



Artur Machura

Uniwersytet Ekonomiczny w Katowicach
Wydział Informatyki i Komunikacji
Katedra Informatyki
artur.machura@uekat.pl

ZAŁOŻENIA DLA PROCESOWEGO WYKORZYSTANIA JĘZYKA OntoUML

Streszczenie: W artykule zaproponowano wykorzystanie języka specyfikacji ontologii OntoUML w procesie wytwarzania ontologii UPON (Unified Process Ontology Development). Wyniki pracy powinny w rezultacie umożliwić procesowe i bardziej przewidywalne wytwarzanie ontologii dziedzinowych w ramach m.in. procesu wytwarzania oprogramowania ODS (Ontology Driven Software Development), tym samym wychodząc naprzeciw ewentualnym problemom jakościowym ontologii dziedzinowych (jakość jest zwykle zawdzięczana uporządkowaniu zadań projektowych). Artykuł stara się wypełnić lukę w literaturze przedmiotu. Przy tym ułatwić dostarczanie wysokiej jakości ontologii dziedzinowych do procesu wytwarzania oprogramowania ODS. Oryginalność artykułu wynika z uspoźnienia języka OntoUML z procesem UPON.

Słowa kluczowe: OntoUML, UPON, ODS.

JEL Classification: A12, C8.

Wprowadzenie

Wytwarzanie ontologii powszechnie, ale również ewoluje, na co ma wpływ złożoność systemów w ogóle, natomiast w wykorzystanym przez artykuł przykładzie – złożoność systemów organizacji społecznych. Powodu takiego stanu rzeczy można doszukiwać się w definicjach ontologii wyszczególniających istotne pojęcia, takie jak: konceptualizacja, formalizacja czy jawność. Analiza i korelacja popularnych definicji zostały dokonane w pracy zbiorowej N. Guarino, D. Oberle i S. Staab [2009, s. 2]. Już w 1993 r. Gruber [1993a, s. 2] wprowadził definicję ontologii, według której jest to „jawna specyfikacja konceptualizacji”, natomiast w 1997 r. Borst [1997, s. 12] określił ją jako formalną specyfikację

współdzielonej konceptualizacji. Ostatecznie w 1998 r. Studer, Benjamins i Fensel [1998, s. 25] połączyli wymienione definicje, wprowadzając tym samym do literatury najczęściej cytowaną definicję: „Ontologia to formalna, jawna specyfikacja współdzielonej konceptualizacji”. Wiele kontekstów i celów wykorzystania ontologii potwierdza również słuszność proponowanych przez Grubera [1993b, s. 2] kryteriów oceny tych systemów ontologii, takich jak: jasność, spójność, przejrzystość, rozszerzalność, minimalne zaangażowanie formalizacyjne, minimalne zaangażowanie ontologiczne.

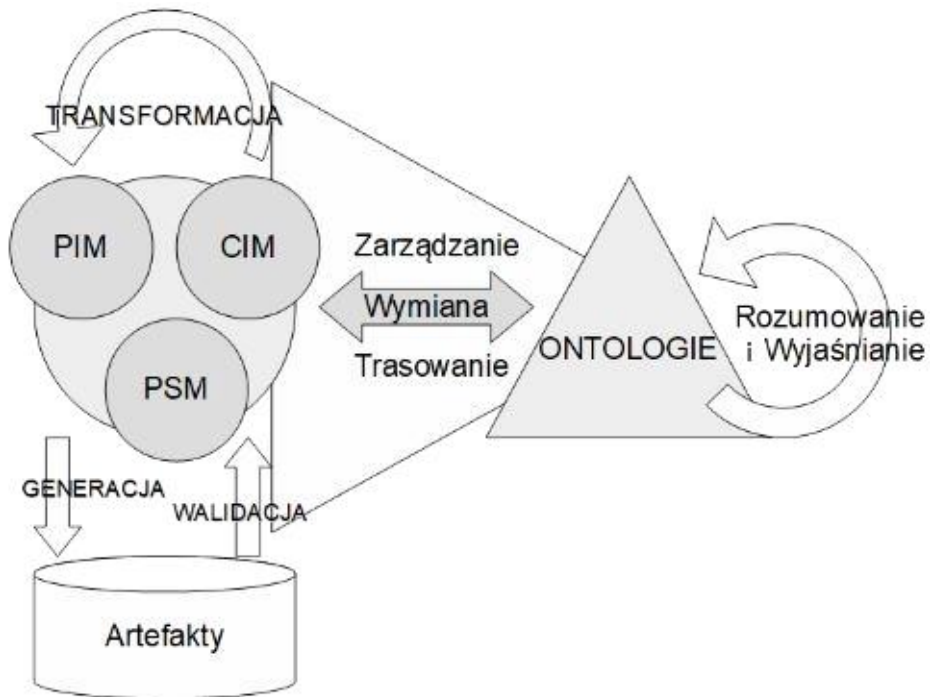
Autor niniejszego artykułu prezentuje interesujące go założenia dotyczące wykorzystania języka *OntoUML* w procesie *UPON*, w szczególności wykorzystania *OntoUML* i *UPON* podczas wytwarzania ontologii dziedzinowej. Identyfikuje się tu cechy wytwarzania ontologii dziedzinowych w środowisku ich wykorzystania *ODSD*. W związku z tym w artykule zawarto również wyjaśnienia na temat znaczenia poszczególnych modeli, z którymi te ontologie dziedzinowe kooperują.

