

METODA PROJEKTOWANIA SYSTEMÓW SMART CITIES BAZOWANA NA CASE BASED REASONING

*Cezary Orłowski¹, Tomasz Sitek¹
Artur Ziółkowski¹, Paweł Kapłański¹
Aleksander Orłowski¹, Witold Pokrzywnicki¹*

Streszczenie

Celem artykułu jest prezentacja wyników badań nad opracowaniem metody projektowania systemów *Smart Cities*. Metoda ta bazuje na analizie przypadków projektowania systemów *Smart Cities* w miastach, wyborze spełniającego wymagania zdecydowanego na wdrożenie ośrodka i zastosowaniu. Do budowy proponowanej metody projektowania wykorzystano *Case Based Reasoning* oraz mechanizmy konwersji procesów i ról projektowych do procesów zunifikowanych RUP. Warunkiem stosowania proponowanej metody jest wiedza kierującego przedsięwzięciem o wysokopoziomowej architekturze systemu *Smart Cities* oraz stosowanych ramach projektowych. Autorzy posiadając wiedzę o takiej architekturze, a także o środowiskach projektowych implementujących modele KPI, proponują rozwiązanie generyczne, które mogą być wykorzystywane dla dowolnych środowisk i architektur systemów.

Słowa kluczowe: Smart Cities, Case Based Reasoning Method, decision support systems, knowledge management.

<https://doi.org/10.34808/remc.2015.02.003>

1. Wprowadzenie

Współczesne aglomeracje z uwagi na złożoność ich funkcjonowania a jednocześnie coraz większą presję społeczną na zapewnienie komfortowych warunków życia dążą do tworzenia struktur zarządczych nazywanych *Smart Cities*. Te struktury definiuje się ogólnie jako te, które wykorzystują technologie ICT dla zwiększenia wydajności własnych zasobów i jej komponentów składowych dla podniesienia komfortu życia mieszkańców (IBM, 2013a).

¹ Politechnika Gdańska, Wydział Zarządzania i Ekonomii/ Gdańsk University of Technology, Faculty of Management and Economics

Proces wykorzystania technologii ICT dla projektowania takich struktur jest złożony z powodu wielu jednocześnie wykorzystywanych technologii, długotrwały z uwagi na potrzebę iteracyjnego ich wykorzystywania oraz angażujący znaczne i z reguły rozproszone zasoby ludzkie, sprzętowe i programowe (Bhowmick, 2009; Common Alerting Protocol, 2013). Z tego też powodu projektowanie systemów informatycznych w oparciu o technologie ICT to przedsięwzięcia obarczone znacznym ryzykiem wykonawczym i wymagającym od zarządzających podejmowania trudnych decyzji dotyczących doboru omawianych uprzednio zasobów. Analiza przedsięwzięć informatycznych, których celem jest projektowanie systemów dla *Smart Cities* wskazuje na potrzebę uwzględnienia trzech krytycznych dla powodzenia tych przedsięwzięć procesów: metody projektowania, zarządzania zespołami realizującymi przyjętą metodę oraz zarządzania technologii ICT dla procesów projektowania w ramach przyjętej metody (Bhowmick, 2009; IBM, 2013a).

Zastosowanie konkretnej metody projektowania zależy w znacznym stopniu od pierwszego z omawianych procesów zarządczych – zarządzania zespołami biorącymi udział w realizacji tych przedsięwzięć. Problemem jest jednak brak stabilności tych zespołów w wieloletnim okresie trwania przedsięwzięcia, zróżnicowana wiedza pracowników urzędów nie w pełni przekonanych, że przedsięwzięcie zakończy się sukcesem. Problemem jest także potrzeba ciągłej, a nie tylko czasowej współpracy specjalistów z wielu dziedzin (pracowników urzędów i projektantów systemów).

Stosunkowo dużo można powiedzieć o zespołach projektowych biorąc pod uwagę ich dojrzałość w sensie CMMI (ang. *Capability Maturity Model Integration*), ITIL (ang. *Information Technology Infrastructure Library*) (Pastuszak i Orłowski, 2013; Pastuszak i inni, 2010) lub COBIT. W przypadku projektowania systemów *Smart Cities* pojawia się pytanie, jak liczny ma być ten zespół, kto będzie w nim reprezentować interesy klientów i na ile w długiej perspektywie czasu celowy jest pomiar tej dojrzałości. Można także przyjmować subiektywne miary oceny dojrzałości agregujące doświadczenie, wiedzę i sposób realizacji procesów wytwórczych i zarządczych przez zespół, jak zaproponowana w pracy kapsuła dojrzałości (Kowalczuk i Orłowski, 2012).

Z kolei w drugim przypadku doboru technologii informatycznych niezbędne staje się uwzględnienie w procesach projektowania zmieniających się w czasie technologii informatycznych i sposobu ich wykorzystywania dla zapewnienia procesów integracji zarówno ram projektowych (ang. *frameworks*) jak i narzędzi wykorzystywanych w procesach projektowania systemów (Smith, 2015; Snadach, 2013). O procesach zarządzania technologiami pisze się stosunkowo dużo, ale problematyka integracji technologii projektowych i zasobów informatycznych w systemach *Smart Cities* jest mało znana. Czy technologie powinny ograniczać się tylko do monitorowania zdarzeń czy też zakres tych funkcjonalności jest częściowy, obejmujący zarówno definiowanie zdarzeń i incydentów (grup zdarzeń) jak też określanie kluczowych wskaźników wydajności? Można założyć także, że zakres funkcjonalności jest pełny, co oznacza, że oprócz tych podanych uprzednio system powinien także generować powiadomienia i alerty, reguły zdarzeń dzięki budowie

standardowych procedur operacyjnych (ang. *Simple Operating Procedure*) jak też prowadzenie analiz na poziomie Business Intelligence (IBM, 2013b, 2013c).

Odpowiedź na te pytania kryje się we właściwym doborze i zastosowaniu technologii integracyjnych przy projektowaniu systemów *Smart Cities*. Jeżeli uwzględnimy zarówno projektowanie architektur SOA (ang. *Service Oriented Architecture*) wraz z szyną integracyjną, hurtownie danych, lub zwykłą integrację baz danych wtedy (wraz z informacjami o produktach) to posiadamy kompendium wiedzy o technologiach integracyjnych (Common Alerting Protocol, 2013; IBM, 2013a). Niestety wymagają one całościowych zmian w miastach i wprowadzania technologii przy pełnej znajomości potrzeb miasta w zakresie tych systemów. Dotychczasowe doświadczenia autorów wniosku wskazują, że taka wiedza jest ograniczona a przy ciągle zmieniających się uwarunkowaniach funkcjonowania miast niestwarzająca warunków do pełnego projektowania tych systemów.

W podejściu do projektowania systemów *Smart Cities* przeważa metoda top down zapożyczona z projektowania dla współczesnych miast architektury korporacyjnej w ramach której analizuje się całościowo miasto jako organizację a następnie projektuje się architekturę SOA definiując poszczególne usługi. Stosunkowo rzadko używa się metody *bottom up*. Wynika to z faktu projektowania pojedynczych systemów dla miast bez wizji wysokopoziomowej architektury (ang. *High Level Architecture*) systemów *Smart Cities*. W wielu publikacjach występują podejścia mieszane, w ramach których procesy miasta traktowane są z perspektywy architektury korporacyjnej a zdarzenia i incydenty z perspektywy silosowego widzenia wydziałów miasta. Reasumując brak jest jednolitej (referencyjnej) metody projektowania systemów *Smart Cities* a większość miast z uwagi na wieloletni horyzont czasowy projektowania mocno rozważa potrzebę projektowania tych systemów całościowo. Z tego też powodu autorzy pracy proponują metodę generyczną, która dla wymagań dowolnych miast może być stosowana jako komponent procesowy organizacji.

