Współczesne kierunki rozwoju informatyki

20. Jesienne Spotkania PTI

Mrągowo 2004
22-26 listopada

- VII Forum Technologii Informatycznych
- Synergia polskiej nauki i branży informatycznej – mity i fakty
- I Forum Menedżerów IT

POLSKIE TOWARZYSTWO INFORMATYCZNE

WSPÓŁCZESNE KIERUNKI ROZWOJU INFORMATYKI

XX JESIENNE SPOTKANIA PTI MRĄGOWO, 22–26 LISTOPADA 2004

Pod redakcją
Piotra W. Fuglewicza, Bogusława Jackowskiego,
Ewy Mizerskiej oraz Anny Ostaszewskiej

KATOWICE–MRĄGOWO 2004

Współczesne kierunki rozwoju informatyki
Materiały XX Jesiennych Spotkań PTI, Mrągowo, 22–26 listopada 2004

REALIZACJA
Górnośląski Oddział Polskiego Towarzystwa Informatycznego

KOMITET ORGANIZACYJNO-PROGRAMOWY
Anna Andraszek, Grzegorz Bliźniuk, Borys Czerniejewski, Jarosław Deminet, Janusz Dorożyński, Piotr W. Fuglewicz (przewodniczący), Bogusław Jackowski, Ewa Mizerska (sekretarz), Jerzy Nowak, Anna Ostaszewska, Wiesław Paluszyński, Marcin Paprzycki

WYDAWCA
Polskie Towarzystwo Informatyczne

Skład systemem TEX: BOP s.c., http://www.bop.com.pl/

SPIS TREŚCI

Piotr W. Fuglewicz
Słowo wstępne 5
Adam Bartos, Zbigniew Wyszomierski
Wartość zintegrowanych metadanych w rozwiązaniach Business Intelligence 9
Tomasz Byzia
Inżynieria etyki 19
Andrzej Czyżewski
Wdrożenia z dziedziny usług multimedialnych i zastosowań cyfrowego przetwarzania sygnałów 25
Helmut V. Gläser
Po pierwsze – nie motywuj! Jak każdy rodzaj motywowania bezpośrednio prowadzi do demotywacji 31
Lech Jóźwiak
Life-inspired Systems: Why and How? 39
Suku Krishnaraj, Murthy Srinivas, Jacek Walicki
Quality of Service Through Infrastructure on Real-Time Demand 53
Leszek A. Maciaszek
Architectural Patterns for Design and Refactoring of Enterprise Information Systems 57
Darek Milewski
Strategia bezpieczeństwa, zarządzanie ryzykiem: trendy i przykłady najlepszych praktyk 69
Tadeusz Nowicki
Symulacja komputerowa w projektowaniu systemów czasu rzeczywistego 73
Leszek Pacholski
Jakie uniwersytety? 79
Grzegorz Rawicz-Mańkowski
Wiedza czy narzędzia? Ewolucja polskich programów nauczania od teorii do praktyki 89
Rafał Stefański
Racjonalne zarządzanie irracjonalnymi zachowaniami 91

Mierzę więc jestem

Marek Średniawa
O infrastrukturze technicznej społeczeństwa informacyjnego czyli na czym stoimy (jedną nogą) 99
Andrzej Kobyliński
Miary procesu i produktu programowego 105
Beata Czarnacka-Chrobot
Pomiar rozmiaru funkcjonalnego systemu informatycznego – sposób na „chaos permanens"? 111
Michał Goliński
Zmierzyć niemierzalne czyli pytania o społeczeństwo informacyjne 121

SŁOWO WSTĘPNE

Jubileuszowe, dwudzieste już, Jesienne Spotkania PTI po raz kolejny stają się miejscem spotkania informatyków zajmujących się technologią, kierowników zajmujących się informatykami, naukowców badających zarówno ludzkie, jak i techniczne aspekty zastosowań środków informatyki w życiu, oraz producentów i konsultantów z branży teleinformatycznej.

Od osób odpowiadających za strukturę teleinformatyczną firmy i za programistów wymaga się z dnia na dzień coraz szerszych kompetencji. Aby nie zabrnąć w ślepą uliczkę, muszą wiedzieć, jakie będą tendencje rozwoju systemów teleinformatycznych w najbliższych miesiącach i latach, a do tego powinny znać się po trosze na dziedzinach, których na studiach nie przerabiały: ekonomii, socjologii, psychologii, prawie, zarządzaniu.

