



**JOHN ATANASOFF SOCIETY
OF AUTOMATICS AND INFORMATICS**

**JOHN ATANASOFF
CELEBRATION DAYS**

International Conference

**AUTOMATICS AND
INFORMATICS'2014**

PROCEEDINGS

**Edited by
Michail Petrov**

**Published by
JOHN ATANASOFF SOCIETY
OF AUTOMATICS AND INFORMATICS**

Bulgaria, Sofia, October 1-3, 2014

JOHN ATANASOFF SOCIETY OF AUTOMATICS AND INFORMATICS

Secretariat Adress

Bulgaria
1000 Sofia
108 Rakovsky str.
Tel. (+359 2) 987 61 69
Fax (+359 2) 987 61 69
e-mail: sai@infotel.bg
www.sai.infotel.bg
www.sai.bg

ТРАНСФОРМАЦИЯ НА UML ДИАГРАМА НА КЛАСОВЕТЕ В OWL ОНТОЛОГИЯ

TRANSFORMATION OF UML CLASS DIAGRAM TO OWL ONTOLOGY

Д. Гочева, И. Бачкова

Dept. of Industrial Automation, University of Chemical Technology and Metallurgy, Bul. Kl. Ohridski 8,
Sofia, Phone: +359 2 8163326, E-mail: [dani, idilia}@uctm.edu](mailto:{dani, idilia}@uctm.edu)

Abstract: One very promising way to achieve enterprise interoperability is to combine the approaches of software engineering and Artificial Intelligence. This paper presents a method for transformation of UML class diagram to OWL ontology. The suggested method is based on the meta-models of UML class diagram and OWL2. Different transformation rules are defined and used in order to enhance semantically the object models proposed in ANSI/ISA-S95 standard. The transformation process is illustrated with a simple example of ANSI/ISA-S95 based "Personell Model" into OWL Class. Finally some conclusions are made.

Keywords: object oriented models, ontology, UML, OWL, ANSI/ISA-S95, enterprise models, transformation

ВЪВЕДЕНИЕ

Интеграцията на данни, информация и знания в предприятията е важна, но изключително трудна задача в стремежа за постигане на оперативна съвместимост между информационните и управляващи системи на предприятията. Една от възможностите в тази насока са процесите на стандартизация за осигуряване на общи рамки и модели на базата на прилагането на най-добрите практики, чрез отделяне на информацията и знанията от конкретните търговски продукти и методи за реализация, предлагани от фирмите. В тази връзка международният стандарт ANSI/ISA S95 предлага обща терминология и съвместими референтни модели, които представят най-добрите практики за интеграция, обхващат целия жизнен цикъл на предприятията, приложими са за подобряване на съществуващите възможности за интеграция, без да зависят от степента на автоматизация в предприятията [1, 2]. В част 1 на стандарта са дефинирани различни модели, като функционален йерархичен модел, йерархичен модел на оборудването и функционален модел на данните. Последният е представен чрез набор от UML диаграми на класовете, съгласно нотациите в стандарта ISO/IEC 19501-1 Информационни технологии – Унифициран език за моделиране (UML) - Част 1: Спецификация [3]. Тези модели детайлизират информацията, която се обменя за това: (i) как да се произведе продукт, (ii) какви ресурси са необходими и налични за това, (iii) какво, къде и кога да се произведе и (iiii) какво е произведено, и какви ресурси са използвани. Дефинираните обектни модели се доразвиват във втората част на стандарта, където са специфицирани атрибутите за всички обекти, дефинирани в първата част. Дефинициите на атрибутите са абстрактни, което не ги свързва с определена реализация или типове данни.

Използваният в стандарта обектно-ориентиран език за моделиране UML е общоцелеви език за визуално моделиране, който представлява фамилия от графични означения, обединени в общ мета-модел, подпомагащ описанието и проектирането на обектно-ориентирани софтуерни системи [4, 5]. Унифицираният език може да бъде използван за моделиране на различни по вид, обем и предназначение системи: хардуерни и/или софтуерни, технически и/или обществени, реални и/или виртуални. Основните недостатъци на обектно-ориентираните модели

и в частност на UML са свързани с невъзможността за отразяване на семантиката на моделираната система, формализацията на моделите и възможността за логически извод. За разлика от UML, OWL се базира на дескриптивната логика и е предназначен да прави изводи относно генерализации и специализации между класовете, както и принадлежността на обектите към определен клас на базата на дефинираните ограничения върху свойствата на класа [6].