Dlatego też w dobie architektur opartych na mikrousługach i wsparciu procesu projektowania mikrousług ontologiami dziedzinowymi (Czarnecki i Orłowski, 2010; Kowalczyk i Orłowski, 2004, 2012) nie jest wymagana pełna wiedza o architekturze systemu dla miasta a proces projektowania realizowany jest iteracyjnie. Takie podejście nie byłoby możliwe przy wysokopoziomowym widzeniu architektury systemu bez potrzeby uwzględniania podejścia opartego na SOA.

2. Analiza podejścia *Case Based Reasoning* do projektowania procesów

Przedstawiona powyżej charakterystyka metod stosowanych do projektowania systemów *Smart Cities* wskazuje, że większość dużych miast odchodzi od podejścia bazującego na *Enterprise Architecture* oraz *SOA* (Smith, 2015; Snadach, 2013). Przyczyną takiej sytuacji są trudności ze spójnym całościowym zdefiniowaniem wymagań miast w stosunku do takich systemów. Jednocześnie prowadzone rozmowy z kierownikami projektów odpowiedzialnych za wdrożenie systemów

Smart Cities wskazują na znaczny zasób doświadczeń, które mogłyby być wykorzystywane przez innych kierowników w innych miastach realizujących podobne projekty. Taka ocena wskazuje na potrzebę analizy poszczególnych przypadków projektowania dla próby ich uogólnienia dla wygenerowania spójnej bądź bazującej na przypadkach specyficznej metody projektowania systemów *Smart Cities* dla potrzeb konkretnego miasta.

Biorąc pod uwagę powyższe uwarunkowania autorzy pracy zaproponowali rozwiązanie bazujące na analizie przypadków a jako proces przetwarzania zaproponowano *Case Based Reasoning* (CBR). Metoda ta stosowana jest powszechnie przy próbach uogólnienia przypadków lub przy doborze przypadku dla wsparcia procesów. Stosunkowo dużo prac dotyczy zastosowania metody CBR we wsparciu prac projektowych (Pokojski, 2013). Analiza tych prac nasunęła autorom pomysł na wykorzystanie tej metody w procesie budowy metody projektowania systemów *Smart Cities*.

Metoda ta opiera się na sekwencyjnym procesie analizy i doboru przypadku dla potrzeb własnego rozwiązania. Konieczny staje się formalny lub półformalny opis analizowanych przypadków oraz stosując podobną (jak w opisie przypadków) notację własnego rozwiązania. Kolejno opisy te są składowane w repozytorium przypadków. Stosuje się trzy główne metody opisu formalnego:

- reprezentacja przypadków na podstawie zbiorów cech;
- reprezentacje zorientowane obiektowo;
- reprezentacje graficzne.

Konieczne jest także określenie metody obliczania podobieństw przypadków. Wśród analizy tych metod należy wyróżnić dwie. Pierwszą z nich jest test identyczności (1) oraz test symetrycznego podobieństwa z dolną granicą (2) wyrażony następującymi zależnościami:

$$S(p_p, p_z) = \begin{cases} 1, & \text{dla } p_z = p_p \\ 0, & \text{dla } p_z \neq p_p \end{cases} \quad (1)$$

$$S(p_p, p_z) = \begin{cases} 0 & \text{dla } p_p \leq p_D \\ 1 - \frac{|p_p - p_z|}{\max(p_p, p_z) - p_D} & \text{dla } p_p > p_D \end{cases} \quad (2)$$

gdzie: p_p – wartość parametru przypadku projektowanego
 p_z – wartość parametru przypadku zrealizowanego
 p_D – dolna granica wartości parametru

Algorytm przetwarzania w ramach CBR obejmuje cztery procesy (zwane jako 4R). Pierwszym procesem analizowanego algorytmu jest wyszukiwanie najbardziej zbliżonego przypadku (ang. *retrieval*) z grupy przypadków zgromadzonych w ba-

zie wiedzy przypadków (z ang. *Case Base*) w repozytorium projektowym. Aby było to możliwe konieczne stało się zdefiniowanie metody obliczania podobieństw przypadków. Można to podobieństwo określać na podstawie testów opisanych powyżej lub proponowanych w pracach (Pokojski, 2003; Smith, 2015). Kolejny krok to zastosowanie (ang. *reuse*) najbardziej zbliżonego problemowo przypadku do analizy własnego problemu. Następnym krokiem jest weryfikacja i adaptowanie pozyskanego rozwiązania (z ang. *revision*) do bazy wiedzy przypadków. Ostatnim procesem algorytmu przetwarzania CBR jest proces zapisania nowego/zmodyfikowanego przypadku (z ang. *retainment*) w bazie wiedzy przypadków CBR (*Case Base*).

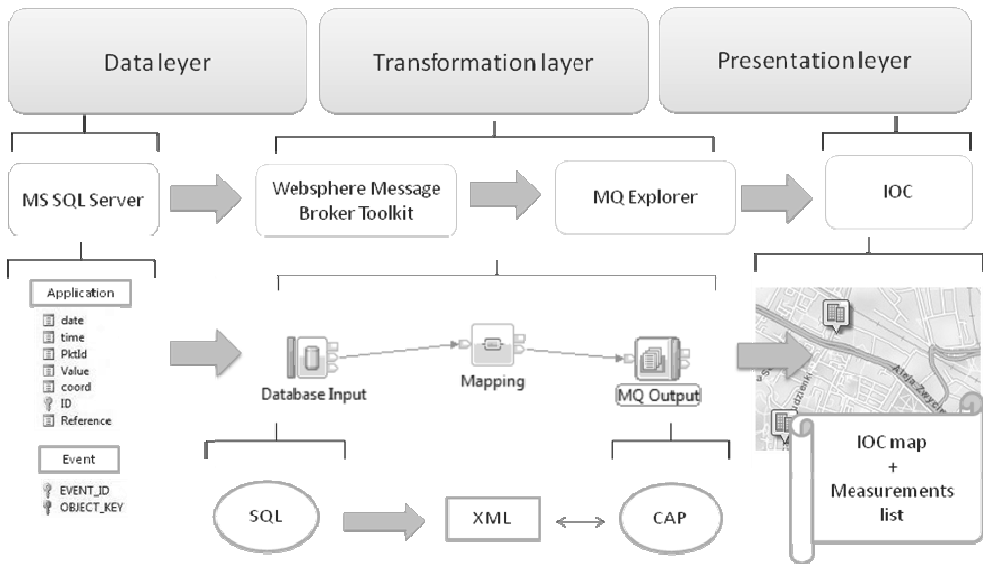
3. Wymagania w stosunku do systemu *Smart Cities*

Jako pierwsze omówiono wymagania w stosunku do systemów *Smart Cities*. Wymagania te były konsekwencją realizowanego dla potrzeb Gdańska projektu Eureka E! 3266 EUROENVIRON WEBAIR. Jego celem jest budowa rozmytych/inteligentnych modeli decyzyjnych do wspomagania zarządzania, jakością powietrza w aglomeracjach. Na etapie przygotowania wniosku nie zakładano, że będzie to system dla inteligentnych miast. Traktowano ten system jako transakcyjny do przetwarzania danych zanieczyszczeń i ich prezentacji na mapie miasta. Zakładano jednak, że w przyszłości właściwym będzie rozbudowa projektowanego systemu, który uwzględni będzie specyficzny układ komunikacyjny Gdańska oraz wpływ dużych przedsiębiorstw (Lotos, Elektrociepłownia, Port Gdański) na kształtowanie się mapy zanieczyszczeń powietrza na terenie aglomeracji. Wówczas opracowane rozwiązanie stanie się łatwo adoptowalnym systemem dla dowolnych aglomeracji borykających się z problemami ochrony środowiska naturalnego oraz zagrożeniami przemysłowymi i wynikającymi z tych zagrożeń konsekwencjami dla mieszkańców.