1. Przegląd literatury

Pojęcie ontologii oraz pokrewnych jej zagadnień, takich jak np. wytwarzanie ontologii, w literaturze występuje od dawna i jest związane z wieloma dziedzinami, w których są one wykorzystywane. Z tego powodu przegląd literatury został zawężony do obszaru produkcji oprogramowania, a także wybranego dorobku naukowego z ostatnich kilkunastu lat, w szczególności w odniesieniu do *Ontology-Driven Software Development (ODSD)*. Z jednej strony zapewnia to spójność pomiędzy rozpatrywanymi pojęciami dotyczącymi ontologii oraz procesu wytwarzania i języka, natomiast z drugiej strony pozwala wykazywać analogię do tych pojęć, a nawet ich związek z powszechniejszymi rozwiązaniami mieszczącymi się w nomenklaturze modelowania oprogramowania i tzw. *Model Driving Engineering (MDE)*.

Rysunek 1 stanowi punkt odniesienia do dalszych rozważań, pozwala zauważyć związek trasowania pomiędzy modelami architektury *MDA (Model Driven Architecture)* [www 1] a ontologiami. Spośród wyszczególnionych to właśnie model *CIM (Computation Independent Model)* jest najbardziej interesujący pod względem celu artykułu. Odnosi się on do konceptualizacji wybranego wycinka rzeczywistości/wiedzy. Definicja modelu *CIM* pozwala wnioskować o potencjalnym związku modelu z ontologią dziedzinową i jej odpowiedzialnością w ramach procesu wytwarzania oprogramowania: „*CIM* to model niezależny od przetwarzań informatycznych/obliczeniowych, który utożsamia się rów-

niez z modelem biznesowym lub dziedziny, gdzie zawarte jest słownictwo ekspertów dziedzinowych. Z modelu wynika, do czego system będzie używany, natomiast ukrywa wszystkie informacje technologiczne związane z jego dalszą implementacją” (tłum. autora) [Truyen, 2006, s. 5].



Rys. 1. Ontology-Driven Software Development (ODSD)

Źródło: Na podstawie: [Pan i in. (eds.), 2013].

Warto zwrócić uwagę na postrzeganie ontologii (usług i technologii ontologicznych) w środowisku wytwarzania oprogramowania [Pan i in. (eds.), 2013]. Mówi się w nim o następujących kwestiach, które rzeczywiście mogą implikować skutki nawet w tak zawężonym obszarze, jakim jest wytwarzanie ontologii dziedzinowych, tj. o: powiązaniu technologii wytwarzania oprogramowania i technologii wytwarzania ontologii, językach ontologicznych i tzw. logice opisowej (*description logics*), podstawowych technologiach dla ODSD (usługi skalowania, identyfikowalność, metamodelowanie), sprawdzeniu spójności w ODSD.

1.1. Podstawowe technologie ODSD

W literaturze przedmiotu pisze się o następujących technologiach podstawowych ODSD.

Usługi skalowania (w rozumieniu ontologicznym)

Usługa skalowania ontologicznego związana jest zasadniczo z tzw. TrOWL (Tractable Reasoning Infrastructure for OWL2 – Web Ontology Language ver. 2). TrOWL wspiera kompletnie wysoką ekspresywność OWL2, w szczególności tam, gdzie jest używane tzw. semantyczne przybliżenie podczas transformacji OWL2 DL (rozszerzenie OWL Lite) do OWL2 QL (Query Languages). Specyfikacja OWL2 DL wraz z wyszczególnieniem różnic względem OWL2 Full oraz OWL2 QL została zamieszczona na stronie organizacji W3 [www 2].

Identyfikowalność/trasowalność (*traceability*), tj. zdolność do weryfikacji np. lokalizacji, użycia, historii

Trasowalność rozumiana jest w obszarze ODSD jako możliwość wsparcia udziałowców projektu w zrozumieniu powiązań i zależności, które istnieją pomiędzy obiektami/artefaktami wytworzonymi podczas procesu produkcji oprogramowania. Schwarz i Ebert [2013, s. 121] wymieniają następujące obszary zastosowania trasowalności: zarządzanie zmianą, utrzymanie, zarządzanie projektem, zapewnienie jakości, zarządzanie wymaganiami, ponowne użycie, inżynieria wsteczna.

Metamodelowanie

Konstruowanie zbioru koncepcji w zakresie dziedziny – odnosi się w tym przypadku do języka OWL. Interesujący jest fakt, że dominujące organizacje, takie jak: OMG, NeOn Technologies Foundation czy W3C OWL Working Group, specyfikują w tym zakresie własne metamodely.

Metamodel OWL grupy OMG (Object Management Group) stanowi część innego metamodelu OMG – ODM (Ontology Definition Metamodel), który wyszczególnia związek uogólnienia klasy języka OWL dla klasy języka RDF. Neon OWL Metamodel koresponduje natomiast ze składnią funkcjonalną języka OWL DL. Ponadto Neon OWL Metamodel bazuje na standardzie MOF (Meta Object Facility Specification), wytworzonym przez grupę OMG [www 3]. W3C OWL Working Group z kolei wydaje się najaktualniejszy, jako że specyfikuje metamodel oparty dla kolejnej wersji języka, tj. OWL 2.

1.2. Sprawdzanie spójności w ODSD

Sprawdzanie spójności w ODSD postrzegane jest w następujących perspektywach.

Ontologie i technologie powiązane/pomostowe

Przestrzeń technologiczna ontologii (Ontology Ware Technology Space) dostarcza wielu usług. Usługi rozumowania pozwalają sprawdzać zawartość ontologii w celu wnioskowania o ukrytej wiedzy. Natomiast usługi zapytań pozwalają wykorzystywać język zapytań SPARQL w ontologiach.

Wykorzystanie ontologii przy modelach struktur

Wykorzystanie modeli struktur znajduje zastosowanie w całym procesie produkcji oprogramowania. W literaturze przedmiotu można spotkać się z podziałem na modele deskrypcyjne (*descriptive*), służące do opisu wymagań systemowych, oraz modele nakazowe (*prescriptive*) wykorzystywane do projektowania i implementacji systemów informatycznych.

