

Małgorzata Pańkowska

JĘZYK SYSML W INŻYNIERII WYMAGAŃ

Wprowadzenie

W konstrukcji każdego systemu ważne jest by wymagania były właściwie rozumiane. Inżynieria systemów oparta na modelach może pomóc w zarządzaniu złożonością, wspomaga doskonalenie jakości produktów i cyklu życia systemu informatycznego, usprawnia komunikację między zespołami oraz ułatwia gromadzenie wiedzy generowanej w projekcie i ewolucję projektu. W opracowaniu przedstawiono zasadnicze własności języka inżynierii systemów – SysML ze zwróceniem uwagi na jego użyteczność w inżynierii wymagań.

1. Inżynieria systemów i wymagań

Inżynieria systemów jest dyscypliną, która koncentruje się na projektowaniu i zastosowaniu systemu jako całości złożonej z części. To wymaga ujęcia holistycznego, przy uwzględnieniu różnych aspektów zarówno technicznych, jak i społecznych. Inżynieria systemów jest procesem iteracyjnym planowania, rozwoju i administrowania złożonymi systemami, szczególnie systemami komputerowymi [EnRo03]. Inżynieria oprogramowania jest tylko częścią inżynierii systemów, która dotyczy systematycznego rozwoju, ewaluacji i konserwacji oprogramowania. Proces inżynierii systemów jest zgodny z modelem SIMILAR:

- postawienie problemu (*state the problem*),
- wyszczególnienie alternatyw (*investigate alternatives*),
- modelowanie systemu (*modelling system*),
- integracja (*integration*),
- uruchomienie systemu (*launching the system*),
- oszacowanie rezultatów (*assess performance*),
- ponowna ocena (*re-evaluation*) [WrMa10].

Wymaganie powinno być zawsze traktowane jako wymaganie systemowe. Wymagania odnośnie do oprogramowania są tylko częścią wymagań systemowych. Inżynieria wymagań staje się istotna z wielu powodów:

- wszystkie podejścia systemowe rozpoznają potrzebę zrozumienia wymagań we właściwy sposób, a to wiąże ze sobą wszystkie rodzaje systemów, technicznych, społecznych, finansowych etc.,
- jakość i dopasowanie do celu, działanie systemu zgodnie z oczekiwaniami,
- projekt sterowany wymaganiami, brak zrozumienia wymagań projektowych generuje brak zrozumienia całości systemu,
- benchmarking i testowanie dla uzyskania akceptacji przez użytkownika,
- generowanie zaufania wśród interesariuszy projektu poprzez demonstracje systemu do akceptacji,
- pełne śledzenie wymagań,
- niezależność wymagań od rozwiązania [HoPe10].

Zarządzanie wymaganiami obejmuje wszystkie działania dla zapewnienia, że artefakty tworzone przy zastosowaniu metod inżynierii wymagań są użyteczne w całym projekcie. Zarządzanie wymaganiami integruje wszystkie dostępne dane projektowe z dostępnymi danymi dotyczącymi wymagań i może być postrzegane jako istota administracji projektowej. Kluczowe działania inżynierii wymagań to ujawnienie, specyfikacja, analiza i przegląd wymagań. Dla rozpoznania wymagań stosuje się techniki, takie jak opisywanie wybranego fragmentu rzeczywistości, analiza dokumentów, listy sprawdzające, burze mózgów, mapy myśli dla graficznej prezentacji powiązań z wybranym obiektem, wywiady, obserwacje, przypadki użycia, scenariusze i opowiadania. Poza tym metody te mogą być stosowane dla identyfikacji oraz definiowania interfejsów, interesariuszy i ich ról w projekcie. Techniki rejestracji wymagań zależą od charakterystyki branżowej przedsiębiorstwa, dla którego jest budowany system, jego aplikacji i projektów. Istnieje wiele technik podstawowych, wspomagających inżynierię wymagań, a zaliczyć do nich można: wywiady, zarówno formalne, jak i nieformalne, wymagania biznesowe zapisane w dokumentach przedsiębiorstwa, informacje zwrotne konserwatorów produktów software'owych, okresowe badania opinii użytkowników, pracę w środowisku docelowym eksploatacji oprogramowania, konferencje użytkowników, studia czasopism fachowych i materiałów handlowych oraz prototypowanie zapewniające użytkownikom wizualizację efektów częściowych produkcji oprogramowania. Wymagania są definiowane w całym cyklu życia systemu, czyli na etapie konceptualizacji, rozwoju, aprobowania projektu, realizacji systemu, konserwacji, aktualizacji i wycofywania z eksploatacji. Inżynieria wymagań obejmuje określenie wymagań klienta i ich akceptację, określenie wymagań dotyczących projektu oraz wymagań systemu, ich integrację i testowanie systemu. Istotne problemy inżynierii wymagań obejmują:

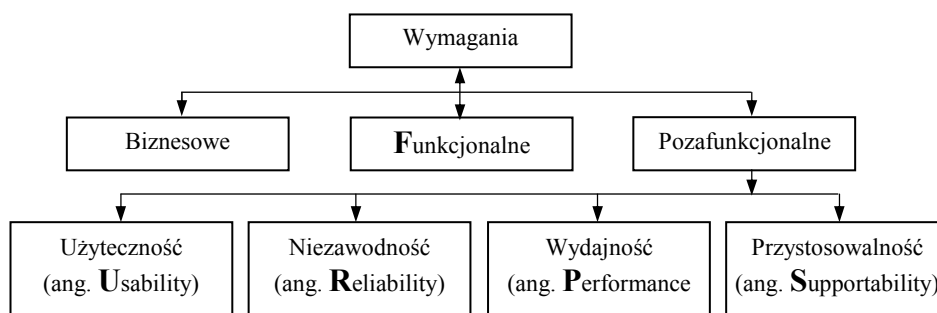
- identyfikowalność wymagań i konieczność zachowania troski, by każde wymaganie zostało unikatowo nazwane i opisane,

- filtrowalność wymagań i zapewnienie możliwości ekstrakcji informacji stosownie do roli interesariusza projektu,
- możliwość śledzenia, czyli obserwowanie różnych komponentów informacji w tym samym czasie lub śledzenie fragmentu informacji (wymagania) w dłuższym przedziale czasowym,
- wiązanie wymagań ze sobą, a także z testami, kryteriami jakości, rolami,
- informacje dodatkowe, tj. autor i właściciel wymagania, data utworzenia, komentarze, priorytety,
- prawa użytkownika odnośnie do przeglądania i edytowania wymagań [HWFPa08].

Należy troszczyć się o specyfikację wymagań służyć minimalizowaniu ryzyka, które może być analizowane w różnych kontekstach, jako ryzyko biznesowe, techniczne, finansowe, środowiskowe. W inżynierii systemów istotne są trzy zagrożenia: złożoność wymagań, brak zrozumienia i problemy nieefektywnego komunikowania się interesariuszy projektu [HoPe10].

2. Wymagania i ich własności

W literaturze przedmiotu wyróżnia się trzy rodzaje wymagań: biznesowe, funkcjonalne i pozafunkcjonalne [HoPe10; GrCa87]. Wymagania biznesowe przedstawiają podstawową charakterystykę przedsiębiorstwa i są oparte na strategii biznesowej, planowaniu zmian, analizie rynków obecnych i przyszłych, misji przedsiębiorstwa. Wymaganie funkcjonalne jest ogólnie utożsamiane z wymaganiami użytkownika. Wymagania funkcjonalne i pozafunkcjonalne ujęte są w modelu FURPS (por. rys. 1). Pozafunkcjonalne wymagania reprezentują zalecenia, które ograniczają w pewien sposób inne wymagania. Przykłady wymagań pozafunkcjonalnych obejmują wymagania jakości, wymagania wdrożeniowe, wymagania zastosowania specyficznych rozwiązań.