Założono także, że opracowany w ramach projektu Eureka WEBAIR system, będzie stwarzał warunki do budowy (przez władze municypalne) wieloletnich planów inwestycyjnych oraz strategii rozwoju. Możliwe to będzie dzięki wykorzystaniu rozmytych scenariuszy rozwoju aglomeracji i ich weryfikację w zintegrowanym informatycznym środowisku urbanistyczno-ekologicznym. Uważano, że takie podejście pozwoli na weryfikację skuteczności stosowania przez przedsiębiorstwa systemów zarządzania środowiskowego bazującego na normach ISO 14000 oraz EMAS, ale również pozwoli na szybką identyfikację ewentualnych zagrożeń wynikających z transportu drogowego. Tak szeroko zdefiniowane wymagania stanowiły podstawę do poszukiwania ramy projektowej, która stwarzałaby warunki do projektowania systemu, który spełniać będzie takie wymagania.

Z tego też powodu przed przystąpieniem do budowy systemu dla Urzędu Miejskiego podjęto decyzję dotyczącą wyboru architektury przyszłego systemu. Stwierdzono (biorąc pod uwagę zalety szyny *Enterprise Service Bus* (ESB)), że będzie to architektura SOA wspartą ESB. Kolejnym określił wymagania w stosunku do przyszłego systemu, zaprojektowano architekturę danych w oparciu o środowisko MS

SQL oraz aplikacji w oparciu o RAS (ang. *Rational Software Architect*). Szybnę integracyjną zbudowano w oparciu o *Websphere Message Broker Toolkit*. Warstwę prezentacji przedstawiono w IOC (ang. *Intelligent Operating System*). Zaletą tego rozwiązania jest przede wszystkim dynamiczna konwersja i transformacja danych (ang. *dynamic data transformation and conversion*), rozproszona komunikacja (ang. *distributed communication*) oraz inteligentny routing usług (ang. *intelligent service routing*) (rys. 1).



Rys. 1. Szyna integracyjna systemu zapewniająca przepływ danych na potrzeby systemu IOC

Stosunkowo szybko okazało się, że o ile możliwa jest budowa systemu oparteo o SOA to całościowe widzenie procesów miasta i ich implementacja w IOC jest mało prawdopodobna. Przyczyna takiego stanu rzeczy są: ryzyko takiego całościowego wdrażania (a miasto na poniesienie takiego podejścia nie jest przygotowane) oraz brak przekonania decydentów o potrzebie wrodzenia takiego systemu. Dlatego też zaproponowano (przy całościowym wdrożeniu IOC) metodę projektowania *bottom up* zamiast *top down* dla systemu *Smart Cities* dla Gdańska. Metoda ta bazuje na iteracyjno-przyrostowym podejściu. Dla stworzenia warunków reużywalności tej metody (wykorzystania dla innych miast zainteresowanych projektowaniem IOC) zaproponowano wykorzystanie podejścia CBR dla definiowania wytycznych – metody projektowania i ich ponownego wykorzystywania dla innych miast.

Proponowane podejście bazuje na doświadczeniach zespołu projektowego CAS Gdańsk i obejmuje iteracyjne podejście zarówno do definiowania wytycznych procesów miasta jak też doboru narzędzi projektowych oraz komponentów projektowych dla potrzeb procesu projektowania.

Punktem wyjścia do definiowania metody jest zebranie doświadczeń w projektowaniu systemu dla Gdańska (ewidencji pyłu, hałasu oraz modeli symulacyjnych pyłu i hałasu jak też wskaźników KPI dla pyłu i hałasu) w pierwszej fazie projektu. Ponieważ pierwsza faza projektu kończy się przekazaniem tej działającej wersji dla miasta na spotkaniu projektowym decydenci miast oczekują odpowiedzi na pytania czy pozytywny przypadek projektowania tylko silosowych rozwiązań dotyczących pyłu i hałasu może przełożyć się na implementacje innych silosowych rozwiązań dla miasta. Biorąc pod uwagę to zapytanie jak też potrzeby innych miast (Szczecina i Koszalin) w projektowaniu IOC przeprowadzono eksperymentalne prace projektowe nad analizą możliwości projektowania dowolnych silosowych funkcjonalności IOC. Prace te prowadził 6 osobowy zespół zarządzany z wykorzystaniem metody SCRUM+dobre praktyki. Dobór SCRUM w tym długości dwu sprintów wynikał z wysokiej dojrzałości zespołu projektowego oraz współpracy z innym zespołem zajmującym się implementacją modeli KPI dla potrzeb Bussines Monitora. Definiowano wskaźniki w IOC w ramach 31 procesów projektowych. Sposób realizacji procesów projektowych zamodelowano jako ciąg zadań w Bussines Modelerze i przedstawiono na rys. 2.

W trakcie trwania obu sprintów opracowano modele procesów dla definiowania operacyjnego KPI dla przekraczania pyłu oraz strategicznego dla awaryjnego ładowania samolotu. Brano pod uwagę ograniczone zasoby, który powinny być wykorzystane obu przypadkach (liczbę karet pogotowienia, liczbę łóżek szpitalnych oraz liczbę jednostek straży pożarnych). Zaprojektowano KPI informujące operatora o stanie dostępnych zasobów. Szczegóły procesu wytwarzania KPI, opisy obu wytycznych znajdują się w pozostałych pracach załączonych do numeru.

4. Analiza przypadku projektowania systemów *Smart Cities* dla Gdańska

Na podstawie przedstawionych powyżej wymagań zespół projektowy opracował sekwencję procesów projektowych dla realizacji przedstawionych w poprzednim punkcie wymagań. Nie udało się wprowadzić procesów projektowych do *product backlog* SCRUM ani też nie udało się stworzyć harmonogramu projektu. Sposób projektowania, ustalanie kolejności poszczególnych zadań projektowych wynikało z wiedzy zarządzających zespołem niż z dobrych praktyk SCRUM lub PRINCE 2 oraz RUP (ang. *Rational Unified Prodeses*). Stąd też zaprezentowany poniżej opis procesów projektowych jest zestawem ad post i stanowi podstawę do uogólnienia procesów co z kolei stanowiło podstawę do formułowania metody procesów projektowych *Smart Cities*. W procesach uogólnienia zwracano uwagę na procesy zunifikowane w RUP jak też definiowane role.

Analizując procesy projektowania pod potrzeby IOC zwrócono uwagę na przenikanie się realizowanych w trakcie procesu projektowania produktów oraz wspierających je procesów i ról. Stąd też zaproponowano aby procesy projektowania podzielić na dwa etapy. Etap pierwszy obejmować będzie mapowanie procesów i ról IOC na procesy RUP oraz etap drugi w którym procesy te zostały zagne-

gowane według warstw wysokopoziomowej architektury systemu *Smart Cities*. Architektura ta powstała w oparciu o doświadczenia projektowania systemów *Smart Cities*, wymagani miasta, znajomości technologii oraz wiedzy o projektowaniu złożonych systemów. Jest ona niezbędna dla kierującego projektem i wydaje się, że wiedza o takiej architekturze jest konieczna bez względu na stosowaną metodę zarządzania zespołem.

Poniżej zebrano procesy projektowe biorąc pod uwagę chronologię ich występowania. Obok procesu projektowania podano identyfikator procesu jak też identyfikatory członków zespołu projektowego odpowiedzialnych za realizację procesu. Dokonano także mapowania tych procesów na RUP. Użyto identyfikatorów dla opisu narzędzi: Business Modeller (BM) i Business Monitor (Bm).

