

Monika Łobaziewicz

Standaryzacja funkcjonalności i modelowanie systemu B2B

Problemy Zarządzania 13/2 (1), 125-135

2015

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej bazhum.muzhp.pl, gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

Standaryzacja funkcjonalności i modelowanie systemu B2B

Nadesłany 22.10.14 | Zaakceptowany do druku 06.11.14

Monika Łobaziewicz*

Celem artykułu jest prezentacja wyników badań przeprowadzonych w ramach drugiego etapu projektu „Opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych”, które doprowadziły do standaryzacji funkcjonalności głównych modułów systemu typu B2B oraz opracowania modelu jego budowy. W pierwszej części artykułu omówiono standard zapisu funkcjonalności w systemie B2B, w drugiej zaś zaprezentowano model systemu B2B.

Słowa kluczowe: system B2B, Unified Modeling Language, standaryzacja funkcjonalności systemu B2B, modelowanie systemu B2B.

Standardizing the Functionalities and Modeling of the B2B System

Submitted 22.10.14 | Accepted 06.11.14

The aim of the paper is to present the results of research carried out within the framework of the second phase of the project called “The development of a modern and advanced B2B system based on Internet technologies through research and development works”, which led to the standardization of the functionality of system core modules and the design of a B2B system model. The first part of the paper discusses the standard recording functionality of the B2B system, and the next part presents the model of the B2B system.

Keywords: B2B system, Unified Modeling Language, functional standardization of B2B system, B2B system modeling.

JEL: M15

* **Monika Łobaziewicz** – dr inż., Katolicki Uniwersytet Lubelski, Wydział Nauk Społecznych, Instytut Ekonomii i Zarządzania, Katedra Zarządzania Przedsiębiorstwem.

Adres do korespondencji: Katolicki Uniwersytet Lubelski Jana Pawła II, Al. Racławickie 14, 20-950 Lublin;
e-mail: mmlobaza@kul.lublin.pl.

1. Wprowadzenie

Systemy B2B są skierowane przede wszystkim do przedsiębiorstw współpracujących ze sobą w formie zamkniętej grupy w celu osiągnięcia korzyści biznesowych, przy czym każdy z podmiotów zachowuje swoją niezależność gospodarczą i nie musi być z tej samej branży. Ze względu na różnorodność systemów informatycznych kooperujących przedsiębiorstw i jednocześnie realizację tego samego rodzaju procesów konieczna jest standaryzacja funkcjonalności obsługiwanych poprzez moduły systemu B2B. Partnerzy, logując się do platformy, mają z góry określone zadania do wykonania (np. realizacja zakupów, sprawdzenie oferty, kontrola rozliczeń) w ramach zdefiniowanych procesów biznesowych. Niezbędne staje się więc takie opracowanie modułów systemu, aby jak najlepiej realizowały założone funkcjonalności procesów, jednocześnie nie powodując ograniczenia działalności firm kooperujących. W związku z tym moduły powinny umożliwiać szybkie dostarczanie odpowiednich danych i realizację zadań przez odpowiednią graficzną prezentację oraz ułatwienia funkcjonalne uwzględniające najczęściej wykonywane operacje. Cele te powinny być osiągnięte między innymi poprzez prezentację najważniejszych danych (np. kody producentów, miniaturki zdjęć, prezentację gabarytów produktów, opakowań zbiorczych), prosty i szybki sposób dodawania dużych ilości towarów do koszyka (np. listy towarowe z szybką inkrementacją zamawianych ilości).