Съществува стремеж за разширяване на възможностите на UML по отношение на степента на формализация и отразяване на семантиката на моделирана област да се предприема трансформация на създадените UML модели и по-специално на модели, отразени с диаграми на класовете в езика за концептуализация на онтологии OWL. На базата на сравнителни анализи са предложени различни подходи за трансформация, като те могат да бъдат обобщени в две основни групи. Първата група е базирана на UML профил за онтологии и XML (XMLSchema, XSLT и XMI сериализация на UML модели) [7, 8]. Втората група подходи се отнася до последната версия OWL2 и обединява методи за автоматична трансформация, базирани на използването на метамодели, като например езиците за трансформация ATL, QVT-R [9] и др. Основен извод, който може да се направи на базата на анализа на тези методи е, че в момента не съществува автоматизиран и пълен метод за трансформация на UML диаграма на класовете в OWL онтология.

Трансформацията на предложените в стандарта ANSI/ISA S95 обектни модели в семантични е важна стъпка към постигане на оперативна съвместимост на информационните и управляващи системи на предприятията. В тази връзка в доклада е представен метод за трансформация на UML диаграма на класовете в OWL2 онтология с цел създаването на мета онтология на предприятията на базата на дефинираната терминология и съвкупността от обектни модели на стандарта ANSI/ISA S95. За целта са предложени правила на съответствие между използваните в моделите нотации от UML диаграмата на класовете и OWL, използван като език за представяне на мета онтология. В следващите раздели на доклада са представени последователно използваните за

целта мета модели на UML и OWL, върху чиято основа са изградени правилата на съответствие. Методът е илюстриран с пример за трансформация на ANSI/ISA S95 базиран UML модел в OWL клас. В края на доклада са представени някои изводи.

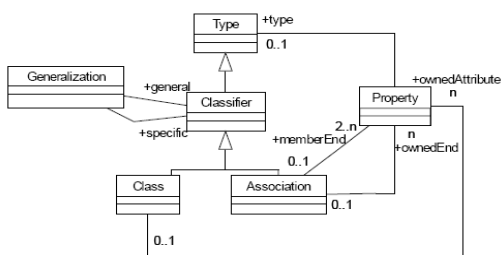
ОПИСАНИЕ НА МЕТА МОДЕЛИТЕ

Мета модел на UML диаграма на класовете. Мета моделът на UML диаграма на класовете представлява избрана част от модела на суперструктурата на UML и е представен на фиг.1. Диаграмата на класовете в UML моделира структурата на проектираната система и се базира на понятието клас (class). Класът в UML е вид класификатор (classifier) и включва съвкупност от обекти, които споделят обща структура и поведение, т.е. притежават едни и същи свойства (дефинирани като атрибути и операции), ограничения и семантика. Освен класове и обекти, диаграмата на класове включва и различни релационни връзки, които представляват специфичен вид логическа връзка в диаграмата. Основните видове релационни връзки могат да бъдат разпределени в 3 групи, както следва:

- Релационни връзки на ниво екземпляри – към тази група спадат релационните връзки „links” и „association”. Асоциацията („association”) специфицира семантична релационна връзка, която възниква между екземплярите на отделните класове, като към нея могат да бъдат прикачени роли, индикатори за собственост, множественост, видимост и други характеристики. Тя се среща в различни разновидности: бинарна, еднопосочна, двупосочна, агрегация и композиция. Последните две връзки са от типа „has a” и се използват при представяне на йерархии с цел редуциране на сложността. Връзката „links” е частен случай на асоциацията и представлява асоциация между два екземпляра.

