

XV Konferencja PLOUG  
Kościelisko  
Październik 2009

# ebXML – XML w służbie handlu

Tomasz Traczyk  
Politechnika Warszawska

*ttraczyk@ia.pw.edu.pl*

**Abstrakt.** Od dawna wiadomo, że Internet jest świetnym „miejscem” do handlowania. O ile handel typu B2C (Business to Consumer) jest dość dobrze znany i nie wymaga specyficznych technologii, o tyle handel B2B (Business to Business) jest zagadnieniem znacznie ciekawszym, a mniej znanym. By internetowe handlowanie rzeczywiście usprawniało działanie firm, konieczne są takie sposoby wymiany informacji, które umożliwiają składanie zamówień, ofert, negocjacje, zawieranie kontraktów, przesyłanie dokumentów (np. faktur) itp. Te wszystkie działania potrzebują odpowiednich metod i języka wymiany informacji. Idealny do tego celu wyda-je się XML: czytelny, łatwy w przetwarzaniu, powszechnie używany, z natury dostosowany do Internetu. Nie jest jednak celowe, by firmy same definiowały odpowiednie do handlu dialekty XML i metody komunikacji, gdyż większość problemów wymiany gospodarczej jest dość typowa. Podjęto zatem próby tworzenia uniwersalnych metod komunikacji i dialektów XML, przeznaczonych do prowadzenia handlu między firmami.

Jedną z bardziej znanych propozycji tego rodzaju stanowi właśnie ebXML. Jest nowoczesną metodą elektro-nicznej wymiany dokumentów, dobrze dostosowaną do zastosowań we współczesnej gospodarce elektronicznej opartej na technologiach internetowych.

## 1. Handel elektroniczny B2B

Współczesne przedsiębiorstwa potrzebują metod umożliwiających szybkie negocjowanie kontraktów, zawieranie transakcji i wymianę dokumentów handlowych. Handel elektroniczny typu B2B (*Business to Business*) ma odpowiadać tym właśnie potrzebom. Dotyczy on transakcji zawieranych między przedsiębiorstwami oraz organizacjami i instytucjami gospodarczymi i rządowymi.

Istnieją różne modele tego handlu: oprócz typowej wymiany bilateralnej są też modele wymagające istnienia pośredników, giełdowe, aukcyjne, a nawet oparte na sieciach inteligentnych automatycznych agentów. W bardziej skomplikowanych modelach bierze się także pod uwagę różnego rodzaju ograniczenia, np. narzucane przez infrastrukturę (komunikacyjną, przesyłową itp.).

Do prowadzenia każdego rodzaju handlu elektronicznego B2B niezbędna jest możliwość skutecznego komunikowania się partnerów. Muszą zatem istnieć odpowiednie mechanizmy komunikacji oraz języki elektronicznej wymiany informacji.

### 1.1. Handel B2B a wcześniejsze technologie

Problemy występujące w handlu B2B są bardzo podobne do tych, które napotymano budując tradycyjne rozwiązania typu EDI (*Electronic Data Interchange*). Dlatego wiele doświadczeń EDI można wykorzystać, choć oczywiście szczegółowe rozwiązania techniczne tradycyjnego EDI, zwykle już bardzo przestarzałe i kosztowne, powinny być zastąpione przez rozwiązania współczesne, głównie język XML i usługi sieciowe (*Web services*). Twórcy ebXML wyraźnie jednak podkreślali, że ich celem nie jest zanegowanie dorobku EDI, lecz wykorzystanie go.

Organizacje chcące przystąpić do handlu typu B2B zwykle mają już systemy informatyczne obsługujące podstawową działalność, np. systemy typu ERP czy MRP. Włączenie się w „gospodarkę elektroniczną” oczywiście nie może wymagać wymiany takich systemów. Rozwiązania właściwe dla handlu B2B muszą zatem móc integrować się z istniejącymi systemami informatycznymi.

### 1.2. Szczegółowe wymagania dla B2B

Jak się okazuje, prowadzenie efektywnego i bezpiecznego handlu typu B2B wymaga spełnienia szeregu wymagań, z których tylko nieliczne są w miarę oczywiste, inne zaś są dość wyrafinowane.

Do prostych wymagań należą:

- umożliwienie komunikacji między partnerami, tj. stworzenie warunków technicznych dla niezawodnej wymiany komunikatów między systemami informatycznymi partnerów;
- istnienie języka wymiany danych, zrozumiałego dla systemów informatycznych wszystkich partnerów.