Podejście ontologiczne dla modeli procesów

W praktyce gospodarczej upowszechniła się notacja BPMN (Business Process Modeling Notation) w celu modelowania i reprezentowania procesów biznesowych. Pomimo wielu możliwości związanych z odwzorowaniem rzeczywistych procesów gospodarczych (np. podprocesy, pętle, rozgałęzienia) dostrzega się potrzebę ulepszenia semantycznego znaczenia tych modeli.

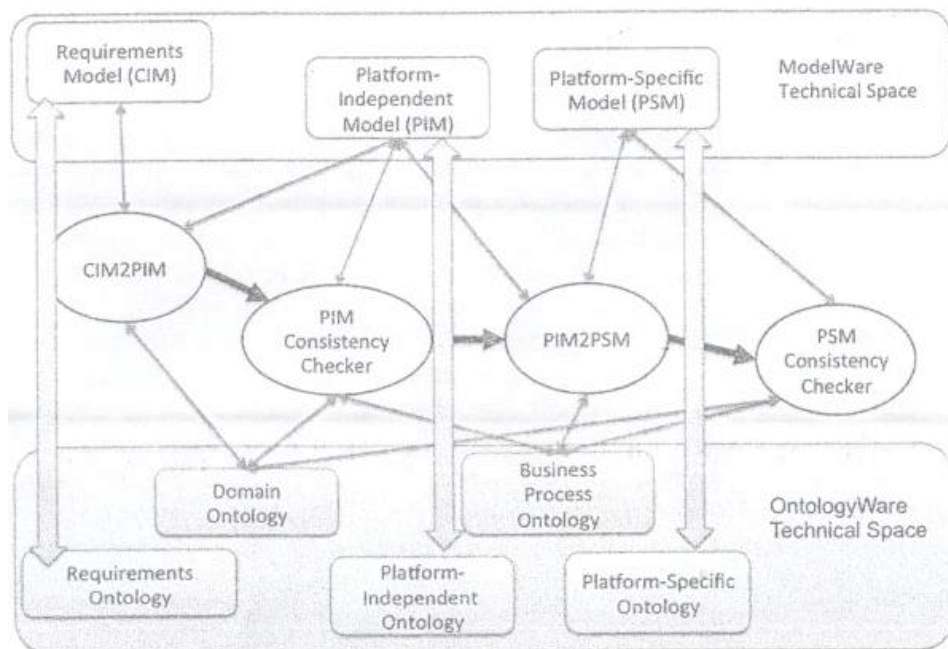
1.3. Metamodel dla modeli ontologicznie zintegrowanych

Złożoność zagadnień ontologicznych oraz modelowania systemów w związku z integracją ODSD sprowadza tę problematykę do formułowania samego metamodelu dla modeli zintegrowanych ontologicznie (Ontology-Driven Meta-Modeling – ODMM). W literaturze przedmiotu można znaleźć określenie mostu integracyjnego (*integration bridge*), za pomocą którego dokonuje się integracji ontologii z modelowanym systemem.

Reasumując, dalszy przegląd literatury zostanie zawężony do tematyki wytwarzania ontologii dziedzinowych, których odpowiedzialność pokrywa się z modelem CIM. ODSD, jako że koncentruje się na całym środowisku i integracji ontologii z modelowaniem, nie wskazuje jednoznacznie sposobu wytwarzania ontologii dziedzinowych. Niemniej jednak, zważywszy na rys. 2 wywodzący się

z nomenklatury ODSO [Pan i in. (eds.), 2013], ontologie dziedzinowe są „pokrewne” zarówno z modelami CIM, jak i ontologiami wymagań. Na uwagę zasługuje cytat U. Assmann (tłum. autora) [Pan i in. (eds.), 2013, s. 10]: „Dla inżynierii wymagań sugerujemy użycie ontologii niezależnej od domeny w celu określenia ogólnych pojęć, wymagań i zależności, które mają być wykrywane podczas ujawniania wymagań”. Rysunek 2 wizualizuje lokalizację m.in. ontologii dziedzinowych w środowisku ODSO, gdzie tzw. Domain Ontology kooperuje z obszarami pracy:

- CIM2PIM (Computation Independent Model to Platform Independent Model),
- PIM (Platform Independent Model),
- PSM (Platform Specific Model).



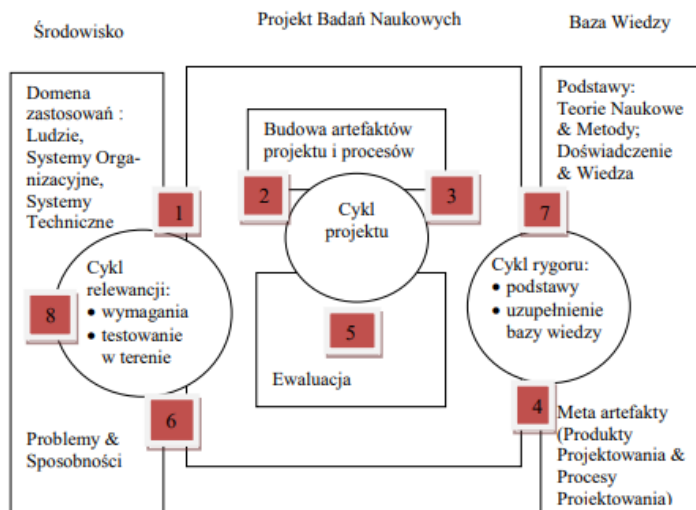
Rys. 2. Komunikacja modeli i ontologii w środowisku ODSO

Źródło: [Pan i in. (eds.), 2013, s. 8].