Systemy B2B współpracują bezpośrednio z innymi systemami informatycznymi, które wspomagają realizację procesów wewnętrznych w przedsiębiorstwach. Najczęściej są to rozwiązania klasy ERP, dlatego też należy tak standaryzować funkcjonalności i konstruować moduły platformy B2B, aby obsługiwały rozwiązania zawarte w różnych aplikacjach ERP. Dotyczy to sposobu realizacji głównych procesów, np. oferty, zamówień, reklamacji, zwrotów czy realizacji wysyłek paczek. Integracja platformy B2B z systemem ERP powinna być nieinwazyjna dla przedsiębiorstwa i firm współpracujących. Oznacza to, że w trakcie jej wdrażania nie występuje konieczność przebudowy istniejących rozwiązań w systemie ERP. Współpraca między platformą B2B a systemem ERP powinna być stabilna, wymiana danych w pełni on-line w obu kierunkach oraz muszą być zapewnione odpowiednie standardy zabezpieczeń wymienianych danych.

Podczas projektowania modułów i funkcjonalności systemu B2B powinny być zatem uwzględniane wszystkie niezbędne dane, które mają być udostępnione z systemu ERP i które opisują procesy realizowane przez platformę, np. wyświetlanie statusów dokumentów, informowanie o zdarzeniach w procesach. Platforma B2B powinna zapewniać niskie koszty wejścia, co jest cechą charakterystyczną dla tego typu rozwiązań. Oznacza to, że powinna mieć możliwość łatwego rozbudowania, przy czym koszty tej rozbudowy powinno ponosić przedsiębiorstwo, które jest jednocześnie właścicielem i administratorem całego systemu.

Zatem konstrukcja modułów i funkcjonalności systemu B2B powinna uwzględniać podstawowe cechy opisujące dobre praktyki projektowania tego typu systemów: łatwość obsługi, skalowalność oraz wydajność.

Modelowanie systemu B2B ma przede wszystkim wpłynąć na sposób realizacji procesów biznesowych, ograniczyć liczbę wymaganych zasobów ludzkich, skrócić czas niezbędny do wykonywania poszczególnych działań, poprawić konkurencyjność firm kooperujących.

Biorąc pod uwagę powyższe przesłanki, celem artykułu jest prezentacja wyników analizy możliwości standaryzacji wydzielonych funkcjonalności i ich relacji w ramach systemu B2B, która została przeprowadzona w ramach drugiego etapu projektu pt. „Opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych”¹. Na podstawie analizy stworzono model systemu B2B.

Do realizacji tak sformułowanego celu zastosowano w projekcie następującą metodologię:

- analiza wyników pierwszego etapu badań określającego kierunki rozwoju platformy B2B²,
- modelowanie procesów, które będą wspierane przez system B2B,
- wydzielenie modułów systemu B2B obsługujących procesy,
- standaryzacja funkcjonalności systemu B2B,
- opracowanie modelu zależności między procesami w systemie B2B,
- szczegółowe określenie niezbędnych danych opisujących procesy i funkcjonalności wydzielone w ramach systemu, wsparte przez diagramy UML w postaci Use Case oraz diagramy procesów.

2. Standard zapisu funkcjonalności w systemie B2B

Tworzenie dużych i skomplikowanych systemów czy oprogramowania składającego się z wielu modułów, realizującego liczne funkcjonalności, o wysokim stopniu złożoności wymaga szczegółowego i jasnego w odbiorze sposobu projektowania i planowania. Odpowiednie przedstawienie w procesów, modułów, klas czy relacji ma znaczący wpływ na prawidłowość konstrukcji systemu, jego bezpieczeństwo oraz wydajność. Przy projektowaniu tego typu rozwiązań uczestniczy wiele osób, należy więc zapewnić zrozumiały i jednolity dla wszystkich sposób opisu zachowania systemu. Ze względu na istnienie wielu metodologii modelowania koniecznością jest ich ujednoczenie i wprowadzenie jednego wspólnego wzorca.

W pierwszym etapie badań dotyczących funkcjonalności modeli systemów B2B opartych na technologiach WWW oraz opracowania metody ich standaryzacji, zostały zidentyfikowane kluczowe wymagania dla systemu B2B. Kolejnym krokiem było przełożenie ich na konkretne procesy oraz moduły, które będą potem realizowane w prototypie systemu B2B. W tym celu zastosowano jedną z technik modelowania systemów informatycznych stosowanych w inżynierii oprogramowania, jakim jest język UML (*Unified Modeling Language*).

