

Andrzej Sobczak

ANALIZA TECHNIK REPREZENTACJI WYMAGAŃ STOSOWANYCH W PROCESIE BUDOWY SYSTEMÓW INFORMATYCZNYCH

1. Wprowadzenie

Definiowanie wymagań jest procesem odkrywania potrzeb i oczekiwań interesariuszy¹ oraz dokumentowania ich w postaci nadającej się do analizy, komunikacji i późniejszego wdrożenia. Bardziej formalna definicja wymagania zawarta jest w dokumencie „IEEE Guide for Developing System Requirements Specifications”, w którym wymaganie określone jest jako [*IEEE Guide...* 1996, s. 3]:

1) warunek lub zdolność systemu niezbędna użytkownikowi lub klientowi do rozwiązania problemu lub osiągnięcia zamierzonego celu;

2) warunek, który musi być spełniony, lub zdolność, którą musi posiadać system lub część systemu, aby spełnić umowę, normę, specyfikację lub inny formalnie przyjęty dokument;

3) udokumentowane wyrażenie zdolności lub warunków systemu określonych w punktach 1 i 2.

Wymagania dzielą się na dwie odmienne kategorie:

- wymagania funkcjonalne, które opisują, jakie funkcje musi realizować system lub jakie powinien świadczyć usługi, bez szczegółowego opisu, jak będzie to realizował;
- wymagania niefunkcjonalne, które można zdefiniować jako ograniczenia odnoszące się do usług lub czynności wykonywanych przez system.

¹ Interesariusze (ang. *stakeholders*) – są to osoby lub organizacje, które w sposób bezpośredni lub pośredni mają kontakt z budowanym lub modyfikowanym systemem informatycznym zarządzania. Terminu „interesariusz” jako odpowiednika angielskiego *stakeholder* używa A. Ehrlich w przykładzie pracy Stoner J.A.F., Freeman R.E., Gilbert jr. D.R., *Kierowanie*, Polskie Wydawnictwo Ekonomiczne, Warszawa 1998.

Istnieją dwie podkategorie wymagań niefunkcjonalnych:

- wymagania jakościowe, które określają szczegółowe oczekiwania co do wydajności, jakie interesariusze mają w odniesieniu do wymagań funkcjonalnych; zapewniają one kontekst do oceny, jak dobrze wymagania funkcjonalne zaspokajają potrzeby interesariuszy;
- ograniczenia, tj. zasady, zależności i konwencje określające granice, w ramach których muszą być realizowane wszystkie inne wymagania.

Systematyczne podejście do formułowania wymagań przez iteracyjny proces analizowania problemu, dokumentowania uzyskanych w ten sposób spostrzeżeń dotyczących wymagań i sprawdzania dokładności uzyskanego efektu końcowego z oczekiwaniami odbiorcy określane jest mianem inżynierii wymagań, będącej częścią inżynierii oprogramowania. Stanowi ona pomost między nieformalnym światem potrzeb interesariuszy i formalnym światem zachowań systemów informatycznych.

Jednoznaczne określenie wymagań jest koniecznym warunkiem wstępnym nie tylko walidacji wymagań², lecz także rozwiązania potencjalnych konfliktów między interesariuszami. Jednocześnie sformułowanie poprawnych wymagań względem systemu informatycznego zarządzania wymaga wiedzy, współpracy i porozumienia pomiędzy kadrą zarządzającą i informatykami. W wielu organizacjach obserwuje się brak takiego porozumienia – głównie dlatego, że obie grupy funkcjonują w różny sposób i koncentrują swoją uwagę na zupełnie odmiennych funkcjach i działaniach [Al-Karaghoul, Al-Shawi, Fitzgerald 2000, s. 93-94]. Sytuację tę pogarsza niemożność opisanego przez menedżerów swojej działalności na takim poziomie abstrakcji, który umożliwiłby jej przełożenie na język techniki informatycznej. Jednocześnie informatycy nie są traktowani jako równorzędni „partnerzy do rozmów” przez kadrę menedżerską.

