СЕМАНТИЧНИ МЕТОДИ ЗА ОПЕРАТИВНА СЪВМЕСТИМОСТ НА ПРЕДПРИЯТИЯТА

SEMANTICS METHODS FOR INTEROPERABILITY OF ENTERPRISES

гл. ас. Гочева Д. Г., проф. д-р Бачкова И. А.
Химикотехнологичен и металургичен университет – София, България
{dani, idilia}@uctm.edu

Abstract: *Interoperability of enterprises is a challenging task whose successfully solution depends on the progress and development of very effective technological spaces. Proposed is a conceptual framework of approach for achieving interoperability, based on models, methods and tools of suitably chosen technological areas, for which the priority areas of application are analyzed and established. The main aim of the proposed paper is to collate and summarize the main methods supporting the interoperability of enterprises and underpin the approach. Presented is the application of the considered combined method for information retrieval and decision making.*

Keywords: SEMANTICS METHODS, MODELING, INTEROPERABILITY, ONTOLOGY, TECHNOLOGICAL SPACES

1. Увод

Постигането на оперативна съвместимост на предприятията е доста сложна и амбициозна задача и въпреки сериозния напредък в областта на информационните и комуникационни технологии, софтуерното инженерство и стандартизацията в тази област все още постигнатите резултати са незадоволителни. Идеята за моделно управлявана оперативна съвместимост (MDI), отчита необходимостта от използване на онтологии и семантични анотации, но в крайна сметка тази концепция е развита и приложена най-вече на ниво данни и използването на UML за представяне на онтологии, което ограничава подхода само до по-елементарни онтологии. Възможностите за логически изводи и разсъждения са твърде ограничени.

Цел на доклада е да представи онтологично управляван подход за оперативна съвместимост на предприятията, акцентирайки върху систематизацията на заложените в него методи, отразяващи различни аспекти на оперативната съвместимост и най-вече върху семантичната група от тези методи.

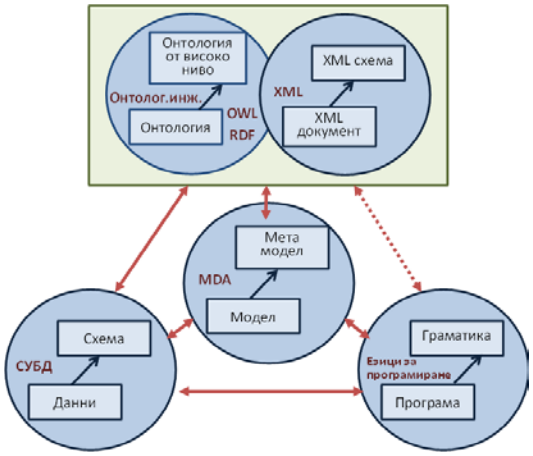
Докладът е представен в 4 части. В следващата част е направен кратък анализ на предлагания подход, в чиято основа е концепцията на технологичните пространства [1]. В третата част на доклада са систематизирани и анализирани използваните методи, а в четвъртата е представено прилагането на два от методите при решаването на конкретни задачи по постигане на оперативна съвместимост.

2.Онтологично управляван подход за оперативна съвместимост на предприятията

Предложеният подход за оперативна съвместимост в предприятията се базира на използване на езици и средства от различни технологични пространства. Технологичното пространство е работен контекст с набор от свързани понятия и утвърдени средства за моделиране [1]. Използваните в рамките на подхода технологични пространства са (фиг.1):

- Технологично пространство на метамоделите, представлявано от MDA на OMG, накратко ТП на „MDA" (Model Driven Architecture);
- Технологично пространство на онтологичното инженерство, представлявано от онтологичните езици OWL, RDF, накратко ТП на „RDF" (Resource Description Framework);
- Технологично пространство на СУБД, наричано накратко ТП на „RDB" (Relational Database);
- Технологично пространство на XML, наричано накратко ТП на „XML" (Extensible Markup Language);
- Технологично пространство на езиците за програмиране, наричано накратко ТП на „EBN" (Backus-Naur Form).