Jak się jednak okazuje, z tych prostych postulatów wynikają wymagania mniej oczywiste. By partnerzy mogli się skomunikować muszą przecież w ogóle wiedzieć o swoim istnieniu oraz znać udostępniane przez drugą stronę metody komunikacji, zatem potrzebne jest:

- istnienie powszechnie znanego i dostępnego rejestru potencjalnych uczestników rynku lub innej formy informowania o zainteresowaniu przystąpieniem do rynku i o parametrach komunikacji z poszczególnymi podmiotami;
- istnienie mechanizmów wyszukiwania potencjalnych partnerów handlowych w rejestrze.

Jeśli chodzi o wspólny język sprawa też nie jest prosta, chodzi bowiem nie tylko o język składniowo zrozumiały dla systemów informatycznych partnerów, ale także o zgodność co do semantyki wymienianych informacji, tj. zapewnienie że pod tymi samymi terminami wszyscy uczestnicy danego rynku rozumieją te same pojęcia. To uzyskać można zaś przez:

- istnienie powszechnie dostępnego repozytorium zawierającego terminologię, którą posługiwać się należy w dokumentach handlowych; takie repozytorium może mieć formę słownika, a w trudniejszych przypadkach może być bardziej sformalizowane, np. przybrać formę ontologii.

By partnerzy mogli prowadzić wymianę dokumentów handlowych skutkującą zawarciem transakcji handlowej, niezbędne jest uzgodnienie prowadzących do tego celu procesów biznesowych, zatem potrzebne jest:

- istnienie uznawanych przez partnerów modeli procesów biznesowych prowadzących do zawarcia transakcji handlowych; takie modele mogą być udostępnione publicznie w odpowiednim repozytorium;
- umożliwienie uzgadniania przez partnerów kontraktu opisującego warunki zawierania transakcji.

W bardziej złożonych modelach handlu może być niezbędne także:

- umożliwienie transakcji multilateralnych, w tym także możliwość uzgodnienia warunków zawierania takich transakcji;
- negocjowanie poszczególnych transakcji, także w handlu multilateralnym.

System handlu B2B może być użyteczny w rzeczywistym obrocie gospodarczym tylko wtedy, gdy zapewnia on bezpieczeństwo transakcji, do czego niezbędne jest:

- uwierzytelnianie stron transakcji, tj. możliwość pewnego stwierdzenia ich tożsamości;
- zapewnienie poufności transmisji, tj. niemożności „podśluchania” przesyłanej informacji przez stronę trzecią;
- umożliwienie wiarygodnego podpisywania dokumentów handlowych przez strony i ich przedstawicieli (z możliwością podpisywania odpowiednich fragmentów dokumentów przez różne osoby);
- zagwarantowanie integralności dokumentów, tj. możliwości weryfikacji, iż otrzymany dokument nie został zmieniony i jest identyczny z dokumentem wysłanym;
- zapewnienie takiego sposobu wymiany dokumentów, by żadna ze stron nie mogła wyprzeć się faktu wysłania lub otrzymania dokumentu;
- utrzymywanie archiwum przesyłanych dokumentów, umożliwiającego ich późniejsze odtworzenie, np. w celu rozwiązania zaistniałych sporów.

W przypadku bardziej skomplikowanych rynków niezbędne może być także istnienie:

- centralnego pośrednika (operatora rynku) lub sieci pośredników zarządzających rynkiem i/lub wypracowujących transakcje: giełdy, domu aukcyjnego, jednostki bilansującej rynek itp.;
- formalnych modeli wypracowywania transakcji, np. różnych modeli aukcji lub modeli optymalizacji prowadzącej do zbilansowania danego rynku;
- modeli infrastruktury w której odbywa się handel, np. sieci transportowej lub przesyłowej (sieć energetyczna, telekomunikacyjna itp.);

- modeli ograniczeń obowiązujących na danym rynku, wynikających z infrastruktury (np. przepustowości w sieciach) lub z natury produktu (np. niemożności magazynowania danego produktu, co prowadzi do wymagania ciągłego zbilansowania rynku – dobrym przykładem jest tu energia elektryczna).

## 2. Czym jest ebXML?

ebXML (*electronic business XML*) jest jednym z najważniejszych rozwiązań dotyczących handlu elektronicznego i elektronicznej wymiany dokumentów. Jest to standard określający sposoby prowadzenia handlu elektronicznego typu B2B oraz rozwiązania techniczne dotyczące tego handlu. Standard ten obejmuje szeroki zakres problemów: od modelowania procesów i pojęć używanych w handlu, przez mechanizmy komunikacji, języki wymiany danych, rejestry informacji, aż po zagadnienia bezpieczeństwa transakcji. Standard zawiera zbiór specyfikacji wyznaczających ramy (*framework*) dla handlu elektronicznego. Wypełnia większość wymagań omówionych wyżej, zapewniając możliwość prowadzenia typowego handlu B2B. W założeniu ma być dostosowany do potrzeb organizacji różnej wielkości na całym świecie.