W celu ustalenia jednego standardu zapisu funkcjonalności zaprojektowane zostały specjalne *Karty działania* (tabele 1, 2), według których została dokonana charakterystyka projektowanych modułów.

Numer modułu	Numer modułu przyporządkowany w strukturze funkcjonalnej systemu
Nazwa modułu	Nazwa modułu w strukturze funkcjonalnej systemu
Przeznaczenie modułu	Możliwe zadanie do realizacji w danym module
Działania w module	Wyszczególnienie zadań do realizacji w ramach modułu

Tab. 1. Wzór karty działania do opisu modułu. Źródło: Opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych. Część 2. (2014). Rzeszów.

Numer działania	Numer działania przyporządkowany w strukturze funkcjonalnej systemu
Nazwa działania	Nazwa opisująca funkcję lub rezultat osiągany poprzez wykonanie danej funkcji
Zadanie (kroki postępowania)	Możliwe zadanie do realizacji w ramach działania
Wsad	Informacja, dane, pliki, dokumenty lub inne materiały, które przyczyniają się do uruchomienia funkcji systemu
Czynnik wyzwalający	Czynniki pozamaterialne (inne niż wsad), które przyczyniają się do uruchomienia funkcji systemu, np. decyzja, potrzeba, zakończenie danego etapu
Rezultat	Rezultat wykonania danej funkcji, np. nowy dokument, nowy plik, nowe zdarzenie

Tab. 2. Wzór karty działania do opisu działań w ramach modułu. Źródło: Opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych. Część 2. (2014). Rzeszów.

Każdy moduł został przedstawiony w perspektywie przypadków użycia (*Use Case*) języka UML, a procesy i czynności w postaci diagramów. System został podzielony na moduły skupiające tematycznie zgrupowane funkcjonalności. Dzięki takiej budowie, w zależności od potrzeb, w łatwy sposób udostępnione zostają odpowiednie widoki danych, które docelowo znajdują zastosowanie w firmach obsługiwanych przez system.

UML znajduje głównie zastosowanie w tworzeniu systemów informatycznych przedsiębiorstw oraz rozproszonych usług internetowych. Istotną cechą języka UML jest to, że pomimo nieklasyfikowania go jako języka programowania graficznego, modele opisane za jego pomocą mogą być powiązane z dużą liczbą języków programowania (np. C++, Visual Basic czy Java). UML składa się z dwóch podstawowych elementów: notacji oraz metamodelu (rysunek 1).

Notacja:	Metamodel:
<ul style="list-style-type: none"> – elementy graficzne – składnia języka modelowania – istota przy tworzeniu modeli 	<ul style="list-style-type: none"> – definicje pojęć języka i powiązania pomiędzy nimi – graficzne modelowanie

Rys. 1. Elementy języka UML. Źródło: <http://www.omg.org/spec/UML/2.2/> (10.03.2014).

UML definiuje dwie podstawowe składowe: notację poszczególnych elementów używanych na diagramach i ich semantykę, czyli tzw. metamodel. Z punktu widzenia analityka istotniejsze jest czytelne i jednoznaczne opisanie modelu tak, aby inne osoby mogły zrozumieć jego znaczenie. Zatem ważniejsza dla niego jest notacja, metamodel zaś powinien być zrozumiały intuicyjnie. Z kolei przy generowaniu kodu i przejściu do implementacji ważniejsze jest ściśle rozumienie znaczenia poszczególnych elementów, tak aby możliwa była automatyczna konwersja modelu do innego formalizmu.