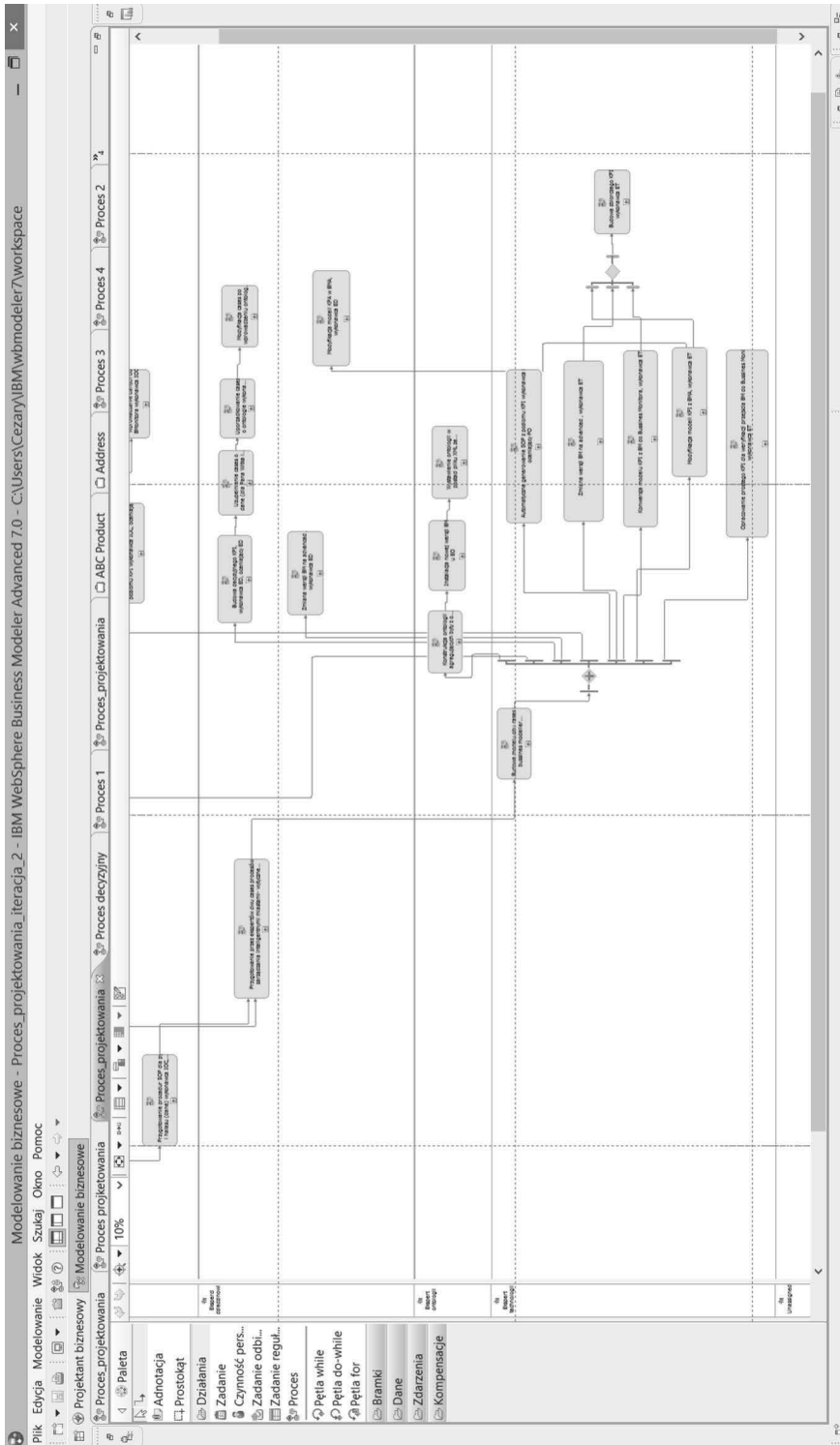
Tab.1. Mapowanie procesów i ról IOC na procesy RUP

Lp.	Proces IOC	Rola IOC	Proces RUP	Rola RUP
1.	Wdrożenie systemu prezentacji danych dotyczących pyłu i hałasu	zespół wsparcia	wdrażanie	<i>Deployment Manager</i>
2.	Opracowanie prostych KPI dla pyłu i hałasu (dane)	zespół wsparcia	implementacja, testowanie, wdrażanie	<i>Tester Designer Implementer</i>
3.	Opracowanie procedur SOP dla pyłu i hałasu (dane)	zespół wsparcia	implementacja, testowanie, wdrażanie	<i>Tester Designer Implementer</i>
4.	Przygotowanie przez ekspertów dwu wytycznych procesów zarządzania inteligentnymi miastami – wytyczne postępowania dla danych operacyjnych – pyłu dla monitorowanych danych oraz strategicznego awaryjnego lądowania samolotu	eksperti dziedzinowi	modelowanie biznesowe	<i>Business Process Analyst</i>
5.	Instalacja integratora	ekspert technologii	implementacja	<i>Implementer</i>
6.	Budowa modelu obu wytycznych BM	ekspert technologii	analiza i projektowanie	<i>Designer</i>
7.	Modyfikacja obu modeli KPI w BM	ekspert technologii	analiza i projektowanie	<i>Designer</i>
8.	Budowa decyzyjnego KPI	ekspert technologii	analiza i projektowanie	<i>Designer</i>