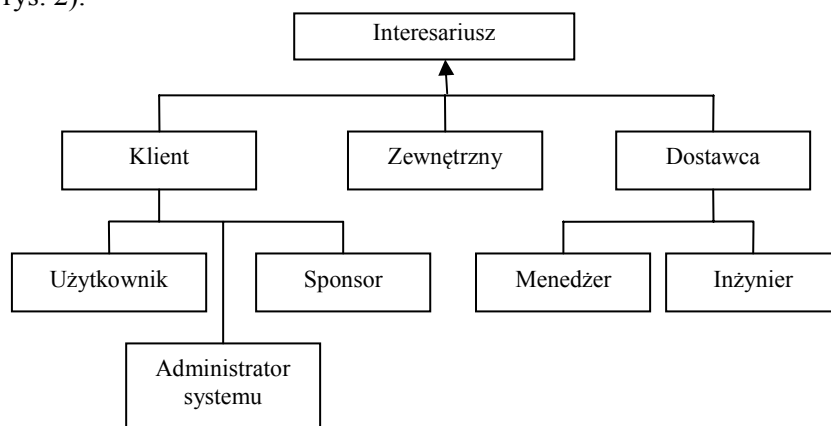


Rys. 1. Wymagania biznesowe i modelu FURPS

Źródło: [GrCa87].

Wymaganie może być prezentowane graficznie jako klasa, a własności wymagania są pokazane jako lista atrybutów. Własności wymagań są następujące: identyfikator, tekst, źródło, priorytet, kryteria weryfikacji i walidacji, opis, aktualność, stabilność. Uogólniony proces zarządzania wymaganiami może również być przedstawiony jako klasa. Atrybuty tej klasy to zbiór interesariuszy, zbiór wymagań, kryteria akceptacji, zbiór wyników przeglądu, wymagania szczególne (specyficzne), dokument rejestracji wymagań, opis projektu. Operacje na tej klasie są zdefiniowane następująco: eksponowanie i oczyszczanie wymagań, definiowanie kontekstu biznesowego, organizacji biznesowej, ról, projektów i usług, cyklu życia projektu, identyfikowanie ryzyka, analiza wymagań, przegląd, definiowanie kryteriów akceptacji, identyfikowanie interesariuszy, dokumentowanie procesu zarządzania wymaganiami [HoPe10]. Interesariuszem może być:

- klient, który korzysta z usług w systemie,
- interesariusz zewnętrzny, np. klient użytkownika, internauta,
- dostawca zaangażowany w rozwój i dostarczanie produktów i usług (por. rys. 2).



Rys. 2. Interesariusze inżynierii wymagań

Źródło: [HoPe10].

W inżynierii wymagań poza modelowaniem wymagań konieczne jest także modelowanie kontekstu. Przyjmuje się, że kontekst określa granice systemu i rozdziela świat zewnętrzny od możliwości i wymagań systemu. Używając technik modelowania, kontekst jest wizualizowany przy użyciu diagramu przypadku użycia i analizowany przy zastosowaniu scenariuszy, przy założeniu, że scenariusz jest sekwencją wydarzeń, które prowadzą do osiągnięcia konkretnego celu. Kontekst przedstawia system z punktu widzenia jakiegoś interesariusza. Zdaniem Jakubczyc i Owoca kontekst to wszystko to, co pośrednio wymusza rozumienie i interpretację pojęć, a więc pojedynczy czynnik lub sytuacja o różnym poziomie złożoności [JaOw11]. Kontekst w modelowaniu systemu służy jako odniesienie znaczeniowe, interpretacyjne lub sterujące.