Релационни връзки на ниво класове – към тази група спадат релационните връзки „generalization” и „realization”. Генерализацията (generalization) е таксономична релационна връзка от тип „is a” или позната като наследствена връзка и се използва за отразяване на общите характеристики на класовете. Реализацията (realization) е специализирана абстрактна релационна връзка между две групи от елементи на модела, едната от които представлява спецификация (доставчик), а другата представлява имплементация на последната (клиент).

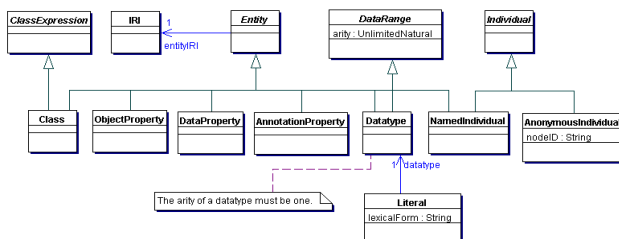
- Общи релационни връзки – към тази група спада релационната връзка „dependency” - зависимост. Тя се използва за да означава, че спецификацията или имплементацията на елемент или съвкупност от елементи изисква такава и на други елементи на модела. Това означава, че пълната семантика на зависимите елементи е или семантично, или структурно зависима от дефиницията на елементите доставчици.



Фиг.1: Мета модел на UML диаграма на класовете

Мета моделът на диаграмата на класовете в UML, показва, че между класовете „class” и „association”, които са подкласове на класа „type” (тип), не съществува директна връзка, а тя се реализира посредством класа „property” (свойство), който представлява структурна характеристика. Свойството може да бъде собственост на класа, но може и да съществува отделно от класа, като може да бъде насочено и не насочено. Крайщата на асоциацията са представени като свойства.

Мета модел на OWL онтология. За целите на създаването на мета онтология на предприятията е предпочетена версията на езика OWL2-DL, която гарантира възможности за мащабируеми разсъждения (reasoning), без да се намали изразителните възможности (експресивността) на езика OWL. OWL2-DL разкрива потенциала на дескриптивната логика в индустриалните проекти. Формалната семантика на OWL осигурява логически анализи, изводи и верификация, които подпомагат изграждането, поддръжката и използването на домейн модели за различни цели. Класовете (Class) (фиг.2) представят групи от индивиди; данните (Datatype) са литерали (стрингове или числа), обектните свойства (ObjectProperty) и свойствата тип данни (DataProperty) се използват за представяне на релациите в описваната област; анонционните свойства (AnnotationProperty) служат за асоцииране на описателна информация към основните елементи на онтологията; именуваните индивиди (NamedIndividual) се използват за представяне на реални обекти в описваната област. Освен чрез именувани индивиди (NamedIndividual), OWL 2 предоставя възможност за дефиниране на анонимни индивиди (AnonymousIndividual). Класовете са основни понятия в областта, които изграждат структурата на различни таксономични дървета. Разграничават се два типа свойства: свойствата „тип данни” (DataProperty) представляват релации между индивиди на класовете и типовете данни на XML Schema, и обектни свойства (ObjectProperty), които представят релации между индивиди на два класа. При дефиниране на свойствата съществуват няколко начина за ограничаване на релацията. Могат да бъдат специфицирани домейн (domain) и обхват (range). Свойството може да бъде дефинирано като специализация (subProperty) на съществуващо свойство. Възможно е да се дефинира множественост на свойствата (Cardinality) и различни характеристики (ограничения) на свойствата (като например Inverse, Transitive, Functional, InverseFunctional), което осигурява мощен механизъм за засилени разсъждения върху определено свойство.