Kolejną charakterystyką UML jest to, że umożliwia przedstawienie projektu z wielu prawie niezależnych perspektyw (Schmuller, 2003):

1. *Perspektywa przypadków użycia* (ang. *Use Case*) – służy głównie do określenia wymagań projektowanego systemu. Przypadki użycia wskazują wiele szczegółów, ale zapewniają również spojrzenie na projekt z szerszej perspektywy. *Use Case* są często określane jako plan systemu (McLaughlin, Pollice i West, 2006). Potrafią wskazać cele systemu w sposób bardziej uproszczony, a przez to bardziej zrozumiały dla zainteresowanych stron, natomiast potrafią modelować funkcjonalności systemu bardziej kompletnie niż diagramy klas (Siau i Lee, 2004, s. 229–237). Są zazwyczaj tworzone w pierwszych fazach projektowania rozwiązań. Łączą ze sobą aktorów (użytkowników) oraz możliwe działania w systemie, natomiast szczegóły ich przebiegu są projektowane za pomocą innych bardziej szczegółowych technik.
2. *Perspektywa projektowa* (logiczna) umożliwia opis sposobu realizacji funkcjonalności oraz przedstawienie systemu z punktu widzenia projektanta.
3. *Perspektywa implementacyjna* umożliwia przedstawienie systemu z punktu widzenia programisty i ilustrację struktury kodu systemu (implementacja klas, podsystemów, modułów czy interfejsów).
4. *Perspektywa procesowa* umożliwia podział systemu na procesy (czynności) i jednostki wykonawcze, opis zagadnień poza funkcjonalnych oraz przedstawienie systemu z punktu widzenia programisty i integratora, instalatora.
5. *Perspektywa wdrożeniowa* umożliwia podział fizyczny elementów systemu oraz ich rozmieszczenie w infrastrukturze, opis systemu dla integratorów i instalatorów.

Na rynku jest dostępnych wiele aplikacji wspierających tworzenie diagramów UML, od prostych darmowych aplikacji, jak np. ARGOUML (<http://argouml.org>), WHITESTARUML (<http://sourceforge.net/projects/whitesta>

ruml), po zaawansowane płatne narzędzia, np. MS VISIO (<http://office.microsoft.com>), VISUAL PARADIGM (<http://www.visual-paradigm.com>).

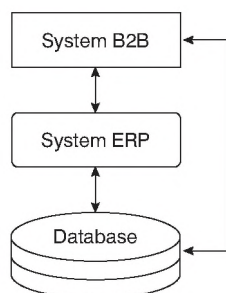
Ze względu na środowisko programistyczne, w którym projektowany system B2B jest tworzony, zostały wykorzystane narzędzia MS VISIO. Do jego zalet należy zaliczyć:

- tworzenie profesjonalnych diagramów,
- łatwość zrozumienia skomplikowanych danych dzięki nowym narzędziom i diagramom połączonymi z danymi,
- posiadanie szablonu modelowania UML,
- obsługę tworzenia modeli obiektowych,
- wyposażenie w eksplorator modelu UML oferujący widok drzewa modelu, za pomocą którego można przełączać się z jednego widoku na inny,
- wstępnie zdefiniowane kształty inteligentne przedstawiające elementy w notacji UML i obsługujące tworzenie typów diagramów UML (kształty są zaprogramowane, aby ich zachowanie było zgodne z semantyką UML),
- łatwość dostępu do okien dialogowych, za pomocą których można do elementów UML dodawać nazwy, atrybuty, operacje i inne właściwości,
- dynamiczne sprawdzanie błędów semantycznych, identyfikowanie i diagnozowanie błędów, takich jak brak danych czy nieprawidłowe użycie notacji UML,
- możliwość odtwarzania projektów utworzonych w programie Microsoft Visual C++ 6.0 lub Microsoft Visual Basic 6.0 w celu wygenerowania modeli struktur statycznych UML,
- możliwość odtwarzania projektów utworzonych w programie Microsoft Visual Studio.NET i generowania modeli struktur statycznych UML.