3. Język modelowania systemów

Systems Modeling Language (SysML) jest językiem modelowania ogólnego zastosowania, który wspomaga specyfikację, analizę, projektowanie i weryfikację systemów informatycznych. SysML jest językiem modelowania graficznego przy zastosowaniu podstaw prezentowania wymagań, zachowania, struktury, własności systemu i jego komponentów [FMSt09]. UML 2 został wybrany jako podstawa dla SysML, co ułatwia integrację modelowania systemów i modelowania oprogramowania. Specyfikacja OMG SysML [WWW1] została przyjęta przez OMG w maju 2006 roku i formalna wersja 1.0 specyfikacji języka została wydana w sierpniu 2007 roku. SysML jest oparty na języku UML, więc nie może być traktowany jako całkowicie nowy język, lecz jako zbiór dodatkowych uzupełnień do istniejącego rdzenia koncepcji i diagramów modelowania UML. SysML korzysta z podzbioru UML2.1, co zwane jest UML4SysML. Znaczna część koncepcji UML została odrzucona, ponieważ uznano je za nieodpowiednie dla modelowania w inżynierii systemów. Zachowano diagram maszyny stanowej, sekwencji, diagram przypadku użycia. Inne diagramy zostały rozszerzone, tak jak diagramy aktywności.

Tabela 1

Porównanie SysML i UML2.x

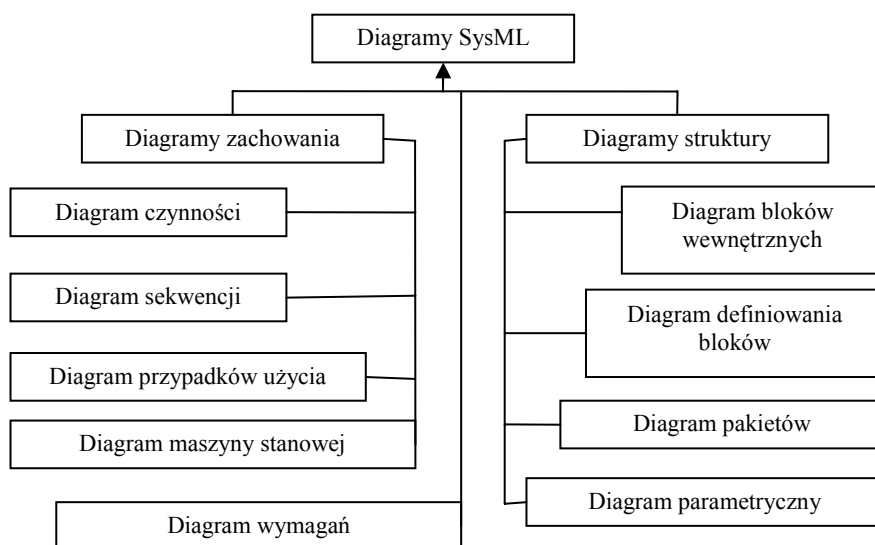
Diagram SysML	Opis	Odpowiednik w języku UML2.x
1	2	3
Diagram wymagań systemowych	Graficzne przedstawienie wymagań systemowych i ich relacji z innymi kategoriami modelowania systemu	Brak
Diagram przypadków użycia	Graficzne przedstawienie przypadków użycia, aktorów oraz związków między nimi	Diagram przypadków użycia
Rozszerzony diagram czynności	Graficzne przedstawienie sekwencyjnych lub współbieżnych przepływów sterowania oraz danych pomiędzy uporządkowanymi ciągami czynności, akcji i obiektów	Diagram czynności
Diagram sekwencji	Graficzne przedstawienie interakcji między aktorami, blokami, częściami bloków i obiektami w postaci sekwencji komunikatów wymienianych między kategoriami modelowania	Diagram sekwencji
Diagram maszyny stanowej	Graficzne odzwierciedlenie dyskretnych zachowań systemów	Diagram maszyny stanowej

cd. tabeli 1

1	2	3
Diagram definiowania bloków	Graficzne przedstawienie struktury systemu w postaci bloków, ich cech i związków	Diagram klas
Diagram bloków wewnętrznych	Graficzne przedstawienie wewnętrznej struktury bloku, wyrażanej przez wzajemnie powiązane części bloków	Diagram struktur połączonych
Diagram parametryczny	Przedstawienie ograniczeń parametrycznych występujących w modelowanym systemie	Brak
Diagram pakietów	Graficzne przedstawienie logicznej struktury systemu w postaci zestawu pakietów połączonych zależnościami i zagnieżdżeniami	Diagram pakietów

Źródło: Na podstawie: [WrM10a].