W dalszej części artykułu, po wyjaśnieniu metody badań, w punkcie 3 zostaną omówione: proces wytwarzania ontologii (UPON) dziedzinowych wraz z adekwatnym językiem wytwarzania ontologii (OntoUML), których użycie będzie lokowane w powyżej scharakteryzowanym środowisku ODSO.

2. Metoda badań

Niniejszy artykuł został napisany z wykorzystaniem paradygmatu badań DSR (Design Science Research). Wobec zawartych w metodzie kooperujących ze sobą cykli relewancji, projektu i rygoru (poniżej wyjaśnionych), przyjęte zostały przyjęte wytyczne, które uwzględniono w celu tego opracowania. Przed zastosowaniem DSR zarówno proces UPON, jak i język OntoUML, stanowią odrębne artefakty. Kluczowa jest tu wiedza poświęcona odpowiedzialności kolejnych cykli DSR oraz sekwencji kroków badacza, pozwalających na realizację badań.



Rys. 3. Cykle w paradygmacie DSR

Źródło: [Pańkowska, b.r.].

Cykl relewancji odpowiada za powiązanie środowiska kontekstowego projektu badań z działaniami naukowymi. „Prowadzący badania w procesie cyklicznym zadaje pytanie, czy prowadzone badanie odzwierciedla rzeczywistość i czy generuje artefakty na podstawie obserwacji i dla tej rzeczywistości” [Pańkowska, b.r.]. W **cyklu projektu** tworzy się oraz weryfikuje się artefakty w procesie badawczym – praca jest organizowana iteracyjnie i artefakty te tworzone są w drodze kolejnych inkrementacji. Natomiast odpowiedzialność **cyklu rygoru** sprowadza się do odpowiedzi na zasadnicze pytanie: czy prowadzone badanie wnosi wkład do nauki w postaci nowej wiedzy i doświadczeń? Jest to osiągalne właśnie poprzez połączenie projektowania i badania naukowego z bazą wiedzy.

Wykorzystując scharakteryzowane powyżej cykle DSR, można wyjaśnić następujące kwestie:

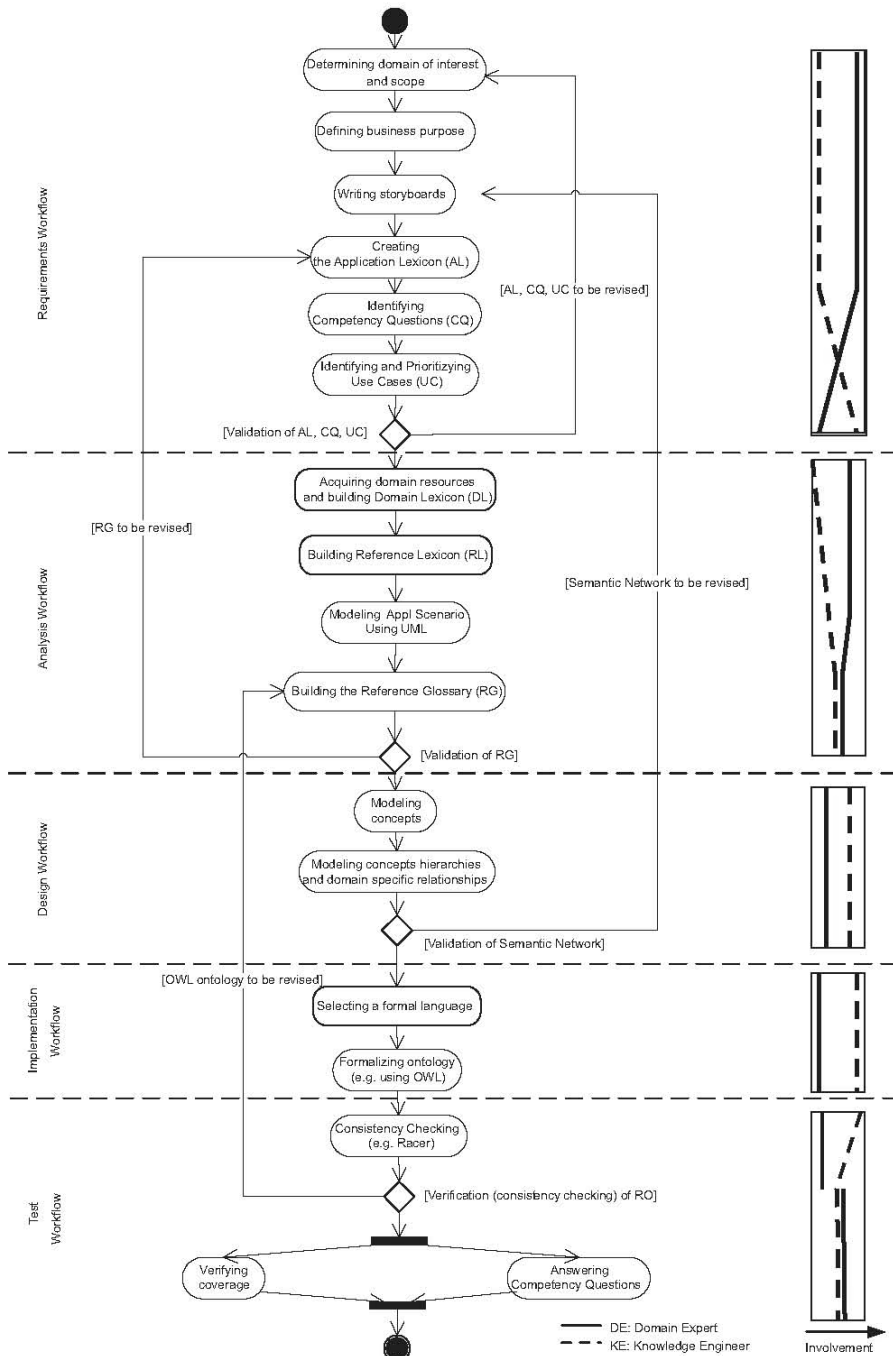
- odzwierciedlenie rzeczywistego środowiska użycia procesu UPON i języka *OntoUML* (cykl relewancji),
- zastosowanie języka *OntoUML* w procesie UPON, w którym rozwiązuje się ewentualne problemy (cykl projektu),
- połączenie wyników projektu z bazą wiedzy i odpowiedź na pytanie: czy dokonano wkładu do nauki? Czy zastosowanie *OntoUML* w procesie UPON jest nowatorskie?