3. Modelowanie systemu B2B

System B2B stanowi jeden z elementów całości rozwiązania informatycznego, które wspiera określone procesy realizowane pomiędzy przedsiębiorstwami współpracującymi. Nie jest platformą zapewniającą całość funkcjonalności niezbędnej do obsługi firmy, dlatego jego budowa powinna zapewniać możliwość integracji z systemami ERP. Powinna stanowić konfigurowalny interfejs umożliwiający podgląd wybranych informacji zawartych w systemie ERP. Usługi WWW są najwyższą warstwą systemów B2B i powinny służyć do bezpośredniego kontaktu z klientem. Zgodnie z wymaganiami stawianymi systemom klasy ERP dane wprowadzane przez partnera za pośrednictwem platformy powinny zostać w czasie rzeczywistym rejestrowane w bazie danych obsługiwanego systemu. Analogicznie informacje zawarte w bazie danych są dostępne w trybie on-line w systemie B2B. Z platformy internetowej mogą korzystać firmy zdefiniowane w bazie danych, dla których zostały ustawione odpowiednie parametry. Zakres uprawnień poszczególnych użytkowników powinien być konfigurowalny. Dostęp do platformy za pomocą przeglądarki jest możliwy po podaniu unikatowych dla każdego użytkownika ID oraz hasła.

Na rysunku 2 przedstawiono ogólny schemat architektury systemowej, dla której projektowany jest system B2B.

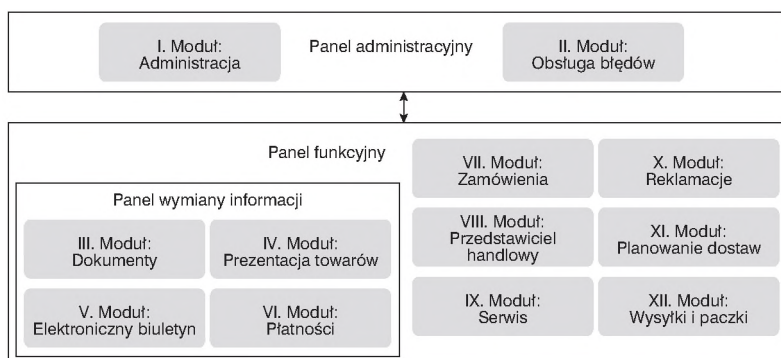


Rys. 2. Ogólny schemat komponentów całego systemu B2B. Źródło: opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych. Część 2. (2014). Rzeszów.

Przeprowadzone badania pozwoliły na wydzielenie następujących modułów w systemie B2B:

1. Administracja – moduł konfiguracyjny i serwisowy.
2. Dokumenty – moduł udostępniający końcowe dokumenty handlowe.
3. Zamówienia – moduł udostępniający dokumenty zamówień sprzedaży.
4. Płatności – moduł informacyjny zawierający aktualne dane rozliczeniowe.
5. Prezentacja towarów – moduł odpowiedzialny za prezentację oferty handlowej.
6. Reklamacje – moduł odpowiedzialny za wsparcie i obsługę procesu reklamacyjnego.
7. Elektroniczny biuletyn (newsletter) – moduł komunikacji z klientami.
8. Serwis – moduł odpowiedzialny za wsparcie i obsługę zleceń serwisowych.
9. Obsługa błędów – moduł pomocniczy służący do wykrywania i raportowania błędów systemu.
10. Przedstawiciel handlowy – moduł wspierający obsługę sprzedaży przez przedstawicieli handlowych.
11. Planowane dostawy – moduł wspierający informowanie klienta o dostępności towarów.
12. Wysyłki i paczki – moduł wspierający obsługę dostaw towarów do klienta końcowego.
13. Elektroniczne płatności – moduł do zdalnego opłacania zamówionych usług i towarów.

Moduły mają charakter logicznych komponentów systemu, funkcjonujących w ramach dwóch paneli zapewniających obsługę poszczególnych procesów wchodzących w ich skład oraz dedykowany i konfigurowalny dostęp użytkowników do systemu.



Rys. 3. Moduły systemu B2B w rozbiściu na główne panele. Źródło: opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych. Część 2. (2014). Rzeszów.