Choć standard obejmuje wiele zagadnień, jest jednak modularny, a moduły są powiązane dość luźno. Zostawia też dużą swobodę co do użytych technologii komunikacyjnych, a przede wszystkim co do zawartości wymienianych informacji i dokumentów. Można go zatem uznać za rozwiązanie dość uniwersalne.

### 2.1. Historia i standaryzacja ebXML

ebXML jest wspólnym dziełem ONZ (UN/CEFACT – *United Nations Centre for Trade Facilitation and Electronic Business*) i organizacji OASIS (*Organization for the Advancement of Structured Information Standards*). W pracach nad ebXML uczestniczyła też organizacja W3C (*World Wide Web Consortium*) oraz amerykański NIST (*National Institute of Standards and Technology*). Prace nad ebXML rozpoczęły się w roku 1999, zaś pierwszą wersję propozycji standardu zatwierdzono w maju 2001 roku. Obecnie obowiązuje wersja 2 standardu [ebXML], wypracowana w roku 2002. Prowadzone są prace nad wersją 3.

Standard ten został także przyjęty przez ISO pod nazwą ISO 15000 *Electronic Business eXtensible Markup Language*. W skład tego standardu wchodzi następujące dokumenty:

- ISO 15000-1: *ebXML Collaborative Partner Profile Agreement*,
- ISO 15000-2: *ebXML Messaging Service Specification*,
- ISO 15000-3: *ebXML Registry Information Model*,
- ISO 15000-4: *ebXML Registry Services Specification*,
- ISO 15000-5: *ebXML Core Components Technical Specification*.

## 3. Architektura ebXML

Architektura ebXML składa się właściwie z dwóch części [Rawlins01]: architektury procesowej, opisującej sposób prowadzenia analizy i modelowania oraz zapisu wyników tego modelowania i architektury produktowej, opisującej infrastrukturę techniczną, komunikację, formaty komunikatów itp.

### 3.1. Architektura procesowa

Architektura procesowa określa metody i środki których należy użyć do modelowania procesów współdziałania stron i do zapisu wyników tego modelowania. Obejmuje ona m. in.:

- *Business Process Analysis Worksheet & Guidelines* – wskazania i pomoce do modelowania procesów i tworzenia zapisu modeli w specjalnym języku BPSS (patrz punkt 0);
- *Catalog of Common Business Processes* – predefiniowaną listę i opisy typowych procesów;
- *E-Commerce Patterns* – kompletne wzorce typowych kontaktów handlowych, opisujące cele, uczestników i sposoby interakcji;
- *Core Components* – katalog podstawowych pojęć i ich znaczeń w różnych kontekstach (patrz punkt 0).

### 3.2. Architektura produktowa

Architektura produktowa definiuje techniczną infrastrukturę ebXML. Zawiera m. in.:

- *Messaging Service* – usługi zapewniające wymianę komunikatów między współdziałającymi stronami;
- *Registry/repository* – bazę danych, zawierająca informacje niezbędne do tworzenia wzajemnie zrozumiałych modeli oraz opisujące konkretnych uczestników rynku, schematy dokumentów itp.;
- *Trading Partner Information* – struktury pozwalające tworzyć sformalizowane opisy uczestników rynku (CPP – patrz punkt 0) oraz specyfikacje współdziałania między konkretnymi partnerami (CPA – patrz punkt 0);
- *Business Process Specification Schema* (BPSS) – język opisu procesów biznesowych i współdziałania partnerów.

### 3.3. Rejestr/repozytorium ebXML

Repozytorium ebXML jest publicznie dostępną składnicą informacji, zawierającą dane potrzebne do prowadzenia handlu. W repozytorium można przechowywać wszelkie informacje, nie tylko te opisane w standardzie, i nie tylko w formacie XML. Struktura danych repozytorium oraz sposób jego implementacji nie są określone przez standard; zawiera on jedynie wskazania co do minimalnej zawartości informacyjnej i ogólnej struktury danych widzianej przez API rejestru (określa to tzw. RIM – *Registry Information Model*).

Rejestr ebXML udostępnia zawartość repozytorium oraz umożliwia efektywne wyszukiwanie informacji. Rejestr udostępnia m. in.:

- metamodelo procesów biznesowych i informacji,
- *Core Library*, czyli zbiór standardowych składowych z których buduje się opisy konkretnych złożonych procesów biznesowych; ten zbiór jest predefiniowany przez standard i zawiera typowe proste procesy i podstawowe pojęcia;
- profile CPP zarejestrowanych podmiotów.