Cel artykułu pokrywa się zasadniczo z uzgodnieniem i uspoźnieniem literatury dotyczącej zarówno języka *OntoUML*, jak i procesu UPON. Wobec tego metoda analizy i krytyki piśmiennictwa wydaje się najwłaściwsza w pracy, tym bardziej, że „metodę tę praktycznie w pierwszym etapie poznania naukowego musi zastosować każdy, kto chce napisać pracę naukową” [Apanowicz, 2002, s. 72].

3. Rezultaty badań

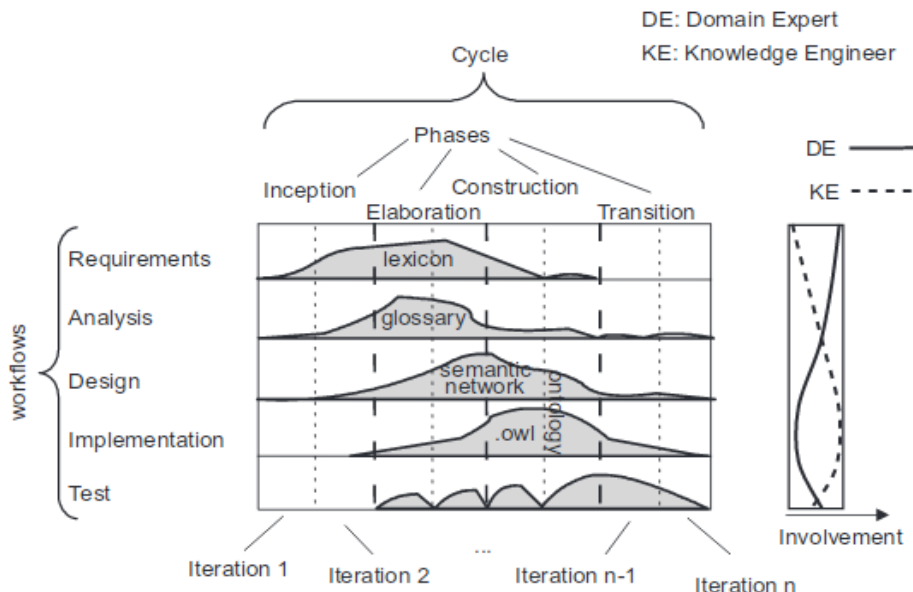
Dokonany przegląd literatury pozwolił przede wszystkim na ugruntowanie środowiska i kontekstu dla opisywanego dalej procesu i języka wytwarzania ontologii (cykl relewancji w DSR). Wydaje się to szczególnie interesujące wobec opisywanych rezultatów badań, bowiem istnieje wiele perspektyw postrzegania ontologii w ogóle (jako jeden z działów filozofii, który przyczynia się do rozumienia istnienia) [Mizoguchi, Ikeda, Sinitsa, 1997], a nawet ich zastosowania w informatyce.

Analiza wybranych metod budowy ontologii [Sobczak, 2006] przyczyniła się do wybrania procesu UPON ze względu na podejście wywodzące się z produkcji oprogramowania, w tym metodę przypadków użycia, co koresponduje ze środowiskiem wytwarzania oprogramowania ODS. Na rys. 4 i 5 przedstawiono framework iteracyjno-przyrostowy procesu UPON, gdzie można wstępnie rozpoznać obszar wykorzystania języka *OntoUML* podczas tzw. design workflows.



Rys. 4. Przepływ sterowania pomiędzy aktywnościami procesu UPON

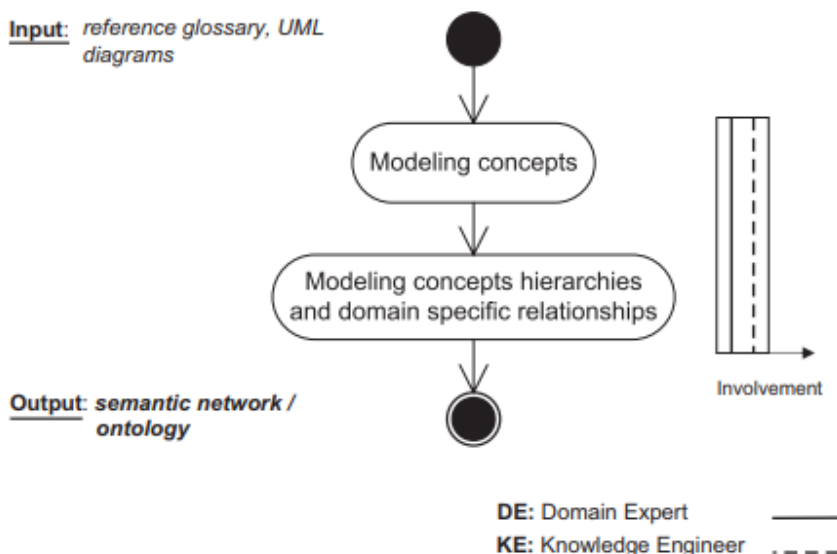
Źródło: [Nicola, Missikoff, Navigli, 2009, s. 273].