W ramach głównych modułów zostały wydzielone następujące funkcjonalności i procesy:

1. Administracja:
 - 1.1. Językowość (tłumaczenia).
 - 1.2. Uprawnienia.
 - 1.3. Śledzenie aktywności użytkownika.
 - 1.4. Dziennik logowań.
2. Dokumenty:
 - 2.1. Lista FS.
 - 2.2. Podgląd szczegółów FS.
 - 2.3. Lista WZ.
 - 2.4. Podgląd szczegółów WZ.
3. Zamówienia:
 - 3.1. Lista ZS.
 - 3.2. Podgląd szczegółów ZS.
 - 3.3. Lista ZW.
 - 3.4. Podgląd szczegółów ZW.
 - 3.5. Proces składania zamówienia.
4. Płatności:
 - 4.1. Lista płatności.
 - 4.2. Płatności elektroniczne.
 - 4.3. Generowanie formularzy poleceń przelewów oraz wpłat gotówkowych.
5. Prezentacja towarów:
 - 5.1. Lista grup towarowych.
 - 5.2. Lista towarów.
 - 5.3. Podgląd szczegółów towaru.
 - 5.4. Wyszukiwarka.
 - 5.5. Lista promocji.

6. Reklamacje:
 - 6.1. Lista reklamacji.
 - 6.2. Podgląd szczegółów reklamacji.
 - 6.3. Proces składania reklamacji.
7. Elektroniczny biuletyn (newsletter):
 - 7.1. Lista informacji.
8. Serwis:
 - 8.1. Lista zleceń serwisowych.
 - 8.2. Szczegóły zlecenia serwisowego.
 - 8.3. Proces składania zlecenia serwisowego.
9. Obsługa błędów:
 - 9.1. Wyświetlenie błędu (użytkownikowi).
 - 9.2. Rejestracja błędu do pliku/tabeli bazy danych.
 - 9.3. Wysłanie powiadomienia o błędzie.
10. Przedstawiciel handlowy:
 - 10.1. Lista kontrahentów.
 - 10.2. Podgląd szczegółów kontrahenta.
 - 10.3. Geolokalizacja.
 - 10.4. Raporty PH
11. Planowanie dostaw:
 - 11.1. Przedstawienie informacji o terminach/ilościach przyszłych dostaw.
12. Wysyłki i paczki:
 - 12.1. Lista dokumentów wysyłek i paczek.

Projektowany system B2B stanowi rozwiązanie informatyczne, które wspiera wymienione procesy. Przyjęto, że jego budowa powinna zapewniać możliwość integracji z systemami ERP kluczowych na rynku producentów. Z platformy internetowej będą mogli korzystać użytkownicy zdefiniowani w bazie danych, dla których zostały ustawione odpowiednie parametry przez administratora systemu. Zakres uprawnień poszczególnych użytkowników jest konfigurowalny. Dostęp do platformy za pomocą przeglądarki jest możliwy po podaniu unikatowych dla każdego użytkownika ID oraz hasła.

4. Podsumowanie

Przedsiębiorstwa współpracujące na platformie B2B są zorientowane na osiągnięcie korzyści, które w sposób bezpośredni lub pośredni przełożą się na ich kondycję ekonomiczną.

Dlatego konstrukcja systemu B2B w efekcie końcowym powinna spełniać główne założenia, m.in.

- podnieść poziom obsługi partnerów biznesowych poprzez oferowanie narzędzi ułatwiających współpracę,
- zautomatyzować kontrolę standardów wymiany handlowej,
- zapewnić indywidualne dostosowanie cenników uwzględniających indywidualne warunki rabatowania,

- obniżyć koszty przeprowadzonych transakcji oraz podnieść ich jakość (brak barier geograficznych, ograniczenie ilości osób biorących udział w obsłudze transakcji, eliminowanie dodatkowych aplikacji wspomagających procesy biznesowe),
- ułatwić planowanie budżetu oraz kontrolę wydatków (np. przyznane limity cenowe i prezentacja ich bieżącego wykorzystania),
- umożliwić śledzenie realizacji procesów zakupowych i logistycznych (statusowanie zamówień, powiadomienia o realizacji itp.),
- umożliwić stały dostęp zarejestrowanych użytkowników (brak logowania anonimowego w systemach B2B),
- udostępniać przez całą dobę, 7 dni w tygodniu informacje o aktualnej ofercie produktowej.

