteborg (Särö). – Linköping University Electronic Press, 2007. – №. 027. – С. 166-172.
10. Miller J. A. et al. Investigating ontologies for simulation modeling //37th Annual Simulation Symposium, 2004. Proceedings. – IEEE, 2004. – С. 55-63.

Model of ontology of the process simulator of primary processing of cocoa beans

Rafael' Kamdum

Master Student,
Moscow State University of Food Production,
125080, 11 Volokolamskoe shosse, Moscow, Russian Federation;
e-mail: kamdoumraphael5@gmail.com

Aleksei S. Maksimov

Professor,
Department of informatics and computer technology of food production,
Moscow State University of Food Production,
125080, 11 Volokolamskoe shosse, Moscow, Russian Federation;
e-mail: maksimov@mgupp.ru

Abstract

The article raises the issue of exploiting the semantic field as a business modeling tool. The concept and principle of operation of this new approach, which is called ontology, are described in detail. Ontologies are models in a semantic form; several programs have been developed for their implementation. Among these programs, the authors chose the Protégé platform because of its many benefits. After its relatively simple installation in the production environment, its operation allowed the creation of ontology. This was done by creating first ontology metadata, then class, subclass and disjoint classes, then relationships, sub-relationships and inverse relationships, after that the characterization and definition of areas and scope of the ontology were carried out, finally, the result of ontology was obtained and discussion followed.

For citation

Kamdum R., Maksimov A.S. (2022) Ekonomicheskie aspekty modeli simulyatora tekhnologicheskogo protsessa pervichnoi pererabotki kakao-bobov [Model of ontology of the process simulator of primary processing of cocoa beans]. *Ekonomika: vchera, segodnya, zavtra* [Economics: Yesterday, Today and Tomorrow], 12 (7A), pp. 145-152. DOI: 10.34670/AR.2022.77.66.016

Keywords

Ontology, model, semantic modeling industry, cocoa bean processing.

References

1. Abadi A., Sekkat S., Zemmuri E.M., Ben-Azza Kh. *Ispol'zovanie ontologii dlya podderzhki odnovremennogo proektirovaniya produkta i tsepochki ego postavok* [Using ontologies to support simultaneous product and supply chain design]. Available at: <http://ceurws.org/Vol-600/paper3.pdf> [Accessed 18/05/2022].
2. Bertran F. i dr. *Ontologii dlya modelirovaniya protsessov ukhoda v meditsinskikh uchrezhdeniyakh* [Ontologies for modeling care processes in medical institutions]. Available at: [.http://ceur-ws.org/Vol-600/paper3.pdf](http://ceur-ws.org/Vol-600/paper3.pdf).
3. Dzhon Sharlet, Odri Beniks, Ksav'e Eme, Dombr F. *Sozdanie ontologiya v OWL s Protégé 4/5* [Creating an ontology in

-
- OWL with Protégé 4/5]. Available at: https://www.researchgate.net/publication/318208041_Creation_d'une_ontologie_en_OWL_avec_Protege_45 [Accessed 12/05/2022].
4. Fortino V. *Vklad v ontologicheskoe modelirovanie informatsii na protyazhenii vsego zhiznennogo tsikla tovara* [Contribution to ontological modeling of information throughout the life cycle of a product]. Available at: https://www.researchgate.net/publication/261026131_Contribution_a_une_modelisation_ontologique_des_information_tout_au_long_du_cycle_de_vie_du_product [Accessed 14/05/2022].
 5. Gandon F. *Komp'yuternye ontologii* [Computer ontology]. Available at: <https://interstices.info/ontologies-informatiques>.
 6. Sharlet D., Steikhen O., Beniks O., Aleku Yu. *Ispol'zuite i stroite ontologii v meditsine. Primat terminologii* [The use and build ontologies in medicine. The primacy of terminology]. Available at: [researchgate.net](https://www.researchgate.net) [Accessed 12/05/2022].
 7. Xiaojun Y.E. *Modelirovanie i simulyatsiya proizvodstvennykh sistem: ob"ektno-orientirovannyi podkhod* [Modeling and simulation of production systems: an object-oriented approach]. Available at: <https://tel.archives-ouvertes.fr/tel-00821121> [Accessed 12/05/2022].
 8. Harrison, W. S., Tilbury, D. M., & Yuan, C. (2011). From hardware-in-the-loop to hybrid process simulation: An ontology for the implementation phase of a manufacturing system. *IEEE Transactions on Automation Science and Engineering*, 9(1), 96-109.
 9. Luukkainen, M., & Karhela, T. (2007, December). Ontology approach for co-use of 3D plant modelling and large scale process simulation. In *The 48th Scandinavian Conference on Simulation and Modeling (SIMS 2007)*; 30-31 October; 2007; Göteborg (Särö) (No. 027, pp. 166-172). Linköping University Electronic Press.
 10. Miller, J. A., Baramidze, G. T., Sheth, A. P., & Fishwick, P. A. (2004, April). Investigating ontologies for simulation modeling. In *37th Annual Simulation Symposium, 2004. Proceedings.* (pp. 55-63). IEEE.
-