При разработката на подхода се тръгва от тезата, че проблемите, свързани с липсата на оперативна съвместимост могат да бъдат решени с използване на методи и средства от различни технологични пространства. Характерно за тези технологични пространства е, че те могат да решават едни и същи задачи с различен успех. Анализът и оценяването на характеристиките и свойствата на всяко от изброените по-горе технологични пространства е от изключително значение за тяхното успешно включване в решаване на проблемите на оперативната съвместимост. Между отделните технологични пространства съществуват еднопосочни и двупосочни връзки, които също са разнообразни и притежават определени свойства. По този начин могат да се създадат различни методи, чрез които се извършват с лекота операции и действия в едно технологично пространство, а след това получените резултати биват импортвани в други такива, с оглед на тяхното по-нататъшна и по-ефективна обработка и използване.



Фиг.1: *Подход за онтологична оперативна съвместимост в предприятията*

Разработеният подход удовлетворява поредица изисквания, покриващи следните важни аспекти на оперативната съвместимост:

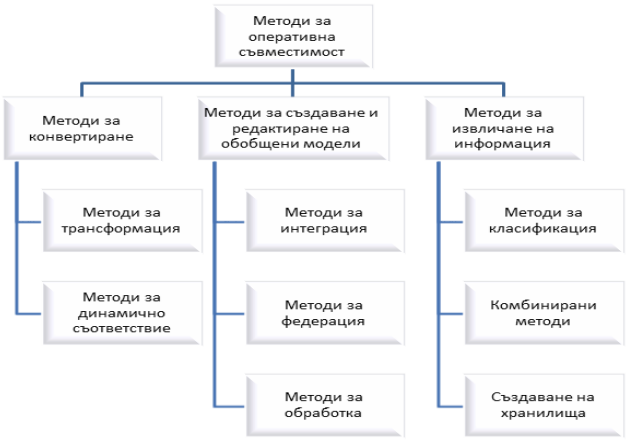
- Гарантира свързване и/или сливане на коренно различни схеми на данни и данни, както и едновременното търсене на информация в различни източници (информационни системи). Това се постига с използването на семантични модели, в основата на които е глобалната, единна схема за именуване - Uniform Resource Identifiers (URIs), най-значимото подмножество, на която са Uniform Resource Locators (URLs) на HTTP и използването на RDF/OWL модели. URIs идентифицират еднозначно ресурсите и представят явно смисъла на данните чрез свързването им в множества от домейн понятия.

- Не налага замяна на използваните технологии и приложения, а предлага само тяхното надграждане. Осигурено е съвместното приложение на утвърдените вече технологии за управление на данни и знания, за моделиране на приложения и вземане на решение, използвани на различни нива в рамките на предприятията.
- Осигурява възможности за достъп, обмен и извличане на информация и от неструктурирани данни, съхранявани в хетерогенни и разпределени източници.
- Осигурява възможност за създаване на семантични хранилища, които комуникират с източниците на данни, като за разлика от по-широко използваните в момента складове за данни не допускат дублиране на данни и загуба на метаданни.
- Подходът се базира на споделеното разбиране за конкретни области във вид на формален и машинно обработваем модел на областта. Това е свързано със създаването на лексиката на конкретна област и формално дефиниране смисъла на термините, на базата на логика чрез:
  - Лексика, използвана за описание на конкретен изглед на конкретна област;
  - Явна спецификация на смисъла на термините в лексиката (чрез класификации и релации);
  - Дефиниране на ограничения, улавящи допълнителни знания за моделираната област.
- Гарантира интеграцията на средства за логически анализ върху схемата на данните, върху самите данни и съответните модели. Едни от най-важните логически анализи са свързани с проверки за противоречивост в логическия модел (Consistency checks), с възможността един клас да не може да има индивиди (Satisfiability checks), с „предсказване“ на йерархични структури (Classification) и др. Софтуерните средства за логически анализ (reasoning) са независими от приложенията и подпомагатпроцесите на изграждане и поддръжка на споделени онтологии чрез разкриване на несъответствия, разкриване на скрити зависимости и релации, избягване на излишна информация, осъществяване на добро класифициране.
- Осигурява интеграция на мета модели с конкретни домейн модели. По този начин се улеснява използването на стандартизирани, референтни модели и многократно използваеми модули, изисквания, дефинирани в архитектурите на [2, 3, 4].
- Предоставя удобен, гъвкав, устойчив и „богат“ по отношение на възможностите за изразяване модел на данните, който е многократно използваем и приспособим при изменения на изискванията
- Позволява разделяне на оперативните знания от знанията в конкретна области. Това дава възможност на потребителите да споделят и използват многократно едни и същи набори от данни в различни приложения и платформи, като това се постига чрез два основни принципа на проектиране: отделянето на моделите на знания от приложенията и интегрирането на модели на знания чрез многократно използване на модули и възможности за разширение на създадените модули. Разделянето на моделите от приложенията е важно условие за осигуряване на възможности за споделяне на данни и многократно използване на данните в различни приложения и среди. При това разделяне, бизнес логиката „излиза“ от приложенията и „влиза“ в моделите на знания чрез онтологии и правила.
- Гарантира модулност и многократна използваемост, стандартизация, както и възможности за разширяване и модификация.
- Приложим е на всички етапи от разработката на информационни и управляващи системи. Започвайки от онтологичен модел на областта, различни видове онтологии на процеси, задачи, интерфейси, за дефиниране на интелигентни агенти, които използват онтологията за да комуникират помежду си в разпределени хетерогенни среди, и се стигне до процеса на управление на качеството и задачите за диагностика.