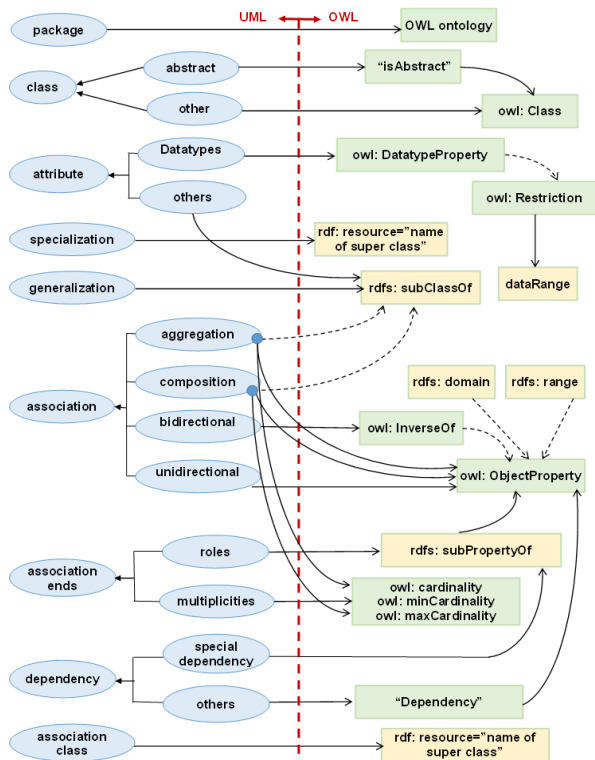
Фиг.2: OWL мета модел

ДЕФИНИРАНЕ НА ПРАВИЛА НА СЪОТВЕТСТВИЕ

Като се вижда от фиг.1 и фиг.2, и представените анализи, формалната структура на UML е доста по-различна от тази на OWL. Целта е да се дефинират съответствия между модел на ниво M1/M0 в UML диаграма на класовете и

еквивалентния модел в OWL, така че UML моделът на ниво M1 да бъде трансформиран в OWL. Дефинираните правила на съответствие са илюстрирани на фиг.3.

Дефинирането на съответствието е направено с цел да се съхрани семантиката на отделните елементи. И двата езика са базирани на класове. Класът в UML е по-обща структура, като една от неговите употреби е като съвкупност от екземпляри. В тази си роля UML класът е съвкупност от нула или повече екземпляри на обекти на ниво M0. Екземплярът се състои от съвкупност от слотове, съдържащи стойност, извлечена от типа на свойството на слота. Екземплярът е свързан с един или повече класификатори. Класът в OWL е съвкупност от нула или повече индивиди. Налице са едва доловими различия между OWL класове и UML класове, но тъй като в OWL 2, за разлика от UML не съществува концепцията „уникалност на наименованията“ (UNA), необходимо е класовете да бъдат разделени чрез owl:disjointWith. Генерализацията (generalization) от UML се съпоставя еднозначно чрез rdfs:SubClassOf. Релационните връзки между класовете в OWL се представят чрез два вида свойства: свойства тип данни (DataProperty) и обектни свойства (ObjectProperty), които имат различни съответствия в UML модела. Едно от съответствията е M2 асоциация (ownedAttribute) между Class и Property, което генерира представянето на класа като пакет от собствени атрибути. M1 екземпляр на клас ownedAttribute Property ще се трансформира като свойство, чийто домейн е Class и чийто обхват е типа на свойството. Съответствието на UML екземплярът ownedAttribute в OWL е обектно свойство (ObjectProperty), ако става въпрос за UML клас, а във всички останали случаи - свойство тип данни (DataProperty).



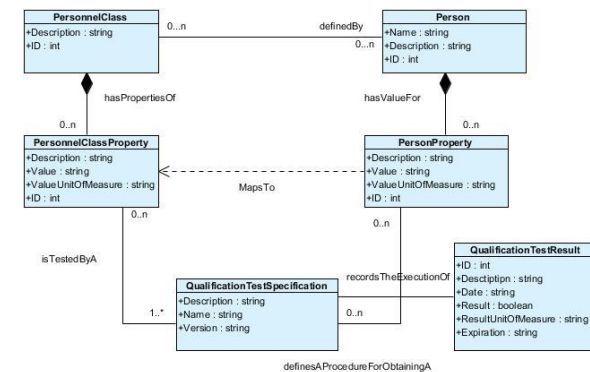
Фиг.3: Илюстрация на правилата за съответствие

Според концепцията за „отворена среда“ (OWA) в OWL за свойствата не е задължително да имат специфицирани домейн (domain) и обхват (range), но за да се направи съответствие с UML („затворена среда“ - CWA), това е

