# 37. Modele i technologie transakcji w środowisku usług webowych

Rozproszony charakter architektury zorientowanej na usługi oraz preferowane luźne powiązania pomiędzy elementami składowymi architektury sprawiają że zapewnienie niezawodności oraz dostarczenie mechanizmów umożliwiających transakcyjne wykonywanie pewnych zadań staje się nietrywialnie. Trudność tego zadania jest spowodowana głównie tym, że każdy z elementów infrastruktury usługowej może ulec awarii niezależnie, może to być związane zarówno z awarią sprzętu, takiego jak np. serwerów hostujących usługi lub routerów i switchy działających w warstwie sieciowej, jak i z awarią oprogramowania działającego w ramach serwerów lub samych usług. Rozwiązanie to ma swoje plusy, ponieważ awaria jednego elementu w infrastrukturze nie powoduje awarii całego systemu. Niestety, wiąże się to również z tym, że problemy związane z usługami webowymi możemy wykryć dopiero w momencie próby wywołania jednej z ich operacji. Problem ten staje się wyjątkowo dotkliwy, gdy wymagana przez użytkownika funkcjonalność jest realizowana poprzez usługę złożoną opisaną w postaci procesu składającego się z szeregu usług atomowych. W tej sytuacji dużym ryzykiem jest sytuacja, w której część procesu zostanie wykonana, podczas gdy część w ogóle nie zostanie uruchomiona ze względu na awarię jednej z usług atomowych.

Świadomi tego ryzyka przedstawiciele IBM, BEA Systems, Microsoft, Arjuna, Hitachi oraz IONA postanowili opracować model pozwalający na zapewnienie transakcyjnego wykonywania usług, dzięki czemu możliwa byłaby kompensacja zmian wprowadzonych przez działające usługi w przypadku krytycznego przerwania wykonywanego procesu w postaci usługi złożonej.

Przed przejściem do analizy zaproponowanego modelu transakcyjnego, warto zwrócić uwagę jak transakcje są realizowane w tradycyjnych systemach informatycznych. Modele transakcyjne powszechnie wykorzystywane w systemach informatycznych charakteryzują się cechami opisanymi poprzez akronim ACID, który można rozpisać na następujące cechy:

* ***Atomicity (atomowość)***– cecha ta oznacza, że transakcja zostanie wykonana pomyślnie w całości lub w przypadku wystąpienia awarii wszystkie wprowadzone w ramach niej zmiany zostaną cofnięte.
* ***Consistency (spójność)*** *–* cecha ta oznacza, że po wykonaniu transakcji system będzie spójny, czyli nie zostaną naruszone żadne zasady integralności.
* ***Isolation (izolacja)*** *–* cecha ta oznacza, że jeżeli dwie lub więcej transakcji wykonuje się współbieżnie, to zmiany wprowadzane każdą z nich nie są widoczne dla pozostałych. Co więcej transakcje w takiej sytuacji wyglądają jakby były realizowane sekwencyjnie, nawet jeżeli były realizowane współbieżnie. Jest to uzyskiwane zazwyczaj poprzez blokowanie zasobów, dzięki czemu nie dochodzi do sytuacji konfliktowych.
* ***Durability (trwałość) -***  cecha ta oznacza, że system potrafi uruchomić się i udostępnić spójne, nienaruszone i aktualne dane zapisane w ramach zatwierdzonych transakcji, na przykład po nagłej awarii zasilania

Bezpośrednie zastosowanie takiego podejścia w ramach systemów usługowych nie jest najlepszym rozwiązaniem. W typowych systemach informatycznych pewne złożone funkcjonalności mogą być w łatwy sposób przedstawione w ramach transakcji. Tradycyjne transakcje bardzo dobrze sprawdzają się do wykonywania krótkotrwałych zadań, które w stabilny sposób zmieniają stan systemu. Nie nadają się one jednak do wykorzystania w strukturalizowanych, długo działających funkcjonalnościach, które mogą trwać minuty, godziny, a czasami nawet dni. W takiej sytuacji tradycyjne transakcje mogą uniemożliwić zapewnienie jakiejkolwiek współbieżności w systemie ze względu na długotrwałe blokady zasobów. Poza tym, wystąpienie awarii w trakcie wykonywanie takiej transakcji wiąże się z dużymi kosztami spowodowanymi bezcelowym przetwarzaniem i wykorzystywaniem zasobów oraz uniemożliwieniem wykorzystania tej funkcjonalności przez kogoś innego.