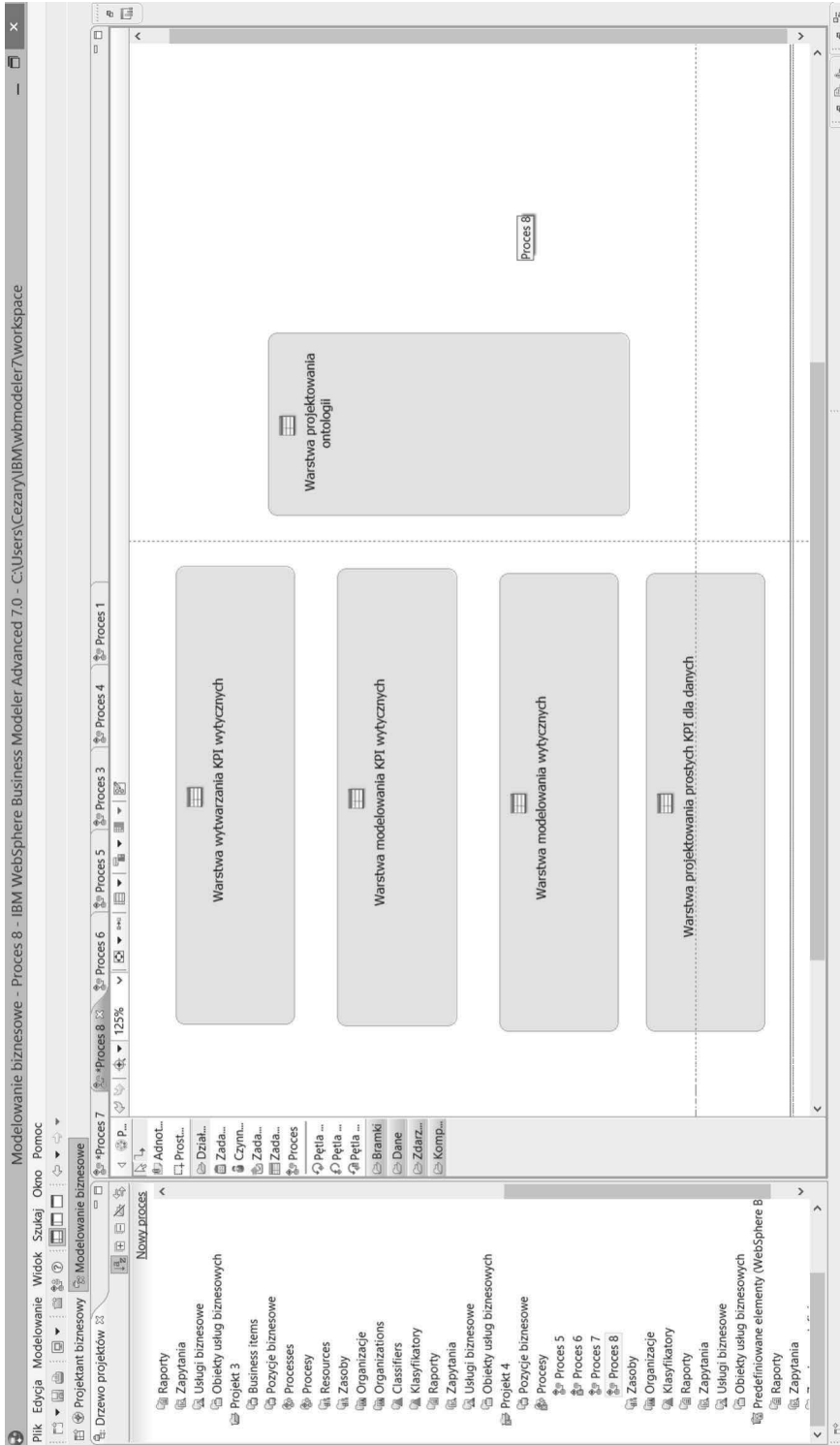
Lp.	Proces IOC	Rola IOC	Proces RUP	Rola RUP
9.	Automatyczne generowanie e-maila z poziomu KPI	zespół wsparcia	implementacja, wdrożenie	<i>Implementer Deployment Manager</i>
10.	Automatyczne generowanie SOP z poziomu KPI	ekspert technologii	implementacja, wdrożenie	<i>Implementer Deployment Manager</i>
11.	Zmiana wersji BM na <i>advanced</i>	ekspert technologii	implementacja, wdrożenie	<i>Implementer Deployment Manager</i>
12.	Konwersja modelu KPI z BM do Bm	ekspert technologii	implementacja, wdrożenie	<i>Implementer Deployment Manager</i>
13.	Modyfikacja modeli KPI z BM	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
14.	Modyfikacja modeli KPA w Bm	eksperti dziedzinowi	modelowanie biznesowe	<i>Designer</i>
15.	Opracowanie prostego KPI dla weryfikacji przejścia BM do Bm	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
16.	Wprowadzanie danych do Bm	zespół wsparcia	wdrażanie	<i>Deployment Manager</i>
17.	Uzupełnienie wytycznych o dane zespołu wsparcia na potrzeby Bm	eksperti dziedzinowi	modelowanie biznesowe	<i>Designer</i>
18.	Konstrukcja ontologii agregujących byty z obu wytycznych przez eksperta od ontologii	ekspert ontologii	modelowanie biznesowe	<i>Designer</i>
19.	Modyfikacja ontologii	eksperti dziedzinowi	modelowanie biznesowe	<i>Designer</i>
20.	Modyfikacja wytycznych po wprowadzeniu ontologii	eksperti dziedzinowi	modelowanie biznesowe	<i>Designer</i>
21.	Instalacja nowej wersji BM	ekspert technologii	implementacja, wdrożenie	<i>Implementer Deployment Manager</i>

Lp.	Proces IOC	Rola IOC	Proces RUP	Rola RUP
22.	Wystawienie ontologii w postaci pliku *.xml ze słownikiem pojęć do weryfikacji wytycznych	ekspert ontologii	modelowanie biznesowe	<i>Designer</i>
23.	Budowa zbiorczego KPI	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
24.	Nazwanie prostego KPI	eksperti dziedzinowi	modelowanie biznesowe	<i>Designer</i>
25.	Modyfikacja nowo nazwanego KPI	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
26.	Agregacja procesów na potrzeby KPI	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
27.	Przyporządkowanie instancji na potrzeby zagregowanych procesów (ET)	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
28.	Określenie KPI dla trzech zagregowanych procesów karetki, pył i samolot	ekspert technologii	analiza i projektowanie, implementacja, wdrożenie	<i>Designer Implementer Deployment Manager</i>
29.	Wysłanie trzech KPI do Bm	ekspert technologii	implementacja, wdrożenie	<i>Implementer Deployment Manager</i>
30.	Implementacja KPI w Bm	ekspert technologii	implementacja	<i>Implementer</i>
31.	Weryfikacja obu modeli projektowanych w BM i Bm dzięki zastosowaniu integratora (ED)	eksperti dziedzinowi	wdrożenie	<i>Deployment Manager</i>

Zaprezentowane procesy (1–31) zostały następnie wpisane do BM (rys. 2) dla zobrazowania ich sekwencji, znaczenia i możliwości przyporządkowania miar biznesowych. Na tym etapie eksperymentu trudno było przypisać tym procesom miary biznesowe stąd też ograniczono się do prezentacji tych procesów i możliwości wspólnej weryfikacji ich znaczenia przez zespół projektowy.



Rys. 2. Sposób realizacji procesów projektowych



Rys. 3. Zagregowany model procesów projektowania

Procesy projektowe zostały następnie poddane klasyfikacji z punktu widzenia wytwarzania produktów na poziomie pięciu warstw architektury systemów *Smart Cities*. Stanowiły także w oparciu o przedstawione na rys. 2 procesy projektowania wskaźników KPI podstawę do ich agregacji dla formalnego opisu przypadku procesu wytwarzania. Architekturę systemów *Smart Cities* dla projektowania KPI z udziałem ontologii przedstawiono na rys. 3.

Wyodrębniono pięć warstw procesów i przyporządkowano je do wytwarzania produktów dla pięciu warstw architektury systemów *Smart Cities*:

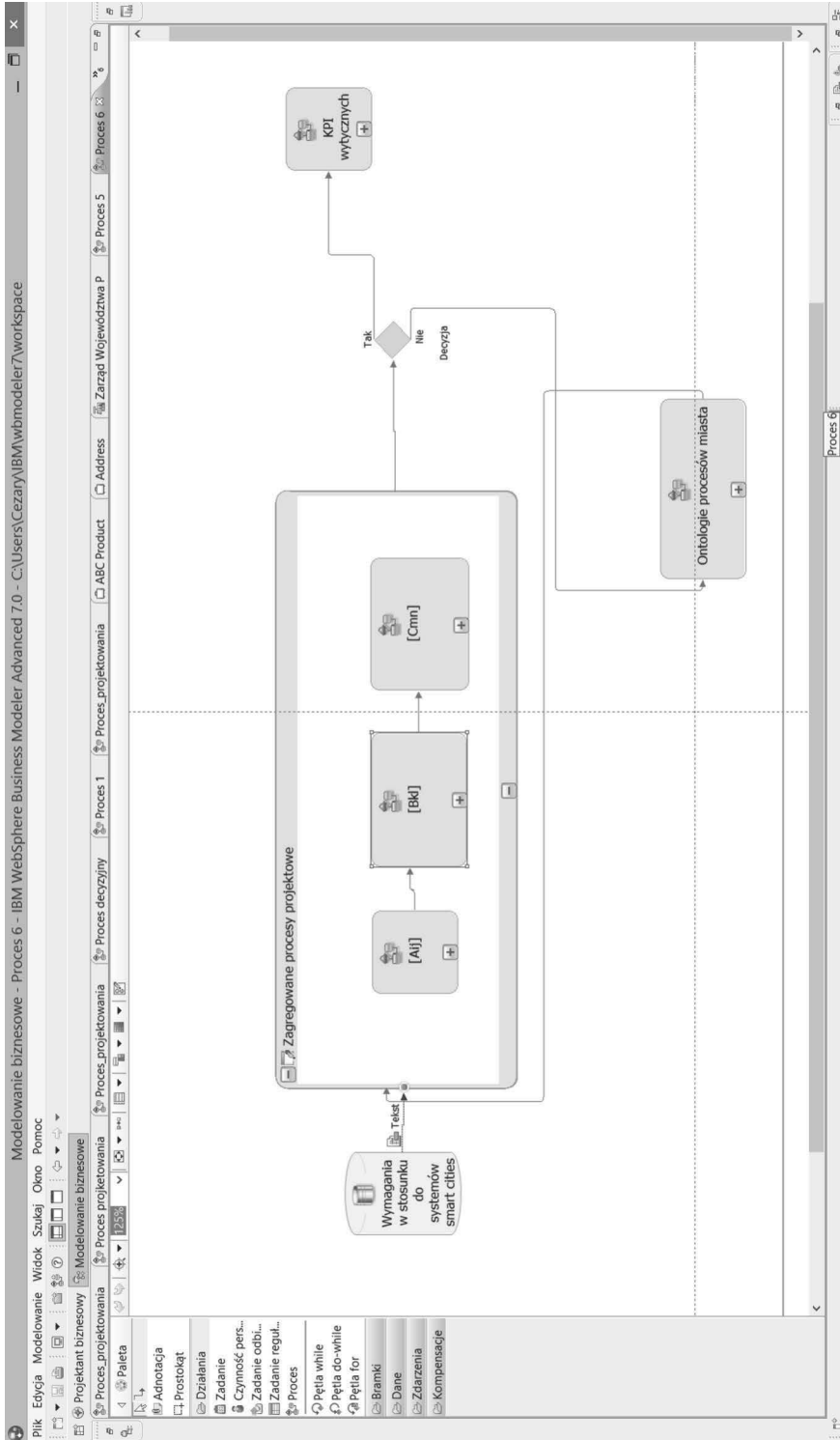
- budowa prostych KPI dla danych (procesy 1–3),
- modelowanie wytycznych (procesy 4–6),
- budowa modeli KPI dla wytycznych (procesy 7–18),
- projektowanie ontologii dla procesów miasta (procesy 19–22)
- projektowanie KPI wytycznych (procesy 23–31).

Wybór tych pięciu warstw procesów wynikał z doświadczeń zespołu projektowego zarządzanego z wykorzystaniem metodyki SCRUM i dobrych praktyk zaczerpniętych z innych metod zarządzania projektami. Stosowanie SCRUM staje się możliwe tylko wówczas, kiedy *Product Owner* zna *Architecture High Level* i na tej podstawie definiuje procesy projektowania.

Dobór tych pięciu warstw był konsekwencją konstrukcji narzędzi projektowych IBM Websphere i filozofii ich integracji dla potrzeb procesu projektowania KPI. Należy tutaj wspomnieć że tworzenie KPI jest integralną częścią projektowania systemów dla potrzeb firm. Wydaje się, że firma IBM wykorzystwała doświadczenia narzędzi projektowych dla modelowania KPI stosowanych dla firm dla potrzeb miast. Dyskusyjne jest zarządzanie miastem z wykorzystaniem KPI, ale jest to temat jednej z załączonych prac w ramach której analizuje się procesy miasta oraz formalne obiekty, aby wykazać na ile rozwiązanie oparte o KPI może być wykorzystane dla potrzeb zarządzania inteligentnymi miastami.

5. Model przypadku projektowania systemów *Smart Cities* dla Gdańska

Zebrane procesy projektowe jak też doświadczenia związane z wykorzystaniem CBR stanowiły bazę dla autorów projektu dla zdefiniowania przypadku dla Gdańska, który stanowić będzie podstawę do jego wpisania do bazy przypadków i późniejszej analizy. W przygotowaniu bazy wiedzy przypadków zastosowano opis generyczny w którym procesy zmapowano za pomocą opisu macierzowego. Opis ten i traktowanie zagregowanych procesów jako systemu ze sprzężeniem zwrotnym zaprezentowano na rys. 4.



Rys. 4. Opis generyczny procesów projektowania systemów *Smart Cities* bazujący na CBR

Zagregowane procesy projektowe i role zostały zapisane w postaci macierzy procesów i ról:

$$\mathbf{A} = [A_{ij}], \quad \mathbf{B} = [B_{kl}], \quad \mathbf{C} = [C_{mn}] \quad (4)$$

gdzie dla analizowanego przypadku Gdańska: $i = 1, 2, 3$; $j = 1, 2$; $k = 1, 2, 3$; $l = 1, 2$; $m = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$; $n = 1, 2, 3, 4$

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad (5)$$

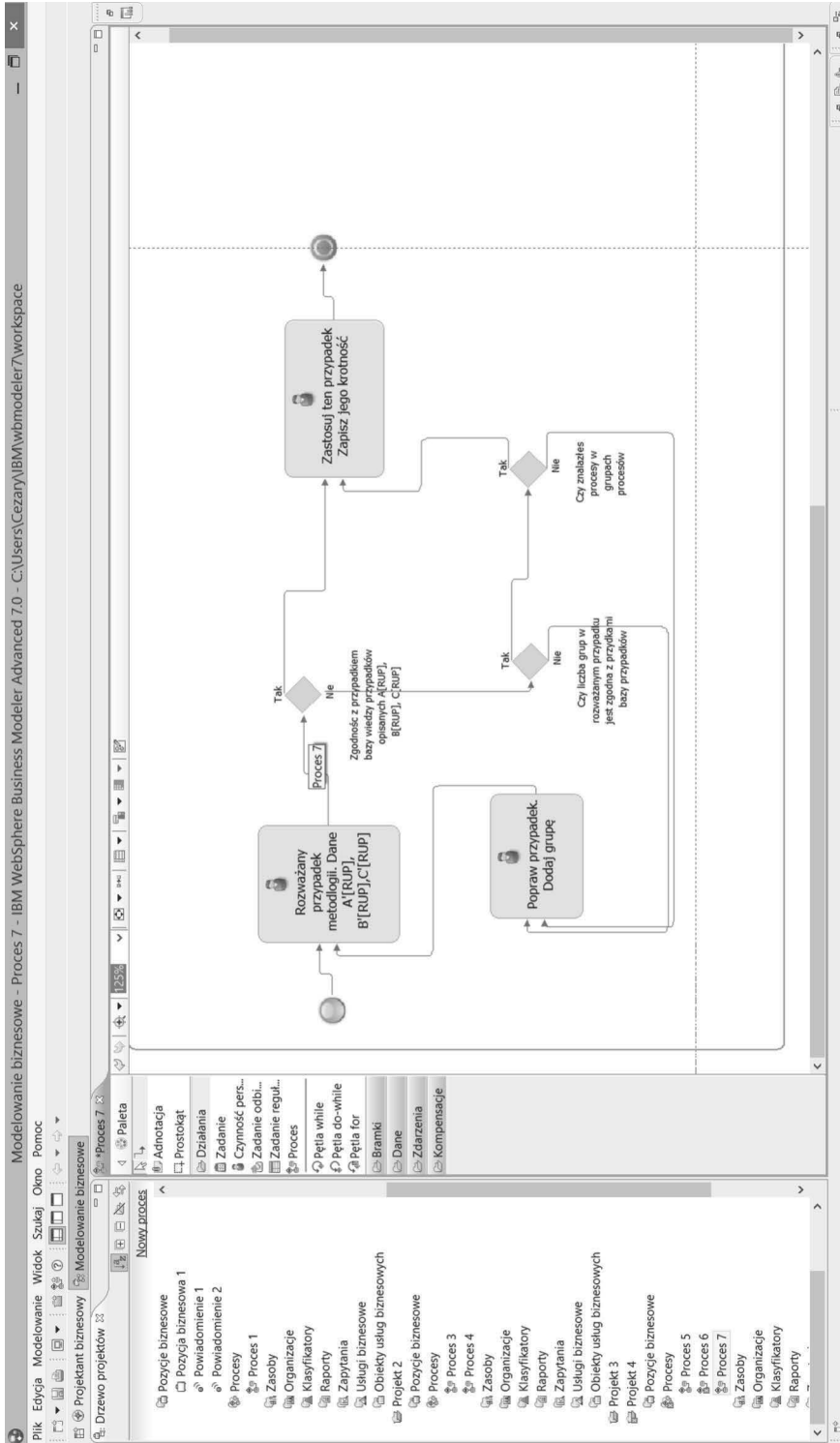
$$\mathbf{B} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix} \quad (6)$$