Rys. 5. Zunifikowany proces wytwarzania ontologii (UPON)

Źródło: [Nicola, Missikoff, Navigli, 2009, s. 3].

Innymi słowy, decyzje projektowe podjęte w obrębie projektowania (*design workflows*), zapewniają kompatybilność z językiem implementacji ontologii np. popularnym OWL. Kolejny diagram czynności (Unified Modeling Language) zaprezentowany na rys. 6 precyzuje wykorzystanie języka OntoUML do czynności odpowiadających za: modelowanie strukturalne koncepcji (*modeling concepts*), hierarchie koncepcji i relacje specyficzne dla domeny (*modeling concepts hierarchies and domain specific relationships*). Ten obszar pracy wnosi zmiany w stosunku do źródłowej publikacji poświęconej UPON, gdzie wybrano podejście OPAL (Object, Process, Actor modelling Language). Uzasadnienie nawiązuje do pragmatyki związanej z obszarem projektowania i następnego obszaru pracy, tj. implementacji ontologii. Język OntoUML, jak wynika z dokonanej analizy rynku oprogramowania, posiada wsparcie wśród narzędzi wspierających proces produkcji oprogramowania CASE (np. Enterprise Architect). Oznacza to co najmniej tyle, że można sporządzić projekt ontologii bezpośrednio w repozytorium projektu oprogramowania, wykorzystując np. jeden z mechanizmów ontologicznych traceability czy też inżynierię do przodu, w celu wygenerowania kodu języka OWL.

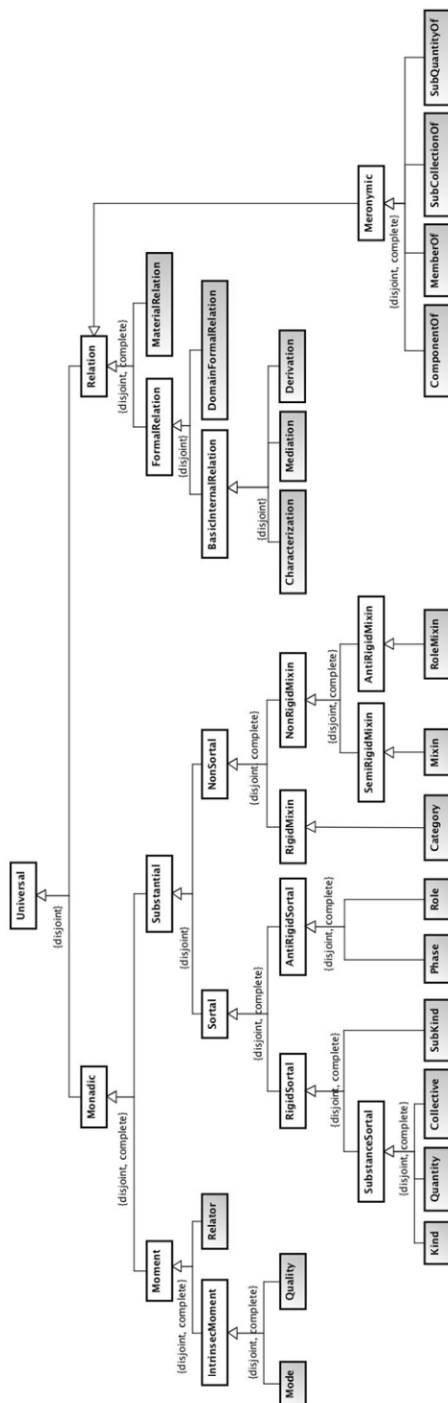


Rys. 6. Przegląd przepływu pracy w obrębie projektowania w procesie UPON

Źródło: [Nicola, Missikoff, Navigli, 2009, s. 10].

Modelowanie strukturalne koncepcji

Modelowanie strukturalne koncepcji z wykorzystaniem języka OntoUML wymaga wyjaśnienia znaczenia stereotypów. Język OntoUML wyszczególnia stereotypy (tabela 1) związane z problematyką konceptualizacji rzeczywistości. Rysunek 7 podsumowuje główne stereotypy języka specyfikacji ontoUML, wraz z krótkim wyjaśnieniem ich znaczenia. Specyfikacja pozostałych stereotypów, znajduje się na stronie [www 4], natomiast gruntowny opis – w publikacji źródłowej [Guizzardi, 2005].



Rys. 7. Hierarchia stereotypów języka specyfikacji OntoUML

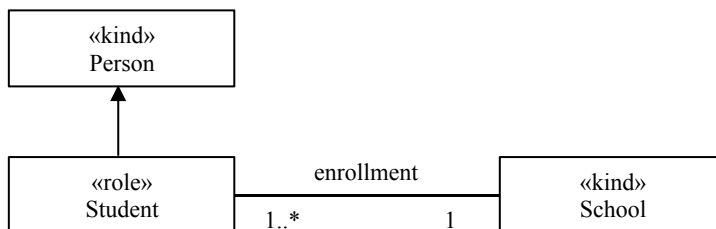
Źródło: [www 5].