Biorąc pod uwagę to, że przemysł informatyczny zmierza coraz bardziej w kierunku luźnych, gruboziarnistych powiązań B2B (business to business) realizowanych poprzez usługi webowe, staje się jasne, że tradycyjna forma transakcji ACID nie pasuje w pełni do oprogramowania działającego w architekturze SOA. Wynika to przede wszystkim z tego, że transakcje oparte na usługach trwają zazwyczaj dłużej, wymagaj ą aby sposób potwierdzenia realizacji transakcji był negocjowany w trakcie wykonania oraz żeby izolacja w ramach wykonywanych transakcji była luźniejsza.

Ze względu na to wspomniane powyżej firmy wraz z grupą OASIS zaproponowały specyfikacje modeli transakcyjnych, WS-Transaction, które mogłyby zostać wykorzystane w architekturze usługowej i które adresują wspomniane powyżej problemy. WS-Transaction składa się z trzech głównych elementów: *WS-Coordination*, *WS-AtomicTransaction* oraz *Ws-BusinessActivity*. *WS-Coordination* opisuje sposób koordynacji pracy usług wykonywanych w ramach transakcji. Pozostałe dwa elementy definiują sposób realizacji koordynacji:

* *WS-AtomicTransaction* - opisuje standardowe transakcje realizowane zgodnie z cechami ACID
* *WS-BussinessActiviety*- opisuje transakcje biznesowe adresujące problem długo działających usług wymagających luźniejszej izolacji

Specyfikacje te zostały bardziej szczegółowo opisane poniżej. Wykorzystywane w opracowaniu pojęcie aktywności może definiować zarówno pojedynczą usługę webową jak i grupę usług realizującą pewną funkcjonalność. W ramach tego opracowania pojęcie aktywności będzie wykorzystywane jako określenie grup usług realizowanych w ramach transakcji, natomiast pojedyncze usługi będą opisywane po prostu poprzez pojęcie usługi.

## WS-Coordination

W sytuacji gdy grupa usług webowych współpracuje ze sobą w celu wykonania pewnej funkcjonalności bardzo często muszą one współdzielić pomiędzy sobą wspólny kontekst w celu jej poprawnej realizacji. W celu zapewnienia tego wspólnego kontekstu zadania wysyłane do usług webowych otrzymują wspólny identyfikator, który pozwala identyfikować to w ramach jakiego kontekstu jest wykonywane dane zadanie. Tak zdefiniowany kontekst nie tylko definiuje istnienie pewnego środowiska w ramach, którego jest wykonywana dana funkcjonalność. Umożliwia on również wprowadzenie pewnego poziomu kontroli nad tym w jaki sposób złożone zadanie jest przetwarzane.

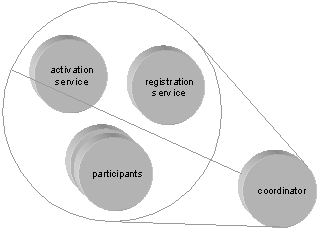
Każda aktywność wykonywana w ramach takiego środowiska wykonawczego wprowadza pewne informacje w ramach kontekstu. Coś co się dzieje lub jest wykonywane ma pewne znaczenie w trakcie okresu życia transakcji, w związku z czym opis znaczenia danej akcji może być klasyfikowany jako informacja kontekstowa. Wynika z tego prosta zależność, że im bardziej złożona jest aktywność, tym więcej informacji kontekstowych będzie ona przynosić.

Głównymi czynnikami mającymi wpływ na złożoność aktywności są:

* Liczba usług biorących udział w aktywności
* Czas trwania aktywności
* Częstotliwość z jaką może zmieniać się charakter i sposób wykonania aktywności
* Możliwość istnienia wielu instancji aktywności, które są wykonywane współbieżnie

*WS-Coordination* tworzy rozszerzalny Framework, który wprowadza pewne generyczne usługi opierające się na modelu koordynatora usług. Framework ten kontroluje trzy usługi odgrywające istotną rolę w zarządzaniu danymi kontekstowymi. Te trzy usługi to:

* **Usługa aktywacji** – usługa odpowiedzialna za tworzenie nowego kontekstu oraz powiązanie go z konkretną aktywnością.
* **Usługa rejestracji** – usługa umożliwiająca usługą biorącym udział w aktywności rejestrację w ramach stworzonego kontekstu oraz w ramach jednego ze wspieranych protokołów.
* **Usługa koordynacji** - usługa kontrolująca i weryfikująca proces wykonania transakcji

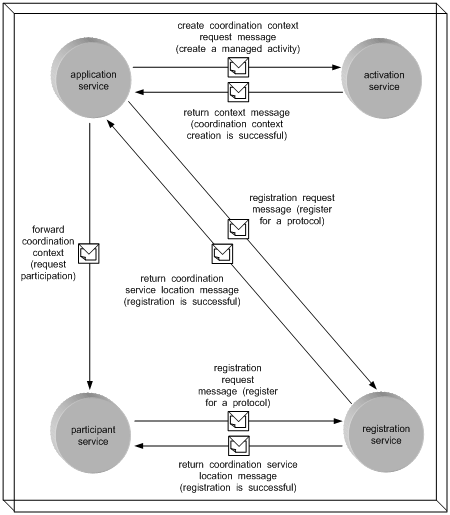
Stworzone w ten sposób środowisko można zobrazować w następujący sposób:

Każda usługa koordynacji bazuje na jakimś typie koordynacji, które specyfikuje sposób realizacji oraz logikę obsługi danej aktywności. Typy koordynacji są opisane w osobnych specyfikacjach. *WS-Coordination* jest rozszerzalne i pozwala na wykorzystywanie różnych typów koordynacji w zależności od specyficznych potrzeb. Jednakże w tym momencie wyspecyfikowane są dwa główne typy koordynacji, czyli wspomniane *WS-AtomicTransaction* i *WS-BusinessActivity*.

Poszczególne typy koordynacji zapewniają zbiór protokołów do koordynacji, które zawierają zbiór specyficznych zachowań i reguł, które powinny być realizowane w ramach danego typu koordynacji. Z tej perspektywy protokół jest zbiorem reguł nałożonych na aktywności, do których muszą się stosować wszyscy ich uczestnicy.

Inicjacja wykonania transakcyjnego realizowana w tym modelu przebiega następująco:

1. Usługa koordynacji jest tworzona gdy usługa aplikacji wykonująca transakcję zażąda stworzenia kontekstu przez usługę aktywacji.
2. Po otrzymaniu tych danych aplikacja może zaprosić inne usługi do uczestnictwa w procesie koordynacji zapewniającym transakcyjne wykonanie aktywności. W celu zaproszenia usług aplikacja przesyła informacje o kontekście otrzymane od usługi aktywacyjnej.
3. Każda usługa webowa posiadająca en kontekst może zarejestrować się w ramach wykonywanej aktywności poprzez wysłanie żądania rejestracji do usługi rejestrującej.
4. Usługa rejestrująca przekazuje nowo zarejestrowanej usłudze informacje o koordynatorze aktywności oraz przesyła temu koordynatorowi adres nowo dodanego użytkownika aktywności.
5. Po za zarejestrowaniu usługa staje się uczestnikiem wykonywanej aktywności i działa w ramach określonego typu aktywności określonego protokołem.



Po stworzeniu takiego środowiska wykonawczego dla aktywności transakcja może zostać zrealizowana zgodnie z jednym z wybranych typów koordynacji egzekwowanym przez koordynatora..

## WS-AtomicTransaction

Specyfikacja ta dostarcza standardowego mechanizmu do realizacji transakcji w środowisku usługowym zgodnego z tradycyjnym podejściem ACID. W celu wykonania takiej transakcji koordynator nadzorujący wykonanie aktywności musi wykorzystywać protokoły definiowane przez tę specyfikację, a wszystkie biorące w niej udział usługi muszą zostać zarejestrowane w ramach kontekstu, tak jak to zostało przedstawione powyżej.

Opisana specyfikacja zapewnia trzy protokoły, które są wykorzystywane w trakcie wykonania:

* *Protokół ukończenia* (Completion protocol) – protokół ten jest wykorzystywany do potwierdzanie lub anulowania i kompensacji wykonania transakcji
* *Protokół trwałych 2 fazowych potwierdzeń* (Durable 2PC) – protokół ten jest wykorzystywany przez usługi działające na trwałych danych, które mogą być utrwalane, np. w repozytoriach i bazach danych
* *Protokół ulotnych 2 fazowych potwierdzeń* (Volatile 2PC) – protokół ten jest wykorzystywany przez usługi działające na danych ulotnych, które są wykorzystywane tylko w trakcie ich realizacji

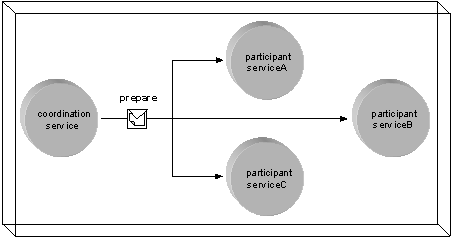
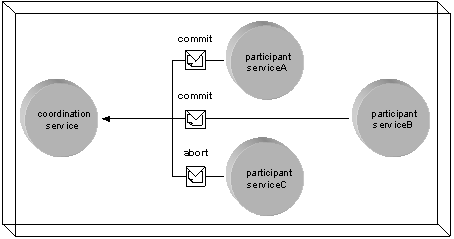
W celu realizacji tego scenariusza usługi biorące udział w komunikacji muszą implementować te protokoły.