2. Klasyfikacja dotychczas stosowanych technik reprezentacji wymagań

Dotychczasowe badania oraz praktyczne aplikacje z zakresu reprezentacji wymagań koncentrują się na dwóch obszarach:

- wykorzystania języka naturalnego do dokumentowania wymagań [Denger, Dorr, Kamsties 2000; Lee, Xue, Kuo 2001];
- użycia semiformalnych i formalnych struktur do reprezentacji wymagań – w tym również notacji graficznych [Easterbrook 1991; Greenspan, Mylopoulos, Borgia 1994; Kowalczykiewicz, Nawrocki 2001; Lamsweerde 2001; Lee, Xue, Kuo 2001; Osterwalder, Pigneur 2002; Santander, Castro 2001; Tse, Pong 1991].

² Walidacja wymagań jest procesem ustalania, czy wymagania stanowią dokładne odzwierciedlenie potrzeb interesariuszy.

Stosowanie języka naturalnego w procesie dokumentowania wymagań w wielu przypadkach uniemożliwia uzyskanie poprawnej (tj. jednoznacznej i spójnej) ich listy. Występuje tutaj bowiem bariera w postaci niejednoznaczności w rozumieniu tego samego tekstu przez różnych interesariuszy.

Powyższej wady nie mają techniki wykorzystujące semiformalne i formalne struktury do reprezentacji wymagań. Zalicza się do nich m.in.:

- pseudokod [Lee, Xue, Kuo 2001, s. 298-299];
- specjalizowane notacje do dokumentowania wymagań [Greenspan, Mylopoulos, Borgia 1994; Kowalczykiewicz, Nawrocki 2001; Tse, Pong 1991];
- formalizmy matematyczne [Easterbrook 1991; Lamsweerde 2001];
- maszyny o skończonej liczbie stanów [Easterbrook 1991; Leffingwell, Widrig 2000, s. 299-301];
- drzewa decyzyjne i tabele decyzyjne [Easterbrook 1991; Leffingwell, Widrig 2000, s. 301-302];
- graficzne drzewa decyzyjne [Easterbrook 1991; Leffingwell, Widrig 2000, s. 301-302];
- diagramy przypadków użycia [Easterbrook 1991; Regnell, Andersson, Bergstrand 1996; Santander, Castro 2001].

W Polsce prowadzone są prace badawcze dotyczące dokumentowania wymagań z zastosowaniem struktur semiformalnych [Kowalczykiewicz, Nawrocki 2001]. Autorzy koncentrują się w nich na specyfice małych firm informatycznych, proponując wykorzystanie języka XML jako standardu reprezentacji wymagań. Istotnym elementem ich rozwiązania jest „wyróżnienie atrybutów pojedynczego wymagania i podział strukturalny jego części opisowej w celu wydzielenia możliwie małych porcji informacji” [Kowalczykiewicz, Nawrocki 2001, s. 87]. W tym celu proponują oni wykorzystanie tzw. dobrych praktyk (ang. *best practices*) Sommerville’a i Sawyera oraz wskazówek opisu wymagań w standardzie ANSI/IEEE 830.

3. Wykorzystanie ontologii do reprezentacji wymagań

W chwili obecnej podejmowane są próby opracowania nowych technik do reprezentacji wymagań – z wykorzystaniem inżynierii wiedzy³ (m.in.: [Breitman, Sampaio, Leite, 2003; Herlea, Catholijn, Treur, Wijngaards 2002; Herlea, Jonker, Tresur 1999; Jin, Bell, Wilkie 1998; Lin, Ho 2000; Moor, Wiegand 1999; Robinson, Pawlowski, Volkov 1999]). Związane jest to z faktem, iż coraz większą wagę przywiązuje się do podejścia, zgodnie z którym „najistotniejsza jest wiedza ukryta w ludziach i w organizacji; trzeba ją wyartykułować, sformalizować i dopiero na tej podstawie budować” system informatyczny [Cempel, Internet 2003].

³ Celem inżynierii wiedzy jest identyfikacja i zbieranie źle ustrukturalizowanych wiadomości z dziedziny przedmiotowej, a następnie transformacja tej wiedzy do postaci, która byłaby akceptowana przez program komputerowy [Radomiński 2001, s. 183].