Tabela 1. Opis stereotypów OntoUML

Nazwa stereotypu	Grupa	Opis
«kind»	Pojęcia fizyczne > Pojęcia niezmiennie	Stereotyp klasy UML reprezentujący pojęcie fizyczne, którego instancje są funkcjonalnie kompletne (np. człowiek, krzesło). Pojęcia typu muzyka, koncert nie są zaliczane do «kind»
«subkind»		Podrodzaj (podklasa), relacyjnie niezależne ograniczenie pojęcia «kind» (np. mężczyzna dla rodzaju człowiek)
«collective»		Pozwala zidentyfikować kolekcje klas (np. klub piłkarski stanowi kolekcję piłkarzy)
«quantity»		Wyróżnia klasy, której instancje są masowe/liczebne/iłościami (np. piasek, woda)
«phase»	Pojęcia fizyczne > Pojęcia zmienne	Zaliczana do ontologicznych właściwości niestałych/zmiennych (<i>anti-rigid properties</i>). Umożliwia zidentyfikowanie obiektu w czasie egzystencji klasy/rodzaju (np. rodzaj człowiek pozwala rozróżnić fazy dziecka, nastolatka, dorosłego)
«role»		Zaliczana do ontologicznych właściwości niestałych/zmiennych (<i>anti-rigid properties</i>). Pozwala określić rozróżnialny obiekt tego samego rodzaju (np. student i człowiek)
«category»	Pojęcia niefizyczne > Mieszanka pojęć niezmiennych	Stereotyp pozwala na uogólnienie i reprezentację mieszanki pojęć/ klas (niezmiennych). Klasa kategorii zawsze jest definiowana jako abstrakcyjna
«mixin»	Pojęcia niefizyczne > Mieszanka pojęć zmiennych	Stereotyp MixIn umożliwia uogólnienie oraz reprezentację mieszanki zarówno pojęć/klas niezmiennych, jak i pojęć abstrakcyjnych czy też innych właściwości, m.in. typu «phase» (np. miejsce siedzenia uogólnia krzeselko i solidne miejsce do siedzenia). Klasa MixIn zawsze jest definiowana jako abstrakcyjna
«rolemixin»		Stereotyp RoleMixIn pozwala reprezentować takie mieszanki pojęć/ klas, które nie są niezmiennie (<i>anti-rigid</i>) – np. klient uogólnia prywatnego i biznesowego klienta. Klasa RoleMixIn zawsze jest definiowana jako abstrakcyjna

Hierarchie koncepcji i relacje specyficzne dla domeny

Zdefiniowane koncepcje na etapie *modeling concepts*, w niniejszej czynności powinny zostać powiązane z wykorzystaniem specyfiki znanej z diagramu klas (UML). Używa się tu związków np. generalizacji, agregacji, asocjacji. Taka hierarchia klas ontologicznych pozwala uzyskać implementację języka OWL. Na rys. 8 zaprezentowano przykład z pracy źródłowej [Guizzardi, 2005] dla OntoUML, tj. *Ontological Foundations for structural conceptual models*.



Rys. 8. Przykład strukturyzowania klas OntoUML

Źródło: [Guizzardi, 2005, s. 129]

Z kolei w publikacji [Barcelos i in., 2013] można zapoznać się z praktykami towarzyszącymi transformacjom modeli OntoUML do reprezentacji OWL (tutaj pomijane ze względu na zakres artykułu). Doświadczenia tego rodzaju potwierdzają również zasadność zastosowania ontoUML opisanego we wcześniej wymienionej pracy [Guizzardi, 2005].

Podsumowanie

W niniejszym artykule, poświęconym tematowi adaptacji języka ontoUML w procesie UPON, starano się uwzględnić definicję „Ontologia to formalna, jawna specyfikacja współdzielonej konceptualizacji” [Studer, Benjamins, Fensel, 1998, s. 25]. Przy czym jej interpretacja autorstwa J. Gołuchowskiego pozwala dalej sprowadzić tę dyskusję zarówno do samego celu artykułu (adaptacji ontoUML w procesie UPON), jak i szerszych, istotnych tu kwestii procesu produkcji oprogramowania oraz wytwarzania ontologii dziedzinowych: „W stwierdzeniu tym konceptualizacja odnosi się do abstrakcyjnego modelu pewnego zjawiska lub bytu, który identyfikuje odpowiednie koncepcje rzeczywistego obiektu. W definicji tej ontologia tworzona jest w celu zapewnienia danym, reprezentującym informacje i wiedzę semantyki zrozumiałej i możliwej do przetwarzania przez komunikujących się agentów. Stanowi ona dla nich wspólną konceptualizację rzeczywistości” [Filipczyk, Gołuchowski, 2006, s. 3].

W dyskusji tej ważne wydają się różne kwestie zawarte w tym artykule, m.in. problematyka:

- organizacji badań z wykorzystaniem paradygmatu DSR i Hevner,
- samego środowiska wytwarzania oprogramowania ODSD i „pokrewnego” MDE,
- wytwarzania i analogii ontologii dziedzinowych do modelu CIM,
- zunifikowanego procesu UPON oraz aktualizacji wykorzystywanego OPAL do OntoUML.

Wobec tych zagadnień warto poddać pod dyskusję sens podjętej pracy. O ile na tle aktualnej literatury przedmiotu można stwierdzić o słuszności opisywanych tu pojęć, to samo wartościowanie tego wysiłku raczej jest subiektywne i osadzone w kulturze organizacji użytkującej, co zresztą znajduje oparcie w problematyce filozofii zarządzania i kolejnego ważnego pojęcia, jakim jest aksjologia. Należy pamiętać, że nauki o zarządzaniu (dla których wytwarzanie ontologii dziedzinowych ma szczególne znaczenie) nie są wolne od wartościowania, tzn. zarówno tworzenie teorii, jak i realizacja działań organizacyjnych wiąże się z wyznawanymi systemami wartości, ukształtowanymi kulturowo [Sułkowski, 2012].