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \\ C_{51} & C_{52} & C_{53} & C_{54} \\ C_{61} & C_{62} & C_{63} & C_{64} \\ C_{71} & C_{72} & C_{73} & C_{74} \\ C_{81} & C_{82} & C_{83} & C_{84} \\ C_{91} & C_{92} & C_{93} & C_{94} \\ C_{101} & C_{102} & C_{103} & C_{104} \\ C_{111} & C_{112} & C_{113} & C_{114} \\ C_{121} & C_{122} & C_{123} & C_{124} \end{bmatrix} \quad (7)$$

W opisie procesów i ról dla każdej z projektowanych warstw architektury systemów *Smart Cities* zastosowano rozwiązanie, w którym elementy macierzy opisane są w następujący sposób:

$$[A_{ij}] = \text{string}(\text{nazwa_warstwy} + \text{nazwa_procesu} + \text{nazwa_rol}) \quad (8)$$

Ograniczono liczbę znaków elementów macierzy do 9 zakładając po trzy znaki dla każdego z elementów opisu. I tak pierwszy proces z tablicy 1 jeden będzie identyfikowany jako: „*dkp*wd*rdem*” co oznacza warstwę prostych KPI dla danych „*dkp*”, proces wdrożenia „*wdr*” oraz rolę *deployment managera* ‘*dem*’. Wszystkie pozostałe procesy i role zostały w podobny sposób zmapowane.



Rys. 5. Schemat wykorzystania metody CBR dla opisu przypadku projektowego zmapowanego do RUP

Następnie macierze te zostały zmapowane do ról i procesów RUP (rys. 5).

$$\mathbf{A}_{\text{RUP}} = [A_{ij}], \quad \mathbf{B}_{\text{RUP}} = [B_{kl}], \quad \mathbf{C}_{\text{RUP}} = [C_{mn}] \quad (9)$$

gdzie dla analizowanego przypadku Gdańska zdefiniowano na podstawie tabeli 1 liczbę procesów i ról dla każdej grupy: $i = 1, 2$; $j = 1, 2$; $k = 1, 2, 3$; $l = 1, 2$; $m = 1, 2, 3, 4$; $n = 1, 2, 3, 4$.

Stąd też procesy i role RUP stanowiły podstawę do definiowania procesów i ról dla metody projektowania systemów *Smart Cities*.

$$\mathbf{A}_{\text{RUP}} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$\mathbf{B}_{\text{RUP}} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$\mathbf{C}_{\text{RUP}} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{35} & C_{36} & C_{37} & C_{38} \end{bmatrix}$$

Opis macierzowy uwzględnia liczbę procesów i liczbę ról odpowiedzialnych za realizację tych procesów. Dla przykładu procesy budowy prostych KPI dla danych (procesy 1–3) uwzględniają trzy procesy i dwie role. Opis macierzowy staje się niezbędny przy identyfikacji procesów i ról. Jeżeli bowiem założymy, że uruchomiony zostanie proces na CBR to sekwencja procesu doboru przypadku następuje zgodnie z zaproponowanym opisem macierzowym i prezentować się będzie jak na rys. 5.

Rozważamy przypadek metody procesu projektowania opisany zagregowanymi procesami:

$$\mathbf{A}'_{\text{RUP}} = [A'_{i'j'}], \quad \mathbf{B}'_{\text{RUP}} = [B'_{k'l'}], \quad \mathbf{C}'_{\text{RUP}} = [C'_{m'n'}] \quad (8)$$

Zakładamy także (rys. 5), że dla każdej z podanych macierzy liczba elementów nie musi być większa od zera (liczba elementów jest pochodną procesu projektowania co oznacza, że nie zawsze dany proces projektowania występuje) i rozpoczynamy proces przeszukiwania bazy wiedzy przypadków. Jeżeli znajdujemy przypadek w którym wszystkie trzy macierze występują wówczas badamy elementy tych macierzy. Jeżeli którakolwiek z macierzy nie występuje analizowany przypadek wymaga modyfikacji. W procesie przeszukiwania analizujemy także liczbę elementów danej macierzy. Jeżeli analizowany proces nie występuje należy zmodyfikować ten przypadek poprzez dodanie danego elementu. Porównywanie przypadku analizowanego z przypadkami w bazie wiedzy przypadków odbywać się będzie z zastosowaniem testu identyczności opisanego wzorem 1.

6. Weryfikacja opracowanego modelu – pozyskiwanie danych dotyczących procesu projektowania dla innego miasta

Podstawą do weryfikacji modelu były dokumenty dostarczone przez inne miasto w Polsce zainteresowane w drożeniem systemu IOC. Dokumenty te stanowiły podstawę do określania wymagań stawianych przed systemem, który ma być projektowany dla miasta. Uwzględniono w pracach analizy wymagań następujące dokumenty:

- charakterystyka zagrożeń oraz ocena ryzyka ich wystąpienia, w tym dotyczących infrastruktury krytycznej. W dokumencie tym określono kategorię zagrożenia, prawdopodobieństwo jego wystąpienia, rodzaj zagrożenia oraz skutki i sposoby działania w momencie pojawienia się zagrożenia;
- siatkę bezpieczeństwa – dokument, w którym przedstawiono kategorie zagrożeń i osoby, które powinny być powiadomione w momencie wystąpienia tych zagrożeń;
- uruchamianie działań – dokument, w którym opisano procedury postępowania w momencie pojawienia się zagrożenia;
- plan działań krótkoterminowych – dokument, w którym przedstawiono procedury postępowania ale dotyczące specyficznych zagrożeń jak pył i hałas.

Zbiór dokumentów obrazuje z jednej strony formalny opis zagrożeń i działań w momencie pojawienia się tych zagrożeń a z drugiej strony stanowi formalny opis wymagań w stosunku do systemu. Na podstawie wymagań opracowano procesy projektowe, które stają się niezbędne dla opracowania systemu na potrzeby miasta.

Ponieważ urząd miasta, który przedstawił te wymagania oczekuje na zaproponowanie metody projektowania systemu IOC, stąd też zaproponowano aby wykorzystać zapisany w Business Modeler przypadek metody projektowania IOC dla Gdańska zastosować dla miasta, które dostarczyło tych wymagań. Najprostsze (wymagania dotyczące pyłu i hałasu) znalazły się w dokumencie „Plan działań krótkoterminowych” stąd też poddano analizie te wymagania pragnąc zastosować podobną metodę projektowania systemu. Oceniono podobieństwa i różnice pomiędzy wymaganiami stawianymi dla systemu w Gdańsku a proponowanymi przez miasto. Okazało się, że mimo podobieństwa procedur (rozporządzenia dotyczące działań w zakresie PM10 są podobne dla wszystkich miast w Polsce) to sposób implementacji tych procedur w oparciu o dokument „Siatka bezpieczeństwa” jest odmienny. Odmiennie staje się też projektowanie KPI.