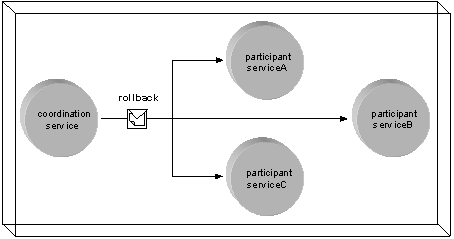
Proces wykonania transakcji odbywa się w dwóch fazach, opisanych poniżej:

*Faza przygotowawcza* (Preparation Phase) – w trakcie tej fazy wszystkie usługi są informowane przez koordynatora o tym, że mają wykonać przesłane żądanie w trybie transakcji. Następnie koordynator czeka na informację od każdej usługi o tym, czy wykonanie zakończyło się pomyślnie.

*Faza wykonania i potwierdzenia* (Commit Phase) – po zebraniu wszystkich informacji zwrotnych od usług koordynator decyduje, czy wykonanie powinno zostać zatwierdzone, czy powinno zostać skompensowane i zmiany powinny zostać cofnięte. Jeżeli wszystkie usługi potwierdziły pozytywnie wykonanie żądań transakcja jest zatwierdzana. Jeżeli, którakolwiek z usług nie potwierdziła wykonania wszystkie zmiany zostają wycofane.

Poniżej przedstawiony został schemat wykonania transakcji atomowej:



## WS-BusinessActivity

Aktywności biznesowe składają się z długo trwających transakcji w ramach, których realizacja wiąże się z wykorzystaniem wielu usług atomowych. Tego typu proces może trwać kilka godzin, dni, a czasami nawet tygodni. W trakcie tego czasu w ramach aktywności zaangażowanych jest wielu różnych uczestników oraz wykonywane są różne zadania.

Główną różnicą pomiędzy aktywnościami biznesowymi, a zwykłymi sposobem realizacji złożonych aktywności jest konieczność stosowania się przez uczestników do pewnych reguł zdefiniowanych w ramach protokółów. Różnią się one jednak od standardowych atomowych transakcji, również opartych na protokołach, sposobem obsługi wyjątków i reakcji na awarii oraz ograniczeniami wprowadzanymi przez reguły zdefiniowane w prpotokołach.

Jako przykład takiej różnicy można podać brak wsparcia dla cofania wprowadzonych zmian w przypadku niepowodzenia w realizacji aktywności. Zamiast tego aktywności biznesowe definiują „proces kompensacji”, który można traktować jako pewnego rodzaju plan awaryjny, który ma zostać wykonany w przypadku wystąpienia wyjątku spełniającego pewne warunki.

Aktywności biznesowe definiują dwa protokoły realizacji transakcji:

* *The Business Agreement With Participant Completion* – protokół ten umożliwia uczestnikowi transakcji na samodzielne stwierdzenie kiedy w całości wykonał swoją część aktywności.
* *The Business Agreement With Coordinator Completion protocol* – protokół ten zakłada, że za stwierdzenie czy dany uczestnik wykonał wszystko w ramach aktywności jest odpowiedzialny koordynator, który ma za zadanie poinformować uczestnik o tym że wykonał wszystko w ramach transakcji gdy tak się stanie.

W celu realizacji tego scenariusza koordynator musi działać w ramach tego typu koordynacji, a usługi biorące udział w komunikacji muszą implementować te protokoły.

Proces wykonania takiej aktywności można zobrazować jako szereg stanów, reprezentowanych przez stan koordynatora oraz wszystkich usług biorących udział w transakcji. Zmiana stanów, tożsama z przejściem do kolejnej fazy wykonania transakcji, występuje w momencie gdy komunikaty są przekazywane pomiędzy komunikującymi się usługami i koordynatorem.

W sytuacji wystąpienia awarii realizowane jest jedno z dwóch podejść. Albo jest uruchamiany jeden z procesów kompensacji zależnie od rodzaju awarii lub dalsze wykonanie procesu jest anulowane bez podejmowania żadnych akcji kompensacyjnych.

Jedną z istotnych cech tego typu transakcji jest luźne powiązanie uczestników z transakcją. Oznacza to, że usługa webowa uczestnicząca w transakcji nie musi czekać do zakończenia całej transakcji po wykonaniu swojej części zadania. Po zakończeniu działania usługa wysyła po prostu komunikat o opuszczeniu transakcji do koordynatora po czym odłącza się od całego procesu.

Kończąc, warto zaznaczyć, że zastosowanie transakcji w formie aktywności biznesowych nie wyklucza wykorzystania usług atomowych. Jest to wręcz wskazane, aby w ramach długo trwających aktywności biznesowych część podprocesów była realizowana w ramach transakcji atomowych.