Важен фактор за постигане на изброените по-горе аспекти на оперативна съвместимост, освен използваната концептуална рамка, е и съвкупността от методи, които поддържат съответните технологични пространства и тези, които реализират връзките между тях. Част от тях са предмет на систематизация и анализ в следващия раздел на доклада.

3. Методи за постигане на оперативна съвместимост

Постигането на оперативна съвместимост чрез използване на семантични технологии може да се представи като последователност от три основни процеса: конвертиране на данните от хетерогенни източници (релационни бази данни, XML, таблици, документи), обработка на интегрирания модел (обединяване, редактиране, логически анализ) и извличане на информация (конвертиране, визуализация). По време на трите етапа се използват различни методи, които позволяват да се контролират данните, които се интегрират, да се сортират данните, да се отхвърлят повторенията, да се извлича информация, да се осъществява логически анализ и верификация. Систематизация на разработените и прилагани методи е представена на фиг.2.

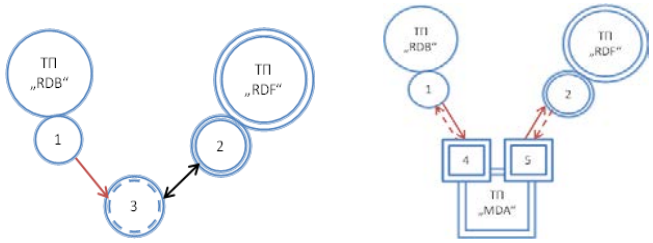


Фиг.2: *Систематизация на методите*

Методите за конвертиране изграждат мостовете между отделните технологични пространства. Така например съществуват различни методи и софтуерни средства за конвертиране на релационни бази данни, електронни таблици, XML файлове, UML файлове, различни онлайн ресурси и др. в RDF/OWL формат. На фиг.3а и фиг.3б са представени илюстрации на двете групи методи, които се прилагат успешно – трансформация и динамично съответствие. Трансформацията се базира на използване на концепцията, методите и средствата, залегнали в ТП на „MDA” и използване на метамодели на различните ТП, както и формулиране на правила за трансформация на мета ниво, както е показано на фиг.3б. При методите на динамичното съответствие пък се използват специални езици за определяне на съответствията между модел (база данни) и RDF Graph (фиг.3а-(3)). За целите на задачата за динамично съответствие между релационна база данни и онтология е използван езика R2RML, като данните могат да бъдат оставени в източника и извличани при поискване или преместени във временни RDF хранилища. Методът може да се приложи към данни от измервания в реално време, към данни от производството, външни и собствени бази данни. Тези два метода се прилагат доста успешно и при двупосочно конвертиране на модели между ТП на „MDA” и ТП на „RDF”. На фиг.4а е илюстриран метод за динамично съответствие на UML модели в OWL онтологии, който е използван при създаването на референтна мета онтология на предприятията.

Методите за създаване и редактиране на обобщени модели се състоят от методи за интеграция, федерация и обработка.