Stwierdzono natomiast, że cztery dostarczone dokumenty stanowią pełny obraz zagrożeń, procedur postępowania i osób decyzyjnych co umożliwi stosunkowo proste definiowanie ontologii dla poszczególnych KPI, grup KPI i całego systemu. Stąd też wydaje się za celowe aby włączyć do warstwy ontologii procesy związane z przygotowaniem pełnego obrazu zagrożeń i ich znaczenia jak to ma miejsce w mieście, które dostarczyło wymagań. Stąd też w warstwie ontologii, mimo że istnieje proces budowy ontologii dla poszczególnych KPI, dodany zostanie proces tworzenia ontologii dla wszystkich procesów miasta.

Wątpliwości autorów budziło umieszczenie zagregowanych procesów ontologii w pętli sprzężenia zwrotnego tak jak to miało miejsce w modelu procesów (rys. 3). Zarówno z punktu widzenia modelu (rys. 2) jak i prezentacji obiektów tworzących przypadek dla procesu CBS nie do końca jest jasna rola procesu sprzężenia zwrotnego. Próbowano to określić w procesach wersyfikacji. Ponieważ procesy weryfikacji dostarczyły informacji na temat znaczenia i roli tego procesu zarówno obiekt (ontologie procesów miasta) jak też jego połączenie z pozostałymi procesami pozostawiono warstwę ontologii zakładając jej kluczowe znaczenie dla procesu projektowania. Stwierdzono także, że celowa będzie modyfikacja dwu procesów dla warstwy projektowania prostych KPI i modelu KPI. Zmiany zostaną naniesione w bazie przypadków projektowania systemów *Smart Cities*.

7. Wnioski

W artykule przedstawiono metodę projektowania systemów *Smart Cities* opartą na analizie przypadków. Przedstawiona w artykule metoda staje się wydajna w sytuacji problemów związanych z projektowaniem systemów o tak dużej skali złożoności i braku możliwości stosowania zarówno miękkich jak i twardych metod zarządzania projektami. Bazuje ona na analizie przypadków, konwersje zapisu procesów projektowych i ich wykorzystywania dla projektowania systemów w dowolnym mieście. Baza przypadków staje się wówczas swoistą bazą wiedzy, w której dokumentowane są metody projektowania systemów dla dowolnych miast.

W pracy wykazano, że budowa takiej bazy wiedzy pozwala na identyfikację procesów projektowych i ról projektowych dla projektowania poszczególnych warstw architektur systemów *Smart Cities*. Ta zaprezentowana w artykule jest konsekwencją wykorzystywanych w procesach projektowania technologii IBM. Te wykorzystywane w pracy do modelowania procesów, ich integracji prezentują stan funkcjonowania miasta w postaci wskaźników KPI. Należy przypomnieć, że wskaźniki te typowe dla zarządzania firmą zostały zaadoptowane dla potrzeb zarządzania inteligentnymi miastami. Tylko dlatego projektując architekturę *Smart Cities* wykorzystano te wskaźniki.

Procesy projektowania wskaźników realizowano w oparciu o procesy projektowe organizacji. Dla ich urzeczywistnienia przeprowadzono eksperyment naukowy, w którym stworzono zespół projektowy z udziałem rzeczywistych specjalistów od projektowania systemów. Eksperyment, którego celem było wypracowanie metody poprzez wybór dwóch przypadków zarządzania miastami, a później weryfikacji na podstawie wymagań innego miasta, wykazał przydatność proponowanej metody.

Znaczącą rolę w projektowaniu systemów i wsparcia wysokopoziomowej architektury odegrały ontologie procesów miasta. W przypadku danych Gdańska proces tworzenia ontologii był złożony (ograniczona liczba danych). Z kolei w przypadku miasta, które było zainteresowane procesem projektowania znaczna liczba dokumentów uporządkowała proces projektowania ontologii, a tym samym uporządkowała wytyczne projektowania i wykorzystanie zapisanych przypadków z bazy wiedzy przypadków.

Zastosowanie przetwarzania przypadków z wykorzystaniem CBS pokazuje, że w sytuacji w której tworzenie systemów *top down* staje się niemożliwe a *bottom up* mocno ograniczone zastosowanie analizy przypadków do projektowania procesów i doboru ról ich konwersja na format RUP i zapis w bazie wiedzy przypadków stwarza warunki do doboru metody projektowania dla potrzeb miast. Należy pamiętać, że wykorzystanie metody jest ograniczone znajomością wysokopoziomowej architektury systemu oraz środowiska projektowego. Prezentowane w pracy rozwiązanie jest wolne od tych ograniczeń, ponieważ tworzy spójną ramę przetwarzania i może być stosowane dla dowolnych architektur i ram projektowych.

Źródła

1. Bhowmick A.: (2009). *IBM Intelligent Operations Center for Smarter Cities Administration Guide 5*. Event flow diagnostic and validation tool for IBM WebSphere Business Monitor. International Business Machines Corporation.
2. *Common Alerting Protocol Version 1.2* (2013). Pobrano z lokalizacji: <http://docs.osasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.pdf>.
3. Czarnecki A., Orłowski C.: (2010). Ontology As a Tool For The IT Management Standards Support. *Lecture Notes in Artificial Intelligence, No 6071*, 330-339. Berlin Heidelberg: Springer –Verlag.
4. *IBM Intelligent operations center Information Center* (2013). Pobrano z lokalizacji: <http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>.
5. *IBM WebSphere application server information center* (2013). Pobrano z lokalizacji: <http://publib.boulder.ibm.com/infocenter/cities/v1r5m0/index.jsp>.
6. *IBM WebSphere Broker Message Broker Information Center* (2013). Pobrano z lokalizacji: <http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/index.jsp>.
7. Kowalczyk Z., Orłowski C.: (2004). Design of knowledge-based systems in environmental engineering. *Cybernetics and Systems: An International Journal* 35 (5–6), 487–498.
8. Kowalczyk Z., Orłowski C.: (2012). *Modelowanie Procesów Zarządzania Technologiami Informatycznymi*. Gdańsk: Pomorskie Wydawnictwo Naukowo-Techniczne PWNT.
9. Pastuszak J., Orłowski C.: (2013). *Model of rules for IT organization evolution Transactions on Computational Collective Intelligence IX*, (55–78).
10. Pastuszak J., Stolarek M., Orłowski C.: (2010). Concept of Generic IT Organization Evolution Model. *Zeszyty Naukowe Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej*, Vol. 18, Nr 8 (235–240).
11. Pokojski J.: (red.) (2003). *Zastosowanie metody case-based reasoning w projektowaniu maszyn*. Gdańsk: WNT.
12. Smith A.D.: (2015). *IBM Intelligent Operations Center KPI Implementers Guide for Websphere Software, Document version 1.0*.
13. Snadach K.: (2013). Graphical data presentation in IBM Intelligent Operations Center., Diploma Dissertation. Gdańsk.

SMART CITIES SYSTEM DESIGN METHOD BASED ON CASE BASED REASONING

The objective of this paper is to present the results of research carried out to develop a design method for Smart Cities systems. The method is based on the analysis of design cases of Smart Cities systems in cities, the selection of the city appropriate to the requirements for implementation and application. The Case Based Reasoning method was used to develop the proposed design methodology, along with mechanisms of the conversion of project processes and roles to Rational Unified Processes (RUP). The prerequisite for the proposed method is that the enterprise manager must be knowledgeable about high-level Smart Cities system architecture and the design framework applied. The authors, being themselves knowledgeable about architecture of this kind and about project environments which implement KPI models, propose a generic solution applicable to any environments and system architectures.

Key words: Smart Cities, Case Based Reasoning Method, decision support systems, knowledge management.