Dostęp do rejestru odbywa się przez specjalne API, zgodne z normą ISO/IEC 11179, z którego aplikacje-klienci rejestru mogą korzystać za pośrednictwem protokołu SOAP. Rejestry ebXML nie wykorzystują typowych dla *Web Services* technologii WSDL ani UDDI, ale UDDI można użyć do odnajdowania odpowiedniego rejestru ebXML.

Rejestr umożliwia nie tylko wyszukiwanie informacji za pomocą odpowiednich komunikatów ebXML lub zapytań SQL, ale także manipulowanie nią (przez tzw. OMS – *Object Management Service*) oraz pozwala klasyfikować informację (system klasyfikatorów ma formę drzewa) i użyć tej klasyfikacji do wyszukiwania.

### 3.4. Wymiana komunikatów

Komunikacja w ebXML opiera się na asynchronicznym przesyłaniu komunikatów. Po wysłaniu komunikatu nadawca nie musi zatem czekać na odpowiedź odbiorcy, choć oczywiście odbiorca może wysłać – także asynchronicznie – potwierdzenie odbioru. Systemy informatyczne nadawcy i odbiorcy są zatem powiązane luźno, a płynność ich działania nie zależy od sprawności systemów drugiej strony.

Komunikat ebXML składa się z nagłówka (*header*) zawierającego informacje kontrolne oraz właściwej zawartości (*payload*), np. dokumentów handlowych. Nagłówek zawiera zwykle:

- unikalny identyfikator komunikatu,
- informacje o nadawcy i odbiorcy,
- łączniki do powiązanych komunikatów (*conversation id*),
- podpis cyfrowy,
- oznaczenie czy duplikaty komunikatu mają być ignorowane,
- oznaczenie czy wymagane jest potwierdzenie otrzymania komunikatu.

Specyfikacja *ebXML Messaging Service Specification* (ebMS) szczegółowo określa sposób wymiany informacji i definiuje budowę komunikatów. Stanowi rozszerzenie specyfikacji *SOAP with Attachments*, która z kolei korzysta ze znanych mechanizmów *MIME multipart/related media type*. Przesyłany pakiet MIME składa się z kilku części, z których pierwszą jest dokument SOAP, zawierający nagłówek komunikatu, a kolejne części to *payload*, czyli właściwa zawartość komunikatu.

Zastosowanie standardu MIME powoduje, że specyfikacja ebMS jest niezależna od metod i protokołów komunikacji. Oczywiście najbardziej odpowiedni wydaje się protokół SOAP na bazie HTTP, ale możliwe jest także użycie do przesyłania komunikatów np. zwykłej poczty elektronicznej (protokołu SMTP).

### 3.5. Bezpieczeństwo w ebXML

ebXML nie narzuca konkretnych rozwiązań w dziedzinie bezpieczeństwa, zawiera jednak pewne wskazania, nawiązujące do wymagań opisanych w punkcie 0. Bezpieczeństwo komunikacji powinno być zapewniane przez konkretne aplikacje, ale wskaźniki, które powinny być spełnione, są opisane w standardzie.

Aplikacja realizująca komunikację ebXML powinna wypełniać trzy zadania związane z bezpieczeństwem:

- uwierzytelnianie (*authentication*), czyli weryfikację właściwego pochodzenia komunikatu,
- autoryzację, czyli ustalenie operacji, jakie mogą być wykonane na podstawie komunikatu,
- szyfrowanie, czyli zapewnienie poufności i/lub integralności informacji.

Specyfikacja zaleca użycie otwartych standardów. Do podpisywania komunikatów użyć można XML Signature (por. [Traczyk03]), co ma tę zaletę, że umożliwia opatrzenie osobnymi podpisami poszczególnych części dokumentu. Zawartość komunikatów może być szyfrowana za pomocą takich metod jak S/MIME czy PGP/MIME. Można także oczywiście prowadzić całą transmisję

danych za pomocą bezpiecznych protokołów, np. SSL. Jeśli do komunikacji używany jest protokół SOAP, to właściwym rozwiązaniem może być użycie standardu SAML (*Security Assertion Markup Language*) przez rozszerzenie koperty SOAP; ten standard jest rozwijany – podobnie jak ebXML – przez organizację OASIS.

## 4. Wykorzystanie ebXML

Zaleca się poprzedzenie implementacji bazującego na ebXML systemu wymiany dokumentów analizą. Wyniki tego modelowania są zapisywane w sposób sformalizowany i mogą stać się podstawą do automatycznej wymiany informacji.