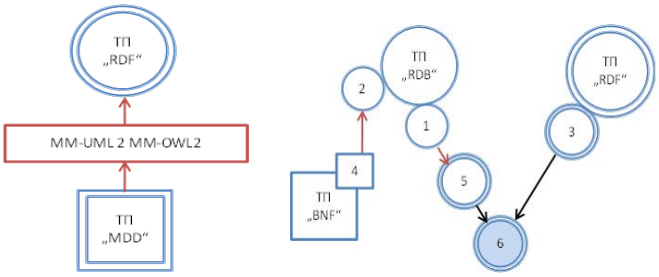
Интеграцията може да се осъществи по различни начини, преминавайки през различни етапи с използване на различни езици, технологии и механизми. Онтологично базираната интеграция обхваща процеси, подобни на ETL трансформациите: (E)xtracte от източник на данни, (T)ransforme в RDF модел (целева онтология) и (L)oade в онтологични хранилища на данни. Основните разлика е, че при онтологично базираната интеграция могат да се използват както структурирани, така и неструктурирани данни, а семантиката в онтологията може да се зададе в процеса на трансформацията, или в последствие при обработката на онтологичния модел чрез операциите върху онтологии: обединяване, съпоставяне, подравняване, интегриране. Като целева онтология може да се използва референтна онтология. При промяна в данните, трансформацията се повтаря. На фиг.4б е илюстриран метод на онтологична интеграция, представящ интеграцията на резултати, получени в ТП на „BNF“, които посредством Data Base Toolbox се съхраняват в ТП на „RDB” като база данни (2) и заедно с база данни (1) могат да бъдат интегрирани в онтология (5), която да бъде обединявана, съпоставяна, подравнявана и интегрирана с (3) в интегрираната онтология (6).



а) б)

Фиг.3: *Илюстрация на методите „динамично съответствие“ и „трансформация“ между различни ТП*

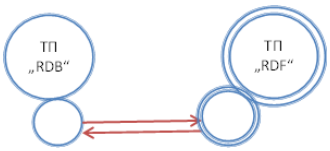
В основата на онтологично базираната федерация на данни е SPARQL [5], който е протокол и език за заявки върху данни, записани или представени в RDF формат. Чрез езика SPARQL могат да изпълняват заявки за търсене в различни източници на данни, да се конструират модели, да се задават структури и правила. Възможностите за онтологично базирана федерация значително нарастват с появата SPARQL 1.1. [6] като език за създаване на правила. Чрез клаузите за обединяване и редактиране на графи GRAPH и SERVICE, SPARQL 1.1 осигурява възможности за запитвания върху различни източници, еднократно чрез заявки към източници, намиращи се в хранилище на данни, или чрез SPARQL крайни точки (SPARQL endpoints). Всеки източник представлява именуван граф чрез базов URI: релационни бази данни, таблици или RDF онтологии се третират еднакво като графи. По този начин, външни източници на данни се запитват чрез клаузата SERVICE. Едновременното използване на SERVICE и GRAPH е възможно в рамките на една заявка. Този метод е илюстриран на фиг.5.



а) б)

Фиг.4: а) *Илюстрация на метода “UML2OWL” и б) метод за интеграция*

Методите за обработка на обобщени модели се прилагат както върху интегрирани, така и федерирани модели и целят да се справят с терминологичните различия, повторенията на данните и създаването на правила за съответствие между различните източници в обобщения модел.



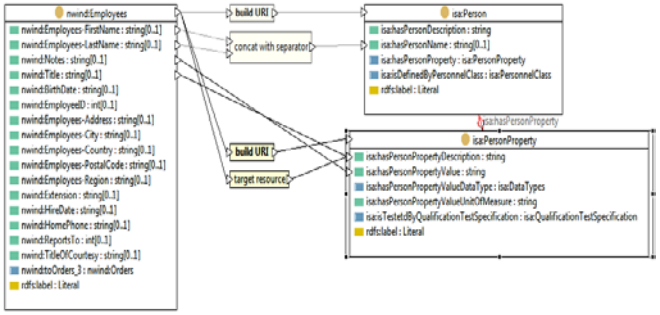
Фиг.5: *Илюстрация на метод за федерация*