Reasumując, cel artykułu, jakim było zaprezentowanie założeń dla wykorzystania języka OntoUML w procesie wytwarzania ontologii UPON, został osiągnięty. Wykazano obszar wykorzystania języka OntoUML w tym procesie UPON, tj. w przepływie projektowania, charakteryzując przy tym wybrane stereotypy na podstawie załączonego metamodelu OntoUML. Niemniej istotna była kwestia samego zastosowania wytwarzanych ontologii dziedzinowych. W ramach środowiska ODSD ontologie te znajdują swoje zastosowanie przy formułowaniu modeli niezależnych obliczeniowo, tj. CIM. Z założenia modele te, za sprawą ontologii dziedzinowych, mogłyby zapewniać podstawy dla realizowanych projektów informatycznych, wynikające np. z analizy biznesowej przedsiębiorstw.

Literatura

- Apanowicz J. (2002), *Metodologia ogólna*, Wydawnictwo Bernardinum, Gdynia.
- Barcelos P.P.F., Santos V.A. dos, Silva F.B., Monteiro M.E., Garcia A.S. (2013), *An Automated Transformation from OntoUML to OWL and SWRL* [w:] M. Peixoto Bax, M. Barcellos Almeida, R. Wassermann (eds.), ONTOBRAS, Workshop Proceedings, Vol. 1041, CEUR-WS.org, s. 130-141.
- Borst W. (1997), *Construction of Engineering Ontologies*, PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede.

- Filipczyk B., Gołuchowski J. (2006), *Perspektywy wykorzystania ontologii w procesie przetwarzania języka naturalnego w systemach zarządzania wiedzą*, Wydawnictwo Uniwersytetu Ekonomicznego, Katowice.
- Gruber T.R. (1993a), *A Translation Approach to Portable Ontologies*, "Knowledge Acquisition", Vol. 5(2), <http://tomgruber.org/writing/ontolingua-kaj-1993.pdf> (dostęp: 17.05.2019).
- Gruber T.R. (1993b), *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, Vol. 5(2), <http://tomgruber.org/writing/onto-design.pdf> (dostęp: 17.05.2019).
- Guarino N., Oberle D., Staab S. (2009), *What is an Ontology?* [w:] S. Staab, R. Studer (eds.), *Handbook on Ontologies*, International Handbooks on Information Systems, Springer, Berlin, Heidelberg, s. 1-17.
- Guizzardi G. (2005), *Ontological Foundations for Structural Conceptual Model*, Telematics Instituut, Netherlands, https://www.researchgate.net/publication/215697579_Ontological_Foundations_for_Structural_Conceptual_Models (dostęp: 17.05.2019).
- Mizoguchi R., Ikeda M., Sinita K. (1997), *Roles of Shared Ontology in AI-ED Research – Intelligence, Conceptualization, Standardization, and Reusability*, Proceedings of AIED-97, s. 537-544.
- Nicola A., Missikoff M., Navigli R. (2009), *Software Engineering Approach to Ontology Building*, "Information Systems", Vol. 34, s. 258-275.
- Nicola A. de, Missikoff M., Navigli R. (2005), *A Proposal for a Unified Process for Ontology Building: UPON*, Dexa, Copenhagen.
- Pan J.Z., Staab S., Assmann U., Ebert J., Zhao Y. (2013), *Ontology-Driven Software Development*, Springer, Berlin.
- Pańkowska M. (b.r.), *Seminarium dla doktorantów. Paradygmat badań naukowych Hevnera i in.*, <http://docplayer.pl/1535388-Seminarium-dla-doktorantow-paradygmat-badan-naukowych-hevnera-i-in.html> (dostęp: 30.08.2017).
- Schwarz H., Ebert J. (2013), *Traceability* [w:] J.Z. Pan i in. (eds.), *Ontology-Driven Software Development*, Springer, Berlin, s. 121-150.
- Sobczak A. (2006), *Analiza wybranych metod budowy ontologii* [w:] T. Porębska-Miąc, H. Sroka, *Systemy wspomagania organizacji. SWO 2006*, Wydawnictwo Akademii Ekonomicznej, Katowice, s. 183-191.
- Stal J. (2009), *Charakterystyka narzędzi wspierających modelowanie ontologii*, „Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie”, nr 798, s. 233-244.
- Sułkowski Ł. (2012), *Epistemologia i metodologia zarządzania*, Polskie Wydawnictwo Ekonomiczne, Warszawa.
- Studer R., Benjamins R., Fensel D. (1998), *Knowledge Engineering: Principles and Methods*, "Data & Knowledge Engineering", Vol. 25(1-2), s. 161-198.
- Truyen F. (2006), *The Fast Guide to Model Driven Architecture (MDA). The Basics of Model Driven Architecture*, Cephias Consulting Corp., USA.

[www 1] <http://www.omg.org/mda/> (dostęp: 2.05.2018).

[www 2] https://www.w3.org/TR/owl2-primer/#OWL_2_DL_and_OWL_2_Full (dostęp: 11.04.2018).

[www 3] <http://www.omg.org/mof/> (dostęp: 2.08.2018).

[www 4] <http://ontology.com.br/ontouml/spec/#Mixin> (dostęp: 2.10.2018).

[www 5] <http://ontology.com.br/ontouml/spec/#Mixin> (dostęp: 2.10.2018).

ASSUMPTIONS FOR PROCEDURAL USING THE OntoUML LANGUAGE

Summary: The article proposes using the ontoUML in the UPON (Unified Process Ontology Development) ontology process. As a result, the production of domain ontologies should be more predictable, e.g., in the development of ODS (Ontology Driven Software Development). Thus, it meets possible quality problems of domain ontologies (quality is usually due to the ordering of project tasks). The article tries to fill the gap in the literature on the subject. At the same time, it is easier to provide high-quality domain ontologies to the ODS software development process. The originality of the article results from the coherence of the OntoUML language with the UPON process.

Keywords: OntoUML, UPON, ODS.