Ostatnio podejmuje się coraz więcej prac badawczych w zakresie wykorzystania ontologii do formalnego zapisu wiedzy. Sama koncepcja ontologii sięga IV wieku p.n.e. – kiedy to Platon badał charakter i struktury rzeczywistości. Nowożytny podejście do ontologii zostało zaprezentowane w XVII w. w pracach m.in. Christiana Wolffa⁴. Pojęcie „ontologia” w odniesieniu do obszaru informatyki pojawiło się po raz pierwszy w 1967 r. w pracach S.H. Mealy’ego dotyczących modelowania danych. Ontologia zajmuje się odkrywaniem i opisywaniem „tego, co jest” – w rzeczywistości, w umysłach ludzi i zapisanego w postaci różnych symboli [Łabuzek 2004]. Formalną definicję ontologii w odniesieniu do nauk informatycznych można znaleźć w pracy Maedche [Abramowicz 2000]: jest to „konstruowany, z użyciem jednoznacznego słownictwa, w celu opisanie rzeczywistości, produkt inżynierii wiedzy, który wzbogacono o zbiór wniosków dotyczących zamierzonego znaczenia tego słownictwa”.

Od strony formalnej ontologię można zdefiniować poprzez dwa zbiory: definiujący strukturę ontologii oraz leksykon [Maedche, Staab 2001].

Struktura ontologii O :

$$O = \{C, R, Hc, Rel, A\},$$

gdzie:

- C – stanowi zbiór wszystkich pojęć wykorzystanych w modelu, przy czym pojęciem nazywamy ideę reprezentującą pewną grupę obiektów posiadających wspólną charakterystykę, przez co zaliczanych do tej samej grupy;
- R – jest zbiorem nietaksonomicznych relacji definiowanych jako nazwane połączenie między pojęciami;
- Hc – stanowi zbiór taksonomicznych relacji pomiędzy conceptami, gdzie w zależności od przyjętego systemu może to być jedna z poniższych relacji:
 - podklasy rozłączne,
 - wyczerpujące podklasy rozłączne,
 - podklasy;
- Rel – są to zdefiniowane nietaksonomiczne relacje pomiędzy pojęciami;
- A – jest zbiorem aksjomatów, określa strukturę pojęć, relacje między nimi, jak i teorię dotyczącą definiowanego modelu.

Leksykon L :

$$L = \{Lc, Lr, F, G\},$$

gdzie:

- Lc – definicje leksykonu dla zbioru pojęć;
- Lr – definicje leksykonu dla zbioru relacji;

⁴ Należy wskazać tutaj przede wszystkim pracę *Lexicon Philosophorum* z 1613 roku.

F – referencje dla pojęć;

G – referencje dla relacji.

określa leksykon, czyli sposób, w jaki należy rozumieć pojęcia, w tym i relacje.

W niniejszej pracy przyjęto definicję ontologii zaproponowaną przez Grubera, według której [Abramowicz 2000]: ontologia jest formalnym, jawnym przedstawieniem wspólnej warstwy pojęciowej.

Powyższe rozważania pozwalają stwierdzić, że do podstawowych korzyści związanych z zastosowaniem ontologii zalicza się [Abramowicz 2000]:

- wspólne rozumienie pojęć⁵, a co za tym idzie, stosunkową łatwość łączenia różnych ontologii,
- możliwość wielokrotnego wykorzystania wiedzy z danej dziedziny dzięki jej jednoznaczniemu opisowi,
- możliwość różnorodnych analiz i łatwego zapoznania się z wiedzą z danej dziedziny (w tym również w środowiskach rozproszonych – na przykład w Internecie).

Zastosowanie ontologii w odniesieniu do inżynierii wymagań umożliwi więc będzie:

- uzyskanie listy poprawnie sformułowanych wymagań dla systemów informatycznych zarządzania;
- kumulację wiedzy z zakresu wymagań dla systemów informatycznych zarządzania⁶, co pozwala na wielokrotne użycie raz określonych wymagań, a tym samym wpływa na zmniejszenie kosztów budowy kolejnych systemów informatycznych.