задължително. Тъй като в OWL2 свойствата не се „притежават“ от класовете, те трябва да бъдат допълнително ограничени. Това става чрез rdfs:SubClassOf, като по този начин се създава виртуален родителски клас, от който се наследяват ограниченията. В UML множественост може да се дефинира за асоциация и за атрибут. OWL2 също притежава аксиоми за дефиниране на множественост както за обектните свойства, така и за свойствата от тип данни. Ако бинарна UML асоциация има множественост на насочения край, съответстващото OWL свойство ще има същата множественост. Ако дадена бинарна UML асоциация има множественост на двата края, то съответното OWL свойство ще бъде инверсна двойка, всяка от които притежава една от декларациите за множественост. За дефиниране на съответствие на агрегация в OWL2 се използва rdfs:SubClassOf и ограничения за множественост.

ПРИМЕР: ТРАНСФОРМАЦИЯ НА „PERSONNEL MODEL“

Примерът, който е използван за демонстрация на метода за трансформация е UML модела „Personnel model“, който е основен модел от представените в стандарта ANSI/ISA-S95 обектни информационни модели на предприятията [1, 2]. Моделът е показан на Фиг.4 и включва информация за конкретния персонал, класовете персонал, както и квалификацията на персонала.



Фиг.4: Обектен модел на „Personnel model“

Класът „Personnel Class“ представя обединение на хора със сходни характеристики за целите на дългосрочното и краткосрочното планиране. Атрибутите на класа „Personnel Class“ са ID и Description. Характеристиките на класа „Personnel Class“ са отразени в класа „Personnel Class Property“ с атрибути ID, Description, Value и Value Unit of Measure. Всеки индивид от клас персонал трябва да има нула или повече утвърдени характеристики, т.е. релационната връзка между двата класа е композиция. Конкретните личности са представени като лица в клас „Person“ посредством атрибутите ID, Description и Name, като едно лице може да бъде член на нула или повече класове на персонала посредством асоциативна релационна връзка. Характеристиките на лицата са описани в клас „Person Property“, чийто атрибути съвпадат с тези на класа „Personnel Class Property“, но включват актуални данни за конкретното лице. Релационната връзка между тези два класа е тип зависимост, докато тази с класа „Person“ е от тип строга агрегация, т.е. композиционна. Представянето на тестовете за квалификация е чрез класа „Qualification Test Specification“, който може да бъде асоцииран с един или и двата класа „Personnel Class Property“ и „Person Property“. Тези релационни връзки се

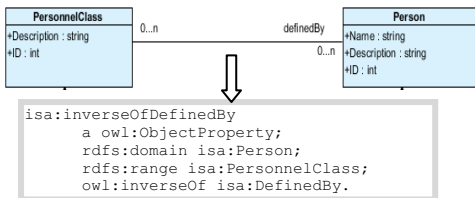
използват, когато се изисква тест за квалификация, за да се гарантира, че дадено лице е с необходимата специалност и/или достатъчно опит за специфични операции. Спецификацията на тестовете за квалификация се отнасят за едно или повече характеристики. Атрибутите на клас „Qualification Test Specification“ са: Name - идентификация на теста, Version - версия на теста и Description - описание на теста. Резултатите от теста за отделно лице се отразяват в клас „Qualification Test Result“, който включва следните атрибути ID, Description, Date, Result, Result Unit of Measure и Expiration. Този клас е свързан с класа „Qualification Test Specification“ посредством асоциативна релационна връзка.

Трансформираният в OWL обектен модел е представен графично на фиг.5, като за целта са използвани възможностите на продукта TopBraid Composer, който позволява използването на графични нотации за изобразяване на OWL класове, свойства и аксиоми. Трансформацията е извършена съгласно представените на фиг.3 правила за трансформация. Моделът “Personnel model” е трансформиран и представен като OWL клас със същото име. UML класовете са трансформирани в OWL класове, представени като правоъгълници, които се явяват подкласове на OWL класа “Personnel model”, като връзката между тях, отразена на фиг.5 като релациите между класовете от типа `rdfs:subClassOf` са означени с черни дебели линии и стрелки, тънките линии означават релации (`owl:ObjectProperty`) между класовете, дефинирани чрез `rdfs:domain` и `rdfs:range`. Атрибутите на класовете са трансформирани в OWL свойства тип данни (`owl:DatatypeProperty`), като типа на данните е дефиниран чрез `rdfs:range`.