Materiały zawarte w niniejszym tomie są próbą odpowiedzi na wieloaspektowość zawodu informatyka w dzisiejszych czasach. Poruszają tematy związane z wieloma dziedzinami, których przynajmniej elementarna znajomość jest co najmniej pożądana, a często konieczna.

Obecność na XX Spotkaniach PTI Polaków z zagranicy, a ich tekstów w niniejszym tomie, ma miejsce w najwłaściwszym czasie. Zwłaszcza teraz wiedza Polaków zdobyta za granicą może i powinna być jak najlepiej wykorzystana. Przykład Irlandii czy Portugalii, które po akcesji skorzystały istotnie z doświadczeń swoich emigrantów, wart jest naśladowania. Dlatego też redaktorzy tomu postanowili opublikować teksty polskich gości z zagranicy w wersji, w jakiej powstały, nie tylko po to, aby dać uczestnikom okazję do treningu językowego, ale również dlatego, że niektóre z nich dotyczą problematyki niemającej jeszcze swojej polskiej terminologii.

Życząc, aby niniejszy tom był mądrym ku nauce, pozwalam sobie zaprosić na kolejne dwadzieścia wydań mrągowskich spotkań.

Piotr W. Fuglewicz
Katowice – Mrągowo, listopad 2004

REFERATY

WARTOŚĆ ZINTEGROWANYCH METADANYCH W ROZWIĄZANIACH BUSINESS INTELLIGENCE

ADAM BARTOS, ZBIGNIEW WYSZOMIERSKI

SAS Polska

Streszczenie: Wystąpienie omawia rolę metadanych w środowisku systemów klasy Business Intelligence (BI), ich potencjalne wykorzystanie zarówno w różnych fazach życia tego typu systemów jak i w poszczególnych elementach ich architektury technicznej wysokiego poziomu. Po dokonaniu tego przeglądu prezentowane są korzyści, jakie wnosi zintegrowana architektura metadanych na przykładzie rozwiązań zastosowanych w najnowszej linii produktów SAS.

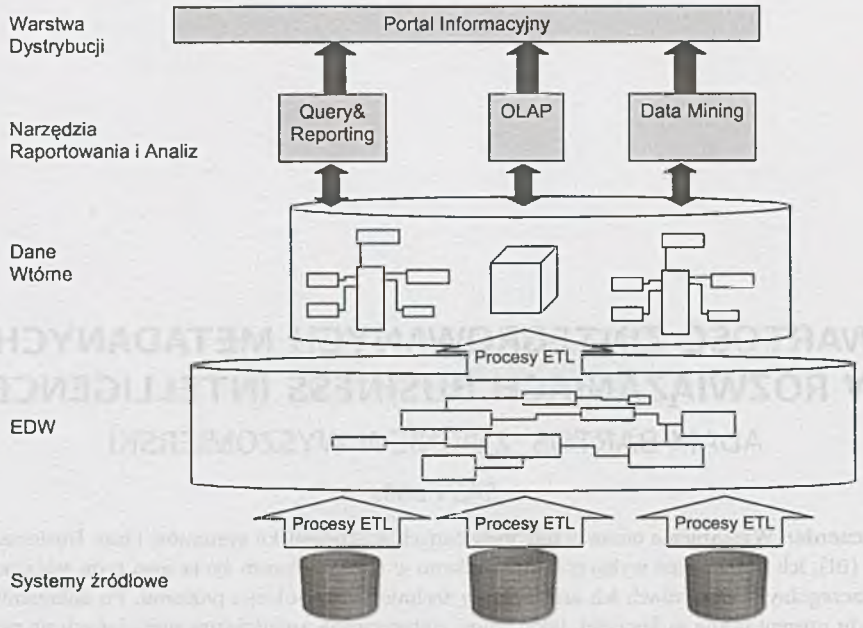
Wstęp

Zagadnienia związane z pojęciem metadanych a nawet same próby definicji czym są metadane, charakteryzują się daleko idącą niejednoznacznością. W związku z tym porzuciliśmy zamierzenie zaadresowania tematu metadanych w możliwie ogólny i uniwersalny sposób, oceniając, że lepiej będzie skoncentrować się na tej tematyce w kontekście zastosowań metadanych w szeroko rozumianych systemach klasy Business Intelligence. Obejmuje to zagadnienia związane z Hurtowniami Danych, ich tworzeniem, administracją i eksploatacją przy pomocy różnorodnych narzędzi raportowania i analiz. Prezentowane w tym artykule tezy i opinie reprezentują wizję i koncepcje SAS Institute wypracowane przez ponad 25 lat doświadczeń firmy w rzeczonym zakresie, niemniej z natury rzeczy mogą mieć subiektywny i ograniczony charakter.