Методите за извличане на информация включват методи за класификация, методи за създаване на хранилища и комбинирани методи, които представляват автоматизирани вериги от методи за извод и вземане на решение. В основата на тези методи са справки за извличане на информация от интегрирани модели – директно чрез SPARQL или уеб услуги, а също и чрез трансформация на SPARQL в SQL към релационни данни. По-важни технологии, изградени на основата на SPARQL са: SPARQL Notation Inferencing (SPIN) [7], SPINMap, SPARQLMotion [8]. Лексиката на SPIN (spin:constraint, spin:rule, spin:function) осигурява свързване на класове със SPARQL заявки и създаване на обектно-ориентирани шаблони за многократно използване. SPINMap е SPARQL базиран език за съпоставка (mapping) на RDF/OWL онтологии. SPARQLMotion е визуален скриптов език за изпълнение на повтарящи се задачи за обработка на данни и задачи, свързани с интеграцията на данни. Интегрирането се осъществява чрез диаграми на потоците от данни. Всеки конкретен алгоритъм за интеграция на данни може да се запише като SPARQLMotion скрипт за многократно изпълнение. По-долу е представен пример, онагледяващ използването на комбинирания автоматичен метод. Други практически примери илюстриращи представените по-горе методи могат да се видят в [9-12].

4. Приложение на комбиниран метод за извличане на информация

Идеята за онтологично базирана интеграция може да се представи като последователност от три основни етапа: импорт (от релационни бази данни, от XML, от таблици и документи), обработка на интегрирания модел (трансформиране, редактиране) и експорт (конвертиране, визуализация). По време на трите етапа се използват семантични технологии и средства за логически анализ (reasoning), които позволяват да се контролират данните, които се интегрират, да се отхвърлят повторенията, да се сортират данните. Тъй като данните и схемите на данните се съхраняват по един и същи начин, е възможно динамично по един и същи начин да се обработват и данните, и схемите на данните.

На фиг.5 е представен вариант на интеграция чрез технологията SPINMap в продукта TobBraid Composer Maestro Edition. В процеса на интеграция се използват част от данните и структурата на релационна база данни, представена като онтология, и референтна, съвместима с ANSI/ISA-S95 мета онтология. Данните от клас nwind:Employees, получени от релационната база данни се трансформират в два класа от мета онтологията: клас „Персонал“ (isa:Person) и клас „Характеристики на Персонала“ (isa:PersonProperty). При създаване и записване на данни в клас “Персонал”, някои от атрибутите от релационната база данни са обединени (FirstName и LastName), други са модифицирани. Свойствата от клас nwind:Employes от релационната база данни се съпоставят на свойства, дефинирани в isa:Person и isa:PersonProperty. Част от свойствата на клас nwind:Employees стават свойства на клас isa:Person, а други – на клас isa:PersonProperty. Записите от базата данни изцяло се трансформират като индивиди в онтологията.

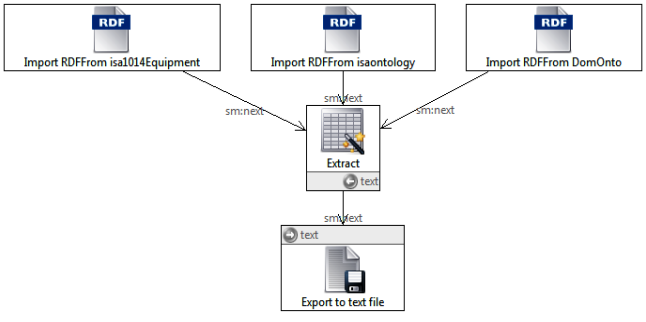


Фиг.5: *Онтологична обработка на интегрирания модел чрез технологията SPINMap*

За автоматизиране на последователността от процеси при онтологично базираната интеграция, обработка и извличане на данни чрез технологията SPARQLMotion е създаден примерен темплейт за многократно използване. Използвани са модули от три типа: ImportModules (sml:ImportFromLocalModules), ProcessingModules (sml:CreateSpreadsheet) и ExportModuls (sml:ExpotrYoTextFile). Чрез създадения скриптов файл се осъществява обединяване на три онтологии, извлича се информация, резултатите от която се преобразуват и записват в текстов файл. Както е показано на фиг.6 в интеграцията се включват: домейн онтология на оборудването в завод за горещо валцуване (isa1014Equipment), мета онтология (isaontology), и домейн онтология на производството в завод за горещо валцуване (DomOnto). Модулът за извличане на информация Extract извлича данни (индивиди) чрез следната SPARQL заявка:

```
SELECT DISTINCT ?ProductionSchedule ?Order
?EquipmentSpecProdDef ?EqSegmentProcessSegment ?Segment
WHERE {?ProductionSchedule
sa:isMadeUpOfProductionRequests ?Order .
?EquipmentSpecProdDef
isa:hasEquipmentSpecificationEquipment ?n .
?EqSegmentProcessSegment
isa:hasEquipmentSegmentSpecificationEquipment
?Segment.}
```

След изпълнение на модул sml:ExpotrYoTextFile, в текстов файл се записва информация за изпълнение на планов график ProductionSchedule_123, в който са включени три производствени заявки (ProductionRequest_1, ProductionRequest_2, ProductionRequest_3), като за изпълнение на графика в процесния сегмент, касаещ оборудването (EquipmentSegmentSpecification_1) е конкретизиран агрегат от модела на оборудването (Методична_пещ_1).