Фиг.5: Обектен модел на “Personnel model”

Двупосочната асоциативна релационна връзка между класовете се трансформира в OWL обектно свойство (`owl:ObjectProperty`), като допълнително се използва свойството `owl:InverseOf`, както и въвеждането на ограничения посредством използването на `rdfs:domain` и `rdfs:range`, както е показано на фиг.6.



Фиг.6: Трансформация на бинарна, двупосочна UML асоциация

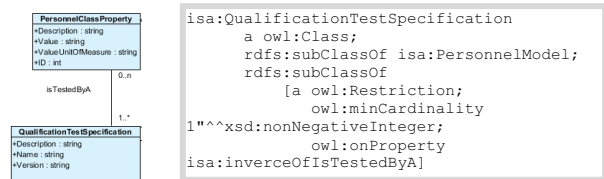
На фиг.7 е илюстрирана трансформацията на UML композиционна релационна връзка, която се реализира с помощта на обектно свойство със съответните област и обхват. В OWL се предполага, че всеки индивид от даден клас може да има произволен брой (нула или повече)

стойности за определено свойство (релация или данни). Множественост `0..n` не се означава на диаграмите.



Фиг.7: Трансформация на UML агрегация (композиция)

Множественост на релациите в UML се означава до релациите между класовете (`owl:ObjectProperty`) (фиг.8) и се ограничава чрез `owl:minCardinality`.



Фиг.8: Трансформация на UML асоциация (множественост)

ЗАКЛЮЧЕНИЕ

Предложен е метод за трансформация на обектно-ориентирани UML модели на класовете в OWL2-DL онтологии с оглед на постигането на семантична оперативна съвместимост. Този метод е в основата на създаването на мета онтология, базирана на моделите, залегнали в стандарта за интегрирани системи за производство и управление ANSI/ISA-S95. За разлика от обектно-ориентиранияте модели, предложената мета онтология притежава следните предимства и предоставя следните възможности: използване на логически анализ (reasoning) върху моделите и данните, улеснява достъпа, интеграцията, обмена и извличането на информация от структурирани и неструктурирани данни, съхранявани в хетерогенни и разпределени източници, предоставя семантичен, гъвкав, устойчив и „богат“ по отношение на възможности за изразяване модел на данните, многократно използваем и приспособим при изменения на изискванията.

ЛИТЕРАТУРА

1. ANSI/ISA-S95.00.01-2000, Enterprise-Control System Integration, Part 1: Models and Terminology”, American National Standard, ISBN: 1-55617-727-5, 2000.
2. ANSI/ISA-S95.00.02-2001, Enterprise-Control System Integration Part 2: Object Model Attributes” ISBN: 1-55617-773-9, 2001.
3. ISO/IEC 19501-1 Information Technologies – Unified Modeling Language (UML) - Part 1: Specification.
4. OMG (2014), UML, Infrastructure Specification, Version 2.4.1, <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF>.
5. OMG (2014), Unified Modeling Language, Superstructure, Version 2.4, <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>.
6. W3C OWL Working Group, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition) W3C Recommendation, 11 December 2012, <http://www.w3.org/TR/owl2-syntax>.
7. Gašević D. V., Djurić D., Devedžić V. (2005), Bridging MDA and OWL ontologies, *Journal of Web Engineering*, Vol.4, No.2, pp.118-143.
8. Viademonte S., Cui Z., Deriving OWL Ontologies from UML Models: an Enterprise Modelling Approach, on-line available on: http://www.researchgate/profile/Sergio_Viademonte.
9. Zedlitz J., Luttenberger N. (2014), Conceptual Modelling in UML and OWL-2, Zedlitz J., Luttenberger N., *International Journal of Advances in software*, vol.7, No 3&4, pp.182-196.