SysML odrzucił diagramy UML, takie jak diagram obiektu, komponentu, wdrożeniowe, komunikacji, przedziałów czasowych i interakcji. Diagramy struktury złożonej i klas zostały znacznie zmodyfikowane oraz zastąpione definicją blokową i diagramami bloków wewnętrznych. Ponadto, dodano dwa nowe diagramy, czyli diagram wymagań i parametryczny.



Rys. 3. Diagramy SysML

Źródło: [HoPe10; WrM10a].

Język SysML w wersji 1.1 zawiera 9 rodzajów diagramów, są to:

- diagram wymagań systemowych, przedstawia wymagania i ich relacje z innymi wymaganiami, elementami projektu, przypadkami testowymi dla wspomagania śledzenia wymagań,
- diagram czynności przedstawia działania oraz transformacje nakładów na efekty,
- diagram sekwencji przedstawia zachowanie w kategoriach sekwencji komunikatów wymienianych między stronami,
- diagram przypadków użycia przedstawia funkcjonalności używane przez jednostki zewnętrzne (aktorzy) dla osiągnięcia zbioru celów,
- diagram maszyny stanowej, przedstawia zachowania encji w kategoriach jej przejścia między etapami uruchamianymi przez wydarzenia
- diagram definiowania bloków przedstawia strukturalne elementy zwane blokami, ich kompozycje i klasyfikacje,
- diagram bloków wewnętrznych przedstawia interpołączenia i interfejsy między częściami bloków,
- diagram parametryczny przedstawia ograniczenia nałożone na wartości, stosowane do wspomagania analizy systemu,
- diagram pakietów przedstawia organizację modelu w kategoriach pakietów, które zawierają elementy modelu [HoPe10].

SysML oferuje możliwości modelowania, które pozwalają na prezentowanie systemów i ich komponentów przy użyciu:

- kompozycji komponentów strukturalnych, klasyfikacji i łączenia,
- modeli zachowań, co obejmuje przepływy działań, scenariusze interakcji, przekazywanie komunikatów, jak też zachowania reaktywne zależnie od stanu,
- przypisanie jednego elementu modelu do innego, jak np. funkcji do komponentów, oprogramowania do sprzętu,
- ograniczeń nałożonych na system, takich jak dotyczących wykonania, wiarygodności, własności fizycznych,
- hierarchii wymagań i ich relacji z innymi elementami modelu [DFJS10].

Oczekuje się, że SysML zapewni unifikację zróżnicowanych języków modelowania obecnie stosowanych, podobnie jako UML. SysML pozwala na modelowanie wymagań systemu, zachowanie systemu i strukturyzację systemu. Wprowadzenie diagramu wymagań w SysML pozwala na modelowanie strukturalnych relacji między wymaganiami, mimo że wymagania są tradycyjnie przedstawiane przy użyciu diagramów przypadków użycia, które ujmują wymagania w aspekcie zachowań.