Фиг.6: *SPARQLMotion script за автоматизиране на онтологично базирана интеграция на данни*

Резултатите се записват в текстов файл от вида, представен на фиг.7.

ProductionSchedule	Order	EquipmentSpecProdDef	EqSegmentProcessSegment	Segment
:ProductionSchedule_123	:ProductionRequest_2	:Equipment_Specification_1	:EquipmentSegmentSpecification_1	:Методична_пещ_1
:ProductionSchedule_123	:ProductionRequest_1	:Equipment_Specification_1	:EquipmentSegmentSpecification_1	:Методична_пещ_1
:ProductionSchedule_123	:ProductionRequest_3	:Equipment_Specification_1	:EquipmentSegmentSpecification_1	:Методична_пещ_1

Фиг.7: *Текстов файл с резултати*

5. Заключение

Постигането на оперативна съвместимост в предприятията, изразяваща се с възможността, различни информационни системи, приложения и услуги да комуникират, споделят и обменят данни, информация и знания по прецизен и ефективен начин, както и да се интегрират с други системи, приложения и услуги, за да се доставят нови информационни продукти и услуги, е достижимо предизвикателство на базата на предложения онтологично управляван подход. Използвайки онтологии и базирани на тях системи бази данни и системи бази знания, могат да се извличат знания, да се систематизират и многократно използват тези знания, когато и където е нужно, като по този начин се съкращава времето за разработка и цената на информационните и управляващи системи.

6. Литература

1. Kurtev I., Bézivin J., Aksit M. (2002), Technological Spaces: an initial appraisal, CoopIS, DOA'2002 Federated Conferences, Industrial track, Irvine.
2. GERAM (1999), Generalized Enterprise Reference Architecture and Methodology, Version 1.6.3, IFAC-IFIP Task Force on Enterprise Integration.
3. Microsoft (2013), Vision for an Upstream Reference Architecture, White Paper, www.microsoft.com/mura.
4. ANSI/ISA–S95.00.01–2000, Enterprise - Control System Integration, Part 1: Models and Terminology", American National Standard, ISBN: 1-55617-727-5, 2000.
5. Prud'hommeaux E., Seaborne, A. (2008), SPARQL Query Language for RDF - W3C Recommendation 15 January, http://www.w3.org/TR/rdf-sparql-query/
6. Harris S., Seaborn A. (2013), SPARQL 1.1 Query Language, W3C Recommendation, 21 March, http://www.w3.org/TR/sparql11-query.
7. Knublauch H., Hendler J., Idehen K. (2011), SPIN - Overview and Motivation, W3C Member Submission 22 February.
8. Knublauch H. (2010), SPARQLMotion, Version 2.1.0, February 01, sparqlmotion.org.
9. Gocheva D., I. Batchkova (2011), Ontology Based Approach for Achieving Interoperability of Manufacturing Execution Systems, Proceeding of the Anniversary Scientific Conference with International Participation "40 Years Department of Industrial Automation", 18 March 2011, UCTM, Sofia, pp.153–158.
10. D. Gotcheva, I. Batchkova (2012), Semantic approaches for interoperability in enterprise engeneering, in Proceedings of the International Conference "Automatics and Informatics'2012", 03-05 October, Sofia, pp.262-266.
11. D. Gocheva, D. Georgiev, I. Batchkova (2013), Ontology-based data management in manufacturing execution systems, Scientific Conference with international participation "60 Years UCTM", 04-05. June, Sofia.
12. Д. Гочева, И. Бачкова (2014), Онтологично управляван подход за оперативна съвместимост в предприятията, Материали от 22-ри международен симпозиум "Управление на топлоенергийни обекти и системи", 8-9 май, г. Банкя, стр.35-38.