Ze względu na to, że projektowany system B2B ma służyć do obsługi dużych wolumenów danych, ważną cechą jest zapewnienie odpowiedniej wydajności i skalowalności rozwiązania w celu ergonomicznej pracy wielodostępowej. Oznacza to, że system będzie umożliwiał podobne warunki pracy przy dostępie zarówno jednoosobowym, jak i wieloosobowym. Rozwiązanie nie ogranicza liczby osób pracujących równolegle w systemie, co więcej liczba ta w domyśle pozostaje nieograniczona. System musi więc zapewnić możliwość efektywnej pracy bez względu na obciążenie poszczególnych komponentów samego rozwiązania B2B, jak też systemu ERP oraz bazy danych. Procesy zachodzące na bazie danych (analizy danych, uruchomienie wielu raportów jednocześnie, duże obciążenie systemu ERP) powinny być jak najmniej odczuwalne dla użytkowników zalogowanych do systemu. Zaprojektowane moduły i zarządzanie danymi wspieranymi przez platformę B2B zostały tak skonstruowane, aby nie doprowadzać do konfliktu dostępu ze strony kilku użytkowników.

Należy pamiętać, że wydajność systemu musi być równocześnie wspierana przez odpowiednie urządzenia sprzętowe. W tym celu system zbudowany będzie jako wielowarstwowy, gdzie w zależności od potrzeb można umieścić komponenty systemu na rozbudowanych platformach sprzętowych, zwiększając wydajność ogólną systemu (np. odseparowanie bazy danych od warstwy pośredniczącej i warstwy interfejsu poprzez umieszczenie ich na osobnych serwerach).

Jednym z ważniejszych aspektów skalowalności projektowanego systemu jest możliwość dostosowania (przez właściciela platformy B2B) własnych systemów informatycznych do systemów przedsiębiorstw, z którymi współpracuje. Oznacza to udostępnienie możliwości integracji poprzez wprowadzanie mechanizmów automatycznych importów danych, które pochodzą z systemów partnerów.

Przypisy

- ¹ Projekt jest realizowany przez firmę OPTeam SA, finansowany w ramach Regionalnego Programu Operacyjnego Województwa Lubelskiego na lata 2007–2013, Oś Priorytetowa: Przedsiębiorczość i innowacje, Działanie 1.6A: Badania i nowoczesne technologie w strategicznych dla regionu dziedzinach, Lublin 2014. W projekcie autorka pełni rolę kierownika naukowego.
- ² Badania zostały przedstawione w opracowaniu pt. „Badanie funkcjonalności modeli systemów B2B opartych na technologiach WWW oraz opracowanie metody ich standaryzacji”. Wyniki badań pierwszego etapu projektu zostały omówione w: Łobaziewicz, 2014.

Bibliografia

- Łobaziewicz, M. (2014). Platformy integracyjne oparte na modelu relacji biznesowych. W: A. Nowicki i D. Jelonek (red.), *Technologie informacyjne w kreowaniu przedsiębiorczości*. Częstochowa: Wydawnictwo Wydziału Zarządzania Politechniki Częstochowskiej.
- McLaughlin, B., Pollice, G. i West, D. (2006). *Head First Object Oriented Analysis and Design*. USA: O'Reilly Media.
- Opracowanie nowoczesnego i zaawansowanego systemu B2B opartego o technologie internetowe w wyniku prac badawczo-rozwojowych. Część 2.* (2014). Rzeszów.
- Schmuller, J. (2003). *UML dla każdego*. Gliwice: Helion.
- Siau, K. i Lee, L. (2004). Are Use Case and Class Diagrams Complementary in Requirements Analysis? An Experimental Study on Use Case and Class Diagrams in UML. *Requirements Engineering*, 9 (4).