4. Diagramowanie wymagań

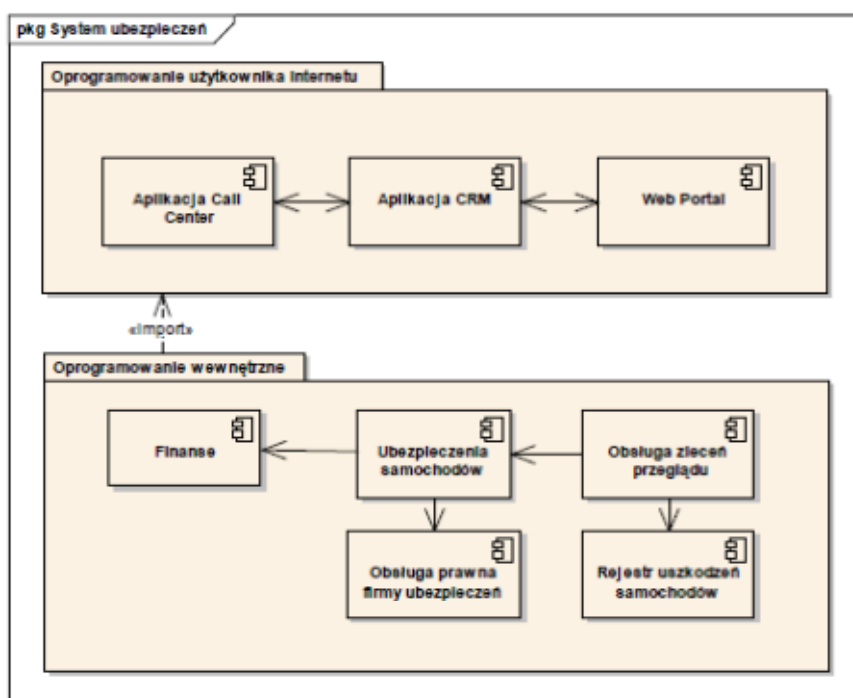
W języku SysML wymagania oznaczają kontrakt między organizacją zamawiającą system a jego wykonawcami. Podstawowymi własnościami, które charakteryzują wymagania są w SysML numer porządkowy i treść wymagania. W praktycznych zastosowaniach wymagania są uporządkowane hierarchicznie. W języku SysML diagram wymagań systemowych umożliwia graficzne przedstawienie wymagań systemowych i ich relacji z innymi kategoriami modelowania systemu. Wymagania specyfikuje się na podstawie kategorii modelowania diagramów wymagań systemowych, tj.:

- wymaganie (*requirement*),
- związek (*relationship*),
- blok (*block*),
- przypadek użycia (*use case*),
- testowy przypadek użycia (*test case*),
- pakiet (*package*).

Wymagania w diagramach wymagań systemowych języka SysML łączy się przez związki zagnieżdżenia (*containment*) umożliwiające tworzenie wielopoziomowej hierarchii wymagań oraz przez zależności (*dependencies*). Wskazują one na charakter logicznej zależności między poszczególnymi wymaganiami. Bloki, przypadki użycia i testowe przypadki użycia są kategoriami modelowania, istotnymi z punktu widzenia precyzji opisu kontekstu poszczególnych wymagań oraz monitorowania sposobu ich implementacji w systemie. Na diagramach wymagań systemowych stosuje się je w połączeniu z odpowiednimi rodzajami zależności. Blok w SysML jest podstawową jednostką struktury używaną dla definiowania encji. Encją może być komponent systemu, część sprzętu, oprogramowania, komponent danych, osoba, usługa. Bloki mają zdefiniowane własności, które można podzielić na własności strukturalne, własności interakcji i ograniczenia. W SysML wyróżniono następujące rodzaje własności:

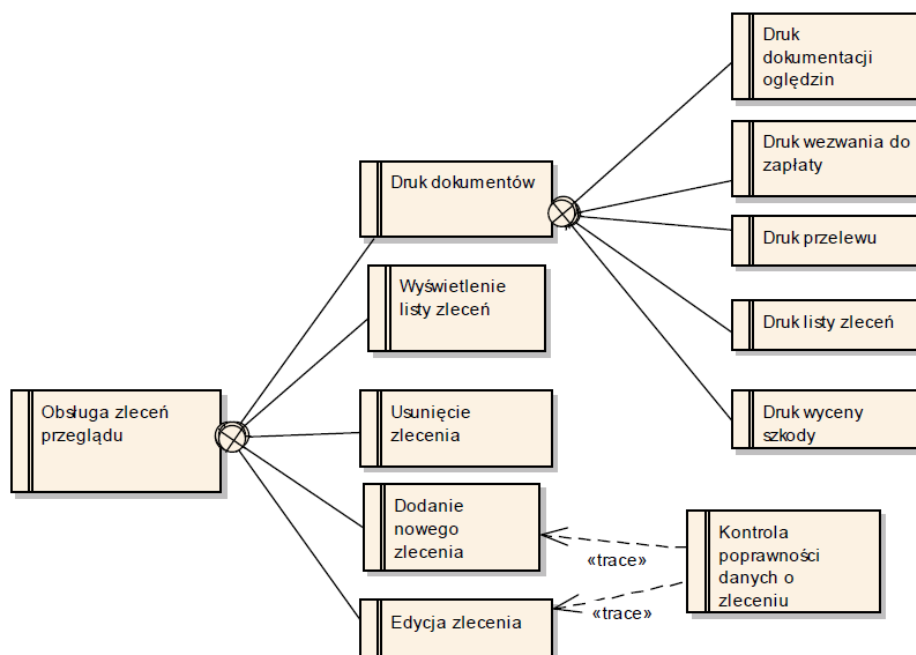
- własności części, które opisują hierarchiczną dekompozycję bloku,
- własności referencyjne, które opisują relacje między blokami,
- własności wartości, które opisują ilościowe charakterystyki bloku [FMSt09].

W SysML pakiety stanowią mechanizm ogólnego zastosowania, służący do organizowania dokumentacji, w tym dokumentacji wymagań. Pakiety i Diagramy pakietów są użyteczne w zarządzaniu złożonością modelu wymagań systemowych. Zastosowanie diagramów pakietów przedstawiono na rys. 4.



Rys. 4. Diagramy pakietów dla oprogramowania firmy ubezpieczeniowej

W SysML wymaganie przedstawia cechę, własność, zachowanie systemu. Zadanie definiowania i listowania wymagań jest wykonane w pierwszych krokach procesu projektowania systemu. Wymagania pozwalają projektantowi jasno ustalić czego się oczekuje od przyszłego systemu. Ponadto, wymagania tworzą punkt centralny procesu weryfikacji i walidacji, ponieważ są one kluczowym składnikiem dla dokładnego ustalania, co budowany system powinien robić i jak powinien to robić. SysML wprowadza diagram wymagań, którego wcześniej nie było w UML. Ten nowy diagram zapewnia środki, by wskazać wymagania i związać je z innymi modelami specyfikacji, projektowania i weryfikacji. Wymagania mogą być prezentowane w formie graficznej, tabularycznej lub struktury drzewiastej. Przykładowy diagram wymagań dla obsługi zlecenia firmy ubezpieczeniowej przedstawiono na rys. 5.



Rys. 5. Diagram wymagań obsługi zlecenia

Wymaganie może być dekomponowane na subwymaganie w celu uporządkowania wielu wymagań jako drzewa wymagań składowych. Wymaganie może być związane z innymi wymaganiami, może być generowane lub wydobyte z innego wymagania. Ponadto, wymaganie może być spełnione przez pewne elementy modelu korzystając z relacji satysfakcji. W SysML diagram wymagań jest utworzony z trzech podstawowych elementów: wymagań, relacji i przypadków testowych. Wymaganie są stosowane dla prezentacji wymagań systemu, które mogą być zarówno wzajemnie powiązane, jak i powiązane z innymi elementami systemu przez relacje. Przypadki testowe mogą być przyłączone do wymagań dla pokazania jak wymagania są testowane. W SysML relacje wymagań są następujące:

- zagnieżdżenie (*nesting*): wymaganie może być dekomponowane na jedno lub więcej wymagań,
- wyprowadzenie (*derive*): jedno lub więcej wymagań pochodzi z danego wymagania,
- kopiowanie (*copy*): dane wymaganie może być użyte w innym kontekście, przy czym powielone wymaganie może być wersją tylko do odczytu oryginalnego wymagania, wymaganie powielone nie może być zmienione bez zmiany oryginału,

- weryfikowanie (*verify*): pokazanie jak przypadek testowy weryfikuje wymaganie,
- satysfakcja (*satisfy*): pokazanie, że element modelu spełnia wymaganie,
- śledzenie (*trace*): pokazanie, że element modelu i wymaganie są ściśle powiązane [DFJS10].