### 4.1. Modelowanie na potrzeby ebXML

Podstawowym modelem analitycznym wykorzystywanym w ebXML jest model procesów biznesowych, a uzupełnia go model komponentów – pojęć i terminów. Wyniki modelowania procesów mogą być umieszczone w rejestrze ebXML i wykorzystane do tworzenia profili biznesowych partnerów.

#### 4.1.1. Model procesów i język BPSS

Model ten opisuje współdziałanie partnerów jako zbiór tzw. współprac (*collaborations*) realizujących określone cele biznesowe, składających się z kroków zwanych transakcjami (*transactions*), które zawierają przepływy dokumentów biznesowych.

Model procesów dla ebXML powinien być zapisany w języku BPSS (*Business Process Specification Schema*), który jest oczywiście określonym przez standard dialektem XML. Modelowania nie zaczyna się od zera, lecz korzysta z predefiniowanych i udostępnionych w rejestrze ebXML procesów bazowych – elementarnych typowych procesów, których można użyć wprost lub budować z ich pomocą bardziej złożone procesy. Model w BPSS zawiera:

- opis zakresu współpracy,
- scenariusz działań,
- opisy transakcji,
- opisy wymiany dokumentów elektronicznych.

Zalecaną metodyką modelowania jest UMM (*UN/CEFACT Modeling Methodology*), bazująca na UML. By jednak uzyskać model w języku BPSS nie trzeba koniecznie tworzyć pełnego modelu w UML. Do tworzenia zapisów w BPSS wykorzystuje się specjalne edytory procesów, pozwalające wypełnić odpowiedni formularz (*Business Process Analysis Worksheet*) i na jego podstawie tworzące dokument BPSS.

#### 4.1.2. Model pojęć

Dla osiągnięcia celów biznesowych współdziałania partnerów i uniknięcia nieporozumień jest niezbędne, by partnerzy operowali tym samym zestawem terminów i rozumieli pod nimi te same pojęcia. W szczególności istotna jest zgodna interpretacja terminów użytych w wymienianych dokumentach elektronicznych.

Dlatego oprócz modelu procesów konieczny jest jeszcze model danych: terminów, odpowiadających im pojęć i opisujących je atrybutów. Podobnie jak w przypadku procesów, modelu tego nie buduje się od zera, lecz korzysta z tzw. *Core Components* – pojęć bazowych, predefiniowanych i udostępnianych przez rejestr ebXML. W przypadku typowego handlu zestaw predefiniowanych pojęć może być wystarczający; jeśli nie jest, trzeba go rozbudować. Nazewnictwo pojęć w modelu jest ściśle sprecyzowane: określają je tzw. *Naming Conventions*, oparte na normie ISO 11179.

Pojęcia opisują obiekty, które mogą mieć określone cechy. Dlatego pojęciom-komponentom można przypisać pewne atrybuty, można też budować komponenty złożone, zawierające inne komponenty.

Zauważono, że te same terminy miewają różne znaczenie w zależności od okoliczności użycia. Dlatego standard posługuje się tzw. kontekstem (*context*) użycia danego terminu. Tzw. *context drivers* opisują ów kontekst niezależnie w wielu wymiarach dotyczących rodzaju procesu biznesowego, zagadnień geopolitycznych, rodzaju produktu, ról partnerów, gałęzi gospodarki itp. Standard określa predefiniowany katalog owych *context drivers*, który może być rozszerzony stosownie do potrzeb danego segmentu rynku. Definiując w sposób formalny terminy, pojęcia i ich atrybuty można korzystać z rodzaju wyrażen warunkowych, uzależniających poszczególne części definicji od kontekstu.

#### 4.1.3. CPP (*Collaborative Partner Profile*)

CPP jest dokumentem zawierającym opis możliwości danego uczestnika rynku. Jest on sformułowany w zdefiniowanym przez standard dialekcie XML i powinien być opublikowany w rejestrze ebXML. CPP można traktować jako rodzaj ogłoszenia uczestnika rynku na temat jego możliwości co do prowadzenia elektronicznej wymiany dokumentów i elektronicznego zawierania transakcji handlowych.

CPP składa się z kilku warstw (por. [IliM07]):

- specyfikacji procesu biznesowego (*process specification layer*) – określającą transakcje,
- specyfikacji przesyłania wiadomości (*delivery channel*) – określającą sposoby dostarczania poszczególnych wiadomości,
- warstwy wymiany dokumentów (*document-exchange layer*) – definiującej dokumenty, sposób ich przetwarzania, szyfrowanie i podpisywanie dokumentów itp.,
- warstwy transportowej (*transport layer*) – określającej protokół transportowy i adresy.