Należy tutaj wskazać na jeszcze jedną właściwość ontologii – tj. możliwość integracji. Wskazywane są przede wszystkim trzy typy integracji ontologii [Abramowicz 2000]:

1) łączenie – polegające na stworzeniu, na podstawie analizy syntaktycznej, nowej ontologii na bazie już istniejących, które chce się zintegrować;

2) mapowanie – określany jest tutaj skończony zbiór reguł, będący łącznikiem pomiędzy obiektami integrowanych ontologii;

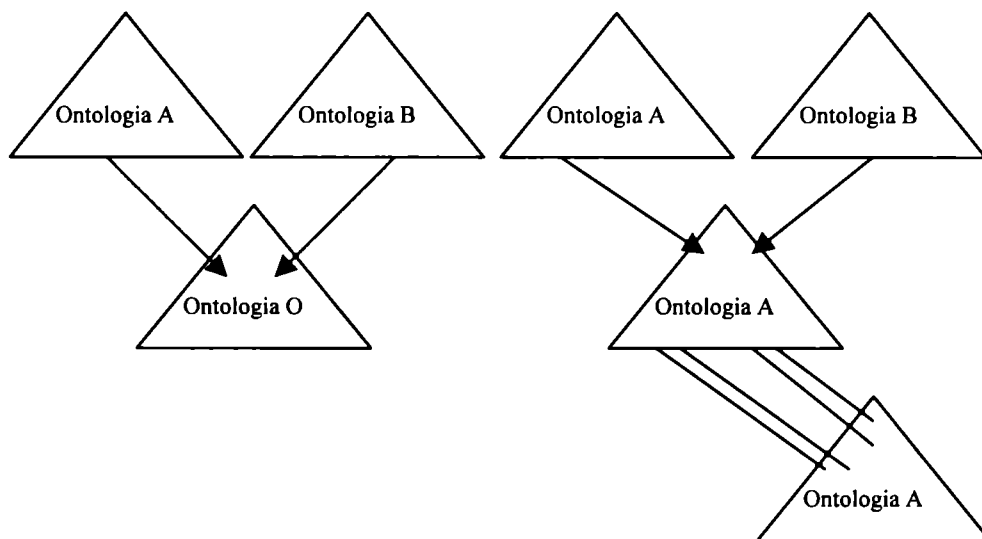
3) translacja – będąca połączeniem dwóch powyższych typów integracji; polega na stworzeniu zbioru reguł mapowania przy wykorzystaniu syntaktyczno-leksykalnych algorytmów porównawczych.

Wprowadza się tutaj pojęcia „zestrojenia” i scalania ontologii. Pierwszy mechanizm polega na wprowadzeniu połączeń (ang. *links*) między poszczególnymi składnikami ontologii, tj. ontologia o szerszym zakresie dziedzinowym

⁵ Dotyczy to zarówno ludzi, jak i agentów programowych (ang. *software agents*).

⁶ Obecnie coraz popularniejsza staje się koncepcja inteligentnej/uczącej się firmy informatycznej (ang. *learning software organisations*), zgodnie z którą istotnym czynnikiem decydującym o powodzeniu realizacji kolejnych projektów informatycznych przez firmę programistyczną staje się możliwość ponownego użycia (ang. *reuse*) raz już pozyskanej wiedzy – w tym dotyczącej inżynierii wymagań [Ruhe, Bomarius 2000].

„przykrywa” ontologię o węższym zakresie dziedzinowym. Drugi mechanizm – scalanie ontologii, tworzy trzecią ontologię na podstawie dwóch wcześniej istniejących. Przedstawia porównanie w formie mechanizmów scalania i zestrzajania ontologii. Na rysunku 1 przedstawiono porównanie scalania i zestrzajania ontologii w formie mechanizmów.



Rys. 1. Porównanie procesów scalania i zestrzajania ontologii

Źródło: opracowanie własne na podstawie [Noy, Musen 1999; 2000].

Dzięki mechanizmom integracji ontologii istnieje możliwość (semi)automatycznego komponowania funkcjonalności systemów z funkcjonalności atomowych; tak powstałe usługi mogą charakteryzować się większym stopniem złożoności i zupełnie nowymi funkcjonalnościami – niedostępnymi dla ich „rodziców”. Podejmowane są prace badawcze, których celem jest dynamiczne komponowanie funkcjonalności systemów informatycznych [Kim, Gil 2004; Medjahed, Bouguet-taya, Elmagarmid, 2003; Pahl, Casey 2003]⁷.

W pracy [Eck, Wieringay 2003] podkreśla się, że przy takim podejściu następuje całkowita zmiana paradygmatu w zakresie procesu zarządzania wymaganiami; do tej pory miał on charakter projektowy – o stosunkowo określonych ramach czasowych, zakończony przygotowaniem specyfikacji wymagań; obecnie zarządzanie wymaganiami będzie miało charakter ciągły – tak jak będą komponowane nowe funkcjonalności.