Aby lepiej określić domenę dyskusji spróbujmy zdefiniować, z jakich elementów składa się typowe środowisko hurtowni danych. Rysunek 1 (p. następna strona) przedstawia najważniejsze z punktu widzenia dyskusji elementy.

Systemy źródłowe, czyli różnego typu systemy wspierające daną organizację w zakresie bieżącej działalności, gromadzące dane opisujące zjawiska i procesy operacyjne. Przykładowo mogą być to systemy klasy ERP, systemy zarządzania relacjami z klientem stosowane np. przez telefoniczne centra obsługi, systemy finansowo-księgowe, systemy bilingowe, sprzedaży itp.

Procesy ETL, czyli zadania mające na celu ekstrakcję danych z systemów źródłowych, ich oczyszczenie, transformację i ostatecznie zasilenie danych w docelowym repozytorium. Procesy ETL występują zarówno na styku między systemami źródłowymi a Enterprise Data Warehouse, jak też pomiędzy EDW a repozytoriami danych wtórnych w środowisku hurtowni danych.



Rysunek 1. Typowe elementy środowiska hurtowni danych

Enterprise Data Warehouse (EDW), czyli warstwa zintegrowanych i oczyszczonych danych atomowych, zorganizowana tematycznie, nieulotna i uporządkowana czasowo, modelująca mniej lub bardziej kompleksowo zagadnienia interesujące organizację w zakresie procesów wsparcia podejmowania decyzji. Zazwyczaj modelowana i implementowana w oparciu o relacyjne bazy danych. Dostępna dla użytkownika dla potrzeb ad hoc ale przede wszystkim będąca pewnym źródłem danych dla kolejnej warstwy.

Data Marts (dane wtórne), czyli warstwa danych wtórnych, zawierająca dane wybrane, zagregowane, przeznaczone często do wykorzystania przez konkretną aplikację czy narzędzia, opisujące najczęściej konkretny temat biznesowy. Realizowane jako tzw. „dimensional warehouse" w strukturach typu gwiazda, w bazach relacyjnych, ale również jako tzw. bazy wielowymiarowe, a w niektórych zastosowaniach (np. data mining) struktury zdenormalizowane.

Narzędzia raportowania i analiz, czyli aplikacje pozwalające na wykonywanie analiz OLAP, tworzenie zapytań do baz danych, tworzenie raportów graficznych, modelowanie i analizy data mining itp. Różnorodne narzędzia przeznaczone są dla użytkowników o różnym stopniu specjalizacji i doświadczenia, mających za zadanie przygotowanie docelowej informacji w odpowiedniej formie.

Warstwa dystrybucji, czyli mechanizmy dostarczania gotowych wyników (różnego typu raportów) dla docelowego odbiorcy informacji, realizowana najczęściej jako portal informacyjny.

W tak zdefiniowanym środowisku systemów informacyjnych można zdefiniować pewne klasy uczestników procesów informacyjnych. Najważniejsze z nich to:

- projektanci hurtowni i danych wtórnych, odpowiedzialni za modelowanie danych w pierwotnych i wtórnych repozytoriach hurtowni,

- programiści procesów ETL, odpowiedzialni za przygotowanie zadań zasilających docelowe modele,
- administratorzy hurtowni, odpowiedzialni za „utrzymanie w ruchu" raz zbudowanej hurtowni i procesów ETL, udostępnianie zasobów hurtowni w zgodzie z polityką bezpieczeństwa,
- analitycy biznesowi, pracujący z danymi hurtowni przy pomocy narzędzi raportowania i analiz, odpowiedzialni za przygotowanie rzetelnej informacji w formie dynamicznych (czyli interaktywnych) jak i statycznych raportów, będący niejako ich „właścicielami" odpowiedzialnymi za ich zawartość informacyjną. Również osoby samodzielnie pozyskujące zasoby informacyjne dla swoich procesów decyzyjnych w oparciu o dane z hurtowni.
- konsumenci informacji, korzystający z wynikowej informacji do celów wsparcia procesów podejmowania decyzji.