#### 4.1.4. CPA (*Collaboration Partner Agreement*)

CPA określa szczegóły elektronicznej interakcji między konkretnymi partnerami. Jest dokumentem XML wypracowywanym na podstawie CPP partnerów chcących nawiązać współpracę. Choć teoretycznie możliwe jest automatyczne tworzenie CPA, zwykle jest to jednak proces z udziałem ludzi (i tradycyjnych negocjacji), ponieważ jego wynik ma charakter umowy między stronami.

CPA stanowi podstawę do prowadzenia elektronicznej wymiany dokumentów oraz zawierania transakcji handlowych między konkretnymi stronami. Zawiera m. in.:

- informacje na temat samego CPA, np. czas jego obowiązywania,
- informacje identyfikacyjne i role stron w wymianie handlowej,
- sposoby komunikacji (protokoły, parametry transmisji itp.),
- adresy sieciowe, nazwy usług (*services*) i akcji itp., parametry itp.,
- informacje dotyczące bezpieczeństwa (wymagane podpisy elektroniczne itp.),
- definicje transakcji jakie mogą być wykonywane,
- określenie wymagań co do potwierdzania otrzymania komunikatów oraz reguły ustalające reakcje na sytuacje nieprawidłowe, np. na brak potwierdzenia, zduplikowane przesłanie dokumentu itp.,
- komentarze tekstowe.

## 4.2. Co trzeba zrobić by użyć ebXML?

Firma, która chce skorzystać z ebXML by prowadzić transakcje elektroniczne, wykonać powinna następujące kroki.

1. Pozyskać i przyswoić specyfikacje ebXML, a następnie odnaleźć w Internecie rejestr ebXML i pobrać z niego informacje potrzebne do zbudowania modelu swojego uczestnictwa w handlu, m. in. *Core Library*.
2. Stworzyć model niezbędnych procesów biznesowych i na jego podstawie zbudować lokalną aplikację ebXML. Aplikacja ta może być tworzona od podstaw lub zakupiona i skonfigurowana. Najczęściej niezbędne będzie zintegrowanie jej z już istniejącymi systemami informatycznymi wspomagającymi działalność firmy.
3. Zbudować swój profil CPP i zarejestrować go w rejestrze ebXML. W ten sposób firma staje się dostępna dla partnerów chcących z nią handlować za pomocą ebXML.
4. Wyszukać w rejestrze ebXML potencjalnych partnerów handlowych.
5. Korzystając z CPP potencjalnego partnera zbudować propozycję CPA, opisującą współpracę z nim, po czym przeprowadzić negocjacje w celu uzgodnienia owego CPA.

Po obustronnym zaakceptowaniu CPA można przystąpić do handlu.

## 4.3. Handel za pomocą ebXML

Uzgodnienie profilu współpracy umożliwia prowadzenie elektronicznej wymiany dokumentów i zawieranie transakcji handlowych. Interakcja stron musi być jednak odpowiednio zorganizowana. Składa się ona – podobnie jak w opisanym w części 0 modelu procesów – z tzw. współprac (*collaborations*), realizujących ściśle określone zadania biznesowe i składających się z transakcji.

### 4.3.1. Transakcje

Atomowymi elementami interakcji z użyciem ebXML są tzw. transakcje (*transactions*), które mają znaczenie dość podobne jak w systemach OLTP – słowo „transakcja” ma tu zatem znaczenie bardziej informatyczne niż handlowe. Każda transakcja może się zakończyć sukcesem lub porażką. Transakcja zakończona sukcesem może mieć moc wiążącą prawnie – skutkuje więc także transakcją w sensie handlowym. Transakcja zakończona porażką jest uznawana za niebyłą, a jej wszystkie skutki są wycofywane.

W ramach transakcji określony jest tzw. przepływ dokumentów biznesowych (*business document flow*). Jest to konkretny scenariusz wymiany informacji: dokumentów oraz tzw. sygnałów potwierdzających pewne zdarzenia. Struktura wymienianych dokumentów jest oczywiście zróżnicowana – dostosowana do konkretnej sytuacji; struktura sygnałów jest zaś sztywna. Typowy przepływ z potwierzzeniami (w ramach pojedynczej transakcji) może np. wyglądać tak:

- strona inicjująca wysyła do drugiej strony dokument zawierający pewne żądanie;
- druga strona odsyła dwa sygnały: potwierdzenie otrzymania dokumentu oraz sygnał akceptacji dokumentu;
- następnie druga strona przesyła dokument zawierający odpowiedź na żądanie;
- strona inicjująca odsyła sygnał potwierdzający otrzymanie odpowiedzi.

### 4.3.2. Współprace

Transakcje składają się na współpracę (*collaborations*), które nieco przypominają sesje z systemów OLTP. Każda współpraca zawiera pewną liczbę transakcji. W ramach współpracy określone są role (*roles*), które mogą pełnić współdziałające strony, oraz działania (*activities*),

które każda z ról może wykonywać w ramach danej transakcji; działania te podzielone są na żądania (*requesting business activity*) i odpowiedzi (*responding business activity*).