⁷ Wprowadza się tutaj pojęcie semantycznej usługi sieciowej (ang. *semantic web services*) [Kim, Gil 2004].

4. Zakończenie

Na podstawie przeprowadzonego przeglądu technik reprezentacji wymagań można wskazać, że szczególnie obiecującym podejściem jest zastosowanie w tym obszarze ontologii. Dzięki jej właściwościom – a zwłaszcza możliwości integracji – proces kompozycji procesów biznesowych będzie mógł być realizowany o wiele efektywniej, niż ma to miejsce w chwili obecnej. Pozwoli to na uelastycznienie funkcjonowania organizacji i zapewni możliwość adaptacji do zmieniających się warunków zewnętrznych.

Literatura

- Abramowicz W., *Integracja ontologii*, [w:] *X Forum Teleinformatyki – X Wcieleń Integracji*, Legionowo 2004.
- Al-Karaghoulis W., Al-Shawi S., Fitzgerald G., *Negotiating and Understanding Information Systems Requirements: The Use of Set Diagrams*, „Requirements Engineering”, 2000 vol. 5 nr 2, s. 93-102.
- Breitman K., Sampaio J., Leite P., *Ontology as a Requirements Engineering Product*, 11th IEEE International Requirements Engineering Conference, Monterey Bay, California, USA, September 08-12, 2003.
- Cempel Cz., *Spółeczeństwo wiedzy – nowy wymiar kreowania i użytkowania wiedzy*, Serwis eGov.pl, Internet, kwiecień 2003.
- Denger Ch., Dorr J., Kamsties E., *A Survey on Approaches for Writing Precise Natural Language Requirements*, Fraunhofer Institut Experimentelles Software Engineering, IESE-Report nr 070.01/E, October 2001.
- Easterbrook S., *Handling Conflict between Domain Descriptions with Computer-Supported Negotiation*, „Appears in Knowledge Acquisition: An International Journal” 1991 vol 3, s. 255-289.
- Eck P., Wieringay R., *Requirements Engineering for Service-Oriented Computing: A Position Paper*, Proceedings of the First International E-Services Workshop, ICEC 03, Pittsburgh (USA) 2003.
- Eck P., Wieringa R., *Web Services as Product Experience Augmenters and the Implications for Requirements Engineering: A Position Paper*, International Workshop in Service-oriented Requirements Engineering, September 6, Kyoto (Japan) 2004.
- Greenspan S., Mylopoulos J., Borgida A., *On Formal Requirements Modeling Languages: RML Revisited*, Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on Software Engineering, Italy, May 1994.
- Herlea D., Catholijn J., Treur J., Wijngaards N., *A Compositional Knowledge Level Process Model of Requirements Engineering*, „International Journal of Software Engineering and Knowledge Engineering” 2002 vol. 12 nr 1, s. 41-75.
- Herlea D., Greenberg S., *Using a Groupware Space for Distributed Requirements Engineering*, Computer Science Technical Reports No. 1998-614-05, Department of Computer Science, The University of Calgary, Canada, 1998.
- Herlea D., Jonker, M., Treur W., *Integration of Behavioural Requirements Specification within a Knowledge Engineering Methodology*, Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and Management, EKAW'99, vol. 1621, Springer Verlag, 1999.