Diagram wymagań jest sposobem dokumentowania i zarządzania wymaganiami, ponadto może być punktem odniesienia dla przypadków użycia i scenariuszy.

Podsumowanie

Ogólnie ujmując wymagania sterują całością projektu, zatem wszystkie modele i informacje w projekcie powinny wspomagać zarządzanie wymaganiami. Dla oceny wymagań są formułowane metryki i kryteria, które dzielą się na kryteria ogólnej specyfikacji i kryteria poszczególnych wymagań. Kompletność, zgodność, konieczność, swoboda duplikacji stanowią przykładowe kryteria ogólne, natomiast do kryteriów szczególnych zalicza się niesprzeczność, wykonalność, identyfikowalność, atomiczność, poprawność dowodzenia, możliwość monitorowania i duplikowania. Sukces końcowy projektu zależy od tego, czy projekt zgodny jest z wymaganiami, zatem rozwój języka modelowania systemów SysML świadczy o zrozumieniu tych potrzeb i zapełnił istotną lukę w procesach budowy systemów informatycznych.

Literatura

- [DFJS10] Debbabi M., Hassaine Fawzi, Jarraya Y., Soeanu A., Alawneh L.: *Verification and Validation in Systems Engineering, Assessing UML/SysML Design Models*. Springer-Verlag, Berlin, Heidelberg 2010.
- [EnRo03] Endres A., Rombach D.: *A Handbook of Software and Systems Engineering, Empirical Observations, Laws and Theories*. Pearson Addison Wesley, London 2003.
- [FMSt09] Friedenthal S., Moore A., Steiner R.: *A Practical Guide to SysML, The Systems Modeling Language*. Morgan Kaufman, OMG Press, Burlington 2009.
- [GrCa87] Grady R., Caswell D.: *Software Metrics: Establishing a Company-wide Program*. Prentice Hall, New York 1987.
- [HoPe10] Holt J., Perry S.: *Modelling Enterprise Architectures*. The Institution of Engineering and Technology, London 2010.
- [HWFPa08] Hood C., Wiedemann S., Fichtinger S., Pautz U.: *Requirements Management, The Interface Between Requirements Development and All Other Systems Engineering Processes*. Springer, Berlin 2008.

- [JaOw11] Jakubczyc J.A., Owoc M.: *Reprezentowanie kontekstu, Wiedza i komunikacja w innowacyjnych organizacjach*. W: *Komunikacja elektroniczna*. Red. M. Pańkowska. Wydawnictwo UE, Katowice 2011.
- [WrMa10] Wrycza S., Marcinkowski B.: *Specyfikacja wymagań systemowych w języku SysML*. W: *Technologie informatyczne w administracji publicznej i służbie zdrowia*. Red. J. Goliński, A. Kobyliński, A. Sobczak. SGH, Warszawa 2010.
- [WrM10a] Wrycza S., Marcinkowski B.: *Język inżynierii systemów SysML. Architektura i zastosowanie, Profile UML 2.x w praktyce*. Helion, Gliwice 2010.
- [WWW1] <http://www.omg.sysml.org>.

SYSML LANGUAGE IN REQUIREMENTS ENGINEERING

Summary

The paper covers justification of the necessity to develop requirements specification for the high quality of information systems. First, system engineering and requirements engineering as disciplines are explained. Next, author focuses on information system requirements' classification and explains their features. Third, the Systems Modeling Language (SysML) is characterized and compared with UML 2.0. The last part includes presentation of SysML diagrams and discussions on their usefulness on the business information systems modeling.