Współprace mogą być bilateralne (*binary collaborations*) lub multilateralne (*multiparty collaborations*). Współprace multilateralne definiuje się przez odwołanie do działań zawartych w definicjach współprac bilateralnych określonych dla różnych par ról.

### 4.3.3. Choreografia

Działania mogą tworzyć tzw. choreografię (*choreography*), która jest rodzajem diagramu stanów: określa stany (*business states*) w jakich może się znajdować współpraca oraz przejścia między nimi (*transitions*). Możliwe są następujące rodzaje stanów: start współpracy, stany końcowe: sukces lub porażka, rozwidlenie (*fork*) sterowane warunkiem oraz połączenie (*join*), polegające na oczekiwaniu na łączne zakończenie kilku działań.

Współprace i transakcje mogą być szczegółowo opisane w języku BPSS, można się do nich odwołać w dokumentach CPP oraz CPA.

## 5. Pozycja ebXML

ebXML jest już standardem dojrzałym. Opiera się na otwartych technologiach, nie będących własnością żadnej firmy (*non-proprietary*). Stanowi standard ISO i jest nadal rozwijany przez UN/CEFACT i OASIS. Doczekał się szeregu implementacji, m. in. w ramach tak znanych rozwiązań jak BEA Web Logic, Sun ONE Integration Server (oba zakupione przez Oracle!), czy IBM Websphere.

Choć istnieje kilka konkurencyjnych rozwiązań, wspieranych przez potężne firmy informatyczne, otwartość i poparcie ważnych organizacji międzynarodowych daje standardowi ebXML bardzo silną pozycję.

### 5.1. Zalety i wady ebXML

Do podstawowych zalet ebXML należą niewątpliwie:

- otwartość standardu i oparcie na otwartych technologiach – co daje możliwość użycia ebXML bez ponoszenia dodatkowych kosztów licencyjnych oraz bez uzależniania się od konkretnego dostawcy;
- przyjęcie XML jako podstawy wszystkich dialektów używanych w ebXML – daje to możliwość zastosowania standardowego darmowego oprogramowania do translacji, generowania czy parsowania dokumentów ebXML, dzięki czemu koszty budowy własnych aplikacji ebXML mogą być stosunkowo niskie;
- wsparcie przez ważne organizacje międzynarodowe i standaryzacyjne – co daje gwarancję, że rozwiązanie to nie zniknie z rynku w niedalekiej przyszłości;
- szeroki zakres proponowanych rozwiązań, nie ograniczający się tylko do protokołów czy formatów komunikacji – pozwala to skorzystać z gotowych rozwiązań licznych nietrywialnych problemów związanych z B2B (patrz punkt 0);
- dojrzałość standardu – prace trwają już 10 lat, a w użyciu jest wersja 2 standardu.

Niestety ebXML ma też szereg wad:

- stosunkowo duża złożoność – ze względu na kompleksowe podejście do problemu standard zawiera wiele dokumentów i obejmuje bardzo szeroki zakres zagadnień, wymagających wysokich i zróżnicowanych kompetencji;

- ciągle niezbyt dużą popularność – mimo istnienia dość licznych gotowych narzędzi ebXML nie jest ciągle stosowany powszechnie;
- znaczną cenę gotowego oprogramowania – produkty mające dobrą pozycję rynkową i znanych producentów są kosztowne (np. *Web Logic ebXML adapter* kosztuje obecnie ponad 11 tys. dolarów na procesor, a oczywiście do tego dochodzi koszt samego serwera Web Logic, bazy danych itd.).

Mimo iż jednym z założeń projektu ebXML była jego dostępność dla organizacji wszelkiej wielkości na całym świecie, wydaje się że obecnie nie jest to jeszcze technologia łatwo dostępna dla sektora MSP ani dla krajów rozwijających się: gotowe rozwiązania są za drogie, a budowa własnych rozwiązań jest technologicznie dość trudna i wymaga szerokich kompetencji, a więc zatrudnienia drogich specjalistów.

Standard, mimo swej złożoności, nie rozwiązuje też wszystkich problemów występujących w bardziej złożonych modelach handlu, np. nie zawiera gotowego wsparcia dla handlu wielotowarowego (sprzedaży wiązanej), choć stworzenie w ramach ebXML odpowiednich rozwiązań jest możliwe. Nie ma też mechanizmów bezpiecznego zatwierdzania transakcji multilateralnych (por. [Pałka09]). Użycie ebXML w sieci automatycznych agentów wydaje się dość kłopotliwe.