- IEEE Guide for Developing System Requirements Specifications*, IEEE Standard 1233-1996, 17 April 1996.
- Jin Z., Bell D., Wilkie F.G., *Automatically Acquiring the Requirements of Business Information Systems by Using Business Ontology*, 13th European Conference on Artificial Intelligence, ECAI'98, Brighton, England August 23-28, 1998.
- Kim J., Gil Y., *Towards Interactive Composition of Semantic Web Services*, AAAI Spring Symposium, AAAI Press, 2004, s. 100-107.
- Kowalczykiewicz K., Nawrocki J., *Dokumentowanie wymagań w języku XML*, III Krajowa Konferencja Inżynierii Oprogramowania KKIO 2001, Mikom, Warszawa 2001, s. 83-92.
- Lamsweerde A., *Formal Specification: a Roadmap*, International Conference on Software Engineering, Limerick, Ireland, 2000, s. 147-159.
- Lee J., Xue N., Kuo J., *Structuring Requirement Specifications with Goals*, „Information and Software Technology” 2001 nr 43, s. 121-135.
- Leffingwell D., Widrig D., *Managing Software Requirements: A Unified Approach*, Addison-Wesley Pub Co, 2000.
- Lin C.Y., Ho C.S., *An Ontology-Based Approach to Acquiring Domain Knowledge for Requirement Analysis*, Proceedings of the National Science Council, vol. 24 nr 1, 2000, s. 44-60.
- Łabuzek M., *Wykorzystanie metamodelowania do specyfikacji ontologii znaczenia opisów rzeczywistości*, Politechnika Wrocławska, Serwis e-informatyka.pl, Internet, styczeń 2004.
- Maedche A., Staab S., *Ontology Learning for the Semantic Web*, IEEE Intelligent Systems, Volume 16, Issue 2, March 2001, s. 72-79.
- Medjahed B., Bouguettaya A., Elmagarmid., *Composing Web services on the Semantic Web*, „The VLDB Journal – The International Journal on Very Large Data Bases” vol. 12, Issue 4, November 2003, s. 333-351.
- Moor de A., Weigand H., *An Ontological Framework for User-Driven System Specification*, Thirty-second Annual Hawaii International Conference on System Sciences, vol. 8, Maui, Hawaii, January 05-08, 1999.
- Noy N., Musen M., *SMART: Automated Support for Ontology Merging and Alignment*, Twelfth Banff Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Alberta, Canada, 1999.
- Noy N., Musen M., *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*, Proceedings of the Seventeenth National Conference on Artificial Intelligence, 2000, s. 450-455.
- Osterwalder A., Pigneur Y., *An e-Business Model Ontology for the Creation of New Management Software Tools and IS Requirement Engineering*, Proceeding of the CAiSE'2002 Doctoral Consortium, Toronto 2002.
- Pahl C., Casey M., *Ontology Support for Web Service Processes*, Proceedings of the 9th European software engineering conference, Helsinki (Finland) 2003, s. 208-216.
- Radosiński E., *Systemy informatyczne w dynamicznej analizie decyzyjnej*, Wydawnictwo Naukowe PWN, Warszawa-Wrocław 2001.
- Regnell B., Andersson M., Bergstrand J., *A Hierarchical Use Case Model with Graphical Representation*, Proceedings of ECBS'96, IEEE International Symposium and Workshop on Engineering of Computer-Based Systems, March 1996.
- Robinson W., Pawlowski S., Volkov V., *Requirements Interaction Management*, Georgia State University Working Paper 99-7, USA, August 30, 1999.
- Ruhe G., Bomarius F., *Learning Software Organization – Methodology and Applications*, Lecture Notes in Computer Science, vol. 1756, Springer 2000.
- Santander V., Castro J., *Developing Use Cases from Organizational Modeling*, WER'2001-IV Workshop de Engenharia de Requisitos; Buenos Aires 2001.
- Tse T., Pong L., *An Examination of Requirements Specification Languages*, „The Computer Journal” 1991 vol. 34, s. 143-152.

ANALYSIS OF THE TECHNIQUES TO REPRESENT REQUIREMENTS IN THE PROCESS OF BUILDING INFORMATION SYSTEMS

Summary

Representation of the requirements acquired from the requirements stakeholders in particular, constitute critical factors affecting the success of developing information systems. Such representation should constitute a coherent set in spite that it is usually received from very different professional groups, of widely varying competences, participating in the implementation of a specific project. This is made difficult by the fact that the requirements collected from the main stakeholders are imprecise and very weakly structured.

The article comprises review of techniques to represent the requirements in the process of building information systems. The research shall focus on new techniques for requirements representation – using knowledge engineering (especially using ontology). This is associated with the fact that an approach is becoming more and more popular according to which “the most essential thing is the knowledge held by people and organisations, and the knowledge should be articulated and formalised before attempting to build” an information system.

Dr inż. Andrzej Sobczak jest adiunktem w Katedrze Informatyki Gospodarczej Szkoły Głównej Handlowej w Warszawie
e-mail: andrzej.sobczak@sgh.waw.pl.