Twórcy standardu zauważyli natomiast potrzebę opisu złożonej semantyki pojęć używanych w trudniejszych dziedzinach handlu, gdy możliwości związane z *context drivers* (patrz punkt 0) i klasyfikacją zawartości rejestru (patrz punkt 0) mogą nie wystarczać. Wychodzi temu naprzeciw propozycja [RPOWL] opisująca sposób wykorzystania języka OWL (*Web Ontology Language*) do opisu pojęć zawartych w rejestrze ebXML.

## 5.2. ebXML a inne podobne standardy

Głównymi konkurentami ebXML są standardy BizTalk [BizTalk] i RosettaNet [Rosetta]. Pierwsze z tych rozwiązań jest jednak własnością pojedynczej firmy (Microsoft) i związane jest ze środowiskiem Windows, nie ma zatem zalet otwartego standardu. Drugie jest dziełem firm przemysłu elektronicznego, bazującym na otwartych technologiach, ale nastawionym głównie na wspieranie łańcucha dostaw i dystrybucji produktów przemysłu IT, co może utrudniać rozpowszechnienie użycia tego rozwiązania. Możliwe jest współdziałanie (*interoperability*) obu tych standardów z ebXML.

## 6. Podsumowanie

W referacie przedstawiono wymagania jakie stoją przed systemami handlu typu B2B oraz pokazano jak standard ebXML realizuje te wymagania. Jak się okazuje, twórcy standardu wzięli pod uwagę większość zagadnień (pominęli tylko takie, które występują wyłącznie w najbardziej skomplikowanych modelach handlu) i zaproponowali dla nich konkretne rozwiązania, bazujące na języku XML. Tak szerokie podejście do problemu spowodowało jednak dość znaczną złożoność standardu, który zawiera cały szereg specyfikacji i odwołuje się do licznych technologii.

Powstało rozwiązanie dość uniwersalne, dające sporą swobodę w wyborze technologii komunikacyjnych, struktury dokumentów itd. Poparcie ze strony ważnych organizacji międzynarodowych i ISO zapewniło standardowi ebXML mocną i stabilną pozycję wśród propozycji systemów elektronicznej wymiany dokumentów. Gotowe oprogramowanie umożliwiające użycie ebXML proponuje wiele firm, wśród nich Oracle.

Mimo licznych zalet upowszechnienie ebXML przebiega dość wolno. Wydaje się jednak, że wśród istniejących propozycji standardów wymiany informacji typu B2B właśnie ebXML jest wyborem najbezpieczniejszym.

## Bibliografia

- [BizTalk] Microsoft BizTalk Server, <http://www.microsoft.com/biztalk/>
- [Chiu02] Chiu E.: ebXML Simplified, Wiley 2002.
- [ebXML] ebXML – Enabling a global electronic market, <http://www.ebxml.org/>
- [IliM07] Elektroniczna gospodarka w Polsce – Raport 2004, cz. VII Standardy, Instytut Logistyki i Magazynowania, 2007, [http://www.e-fakty.pl/index.php?option=com\\_content&task=view&id=4948&Itemid=0](http://www.e-fakty.pl/index.php?option=com_content&task=view&id=4948&Itemid=0)
- [Irani01] Irani R.: An introduction to ebXML – Collaborative Electronic Business is here to stay, 2001, <http://www.webservicesarchitect.com/content/articles/irani02.asp>
- [Mertz01] Mertz D.: Understanding ebXML, 2001, <http://www.ibm.com/developerworks/xml/library/x-ebxml/>
- [Pałka09] Pałka P., Kaleta M., Toczyłowski E., Traczyk T.: Wykorzystanie standardu FIPA na potrzeby M<sup>3</sup> – otwartego modelu wymiany wielotowarowej, Studia Informatica vol. 30 no 2B (84), ss. 127-140, Gliwice 2009.
- [Rawlins01] ebXML - A Critical Analysis, Rawlins EC Consulting, 2001, <http://www.rawlinsecconsulting.com/ebXML/>
- [Rosetta] RosettaNet, <http://www.rosettanet.org/>
- [RPOWL] ebXML Registry Profile for Web Ontology Language (OWL) Version 1.5, OASIS 2006.
- [Traczyk03] Traczyk T.: XML – stan obecny i trendy rozwojowe. Materiały IX Konferencji użytkowników i developerów Oracle. Kościelisko, październik 2003.
- [Webber04] Webber D.: ebXML Forum's Software Resources and Products Guide, 2004, [http://www.ebxmlforum.net/articles/ebfor\\_SoftwareProducts.html](http://www.ebxmlforum.net/articles/ebfor_SoftwareProducts.html)