Na tym tle można pokusić się o zdefiniowanie, czym są metadane (lub czym mogą być) w takich systemach i jaką rolę mogą pełnić. Najczęściej spotykana definicja mówi, że metadane to „dane o danych". Często spotykany jest podział na tzw. metadane techniczne i biznesowe. Metadane techniczne używane są przez profesjonalistów IT (projektanci, programiści, administratorzy) podczas planowania, projektowania, tworzenia i administracji hurtownią danych; z drugiej strony metadane biznesowe pozwalają użytkownikom bez specjalistycznej wiedzy technicznej (analitycy biznesowi, konsumenci informacji) na odpowiedzenie sobie na pytania gdzie znaleźć informacje w hurtowni, skąd one pochodzą i w jaki sposób w niej się znalazły, jakiej jakości są te informacja jest oraz w jaki sposób należy je interpretować [2]. Metadane techniczne to przykładowo:

- opis struktury i formatu danych źródłowych, docelowego modelu danych, definicje i charakterystyki tabel i kolumn, indeksów, kluczy, widoków itp.,
- specyfikacja zadań ładowania, ich kod, parametry, harmonogram wykonywania i wzajemne zależności,
- definicje elementów infrastruktury, takich jak serwisy sieciowe, serwery danych, serwery aplikacji,
- metody dostępu, uprawnienia i hasła dostępu do danych źródłowych,
- podstawowe statystyki dot. procesów ładowania danego obiektu danych: czas utworzenia i modyfikacji, czas dodania rekordu, status aktualności, informacje o błędach,
- metadane opisujące reguły czyszczenia, wzbogacania, transformacji, mapowania pól danych,
- metadane opisujące hierarchie w strukturach OLAP, sterujące aplikacjami wykorzystującymi modele wielowymiarowe.

Metadane biznesowe z kolei to na przykład:

- informacja o właścicielu i opis biznesowy systemów źródłowych,
- informacja o właścicielu i biznesowy opis i nazwa elementów danych,
- definicje elementów danych w oparciu do reguły biznesowe,
- opis poprawnych wartości pól o charakterze wyliczeniowym,
- wskazówki dot. wykorzystania danych, możliwych powiązań i relacji,
- opis dokonanych przekształceń dla kolumn wyliczanych.

W pewnym sensie odrębną grupę metadanych stanowią informacje administracyjne takie jak profile użytkowników, ich prawa dostępu, przynależność do grup itp.

Przyjrzyjmy się teraz bliżej jak metadane mogą być wykorzystywane i w jaki sposób powstają w procesach planowania i konstrukcji, procesach ETL a następnie w procesach raportowania i analiz oraz administracji środowiskiem. Popatrzmy, jakie korzyści przynoszą, jeżeli są konsekwentnie definiowane i wykorzystywane.

Metadane w procesie planowania i projektowania systemów klasy BI

Proces projektowania systemów klasy BI jest niewątpliwie etapem krytycznym. Przyjęcie właściwej koncepcji i założeń może w skali czasu okazać się czynnikiem decydującym o powodzeniu lub fiasku całego przedsięwzięcia. Możemy zatem w tym miejscu zastanowić się nad rolą i znaczeniem metadanych w procesie projektowania.

Model metadanych może składać się z setek obiektów różnego typu, które w praktyce mogą istnieć w dowolnej ilości instancji. Odpowiednie obiekty powiązane są pomiędzy sobą relacjami i mogą być wykorzystywane do składowania informacji technicznych i biznesowych. Przykładem powszechnie przyjętego standardu w tym zakresie jest Common Warehouse Metamodel (CWM), wspierany przez kilkaset firm i organizacji na całym świecie. Wykorzystanie środowiska BI, pracującego w oparciu o standard CWM, gwarantuje swobodę w wymianie informacji (metadanych) pomiędzy różnymi środowiskami aplikacyjnymi.

Przy projekcie rozwiązań klasy BI pierwszym krokiem jest niewątpliwie budowa właściwych modeli danych wykorzystywanych zarówno w warstwie Enterprise Data Warehouse jak i na poziomie Data Marts. Modele danych wykorzystywane na tym etapie są pewnego rodzaju pochodną modeli danych wykorzystywanych w systemach źródłowych zasilających danymi systemy BI.

Podstawowym narzędziem modelowania danych używanym przez projektantów EDW i Data Marts są środowiska klasy CASE. Również modele opisujące systemy źródłowe często dostępne są w formie elektronicznej. A więc możliwość wymiany metadanych z tymi środowiskami jest istotnym czynnikiem płynnego procesu projektowania środowiska BI. Informacje otrzymane w tym etapie znacznie redukują czas potrzebny na projektowanie systemu BI, z drugiej zaś strony są doskonałym źródłem uzupełnienia wiedzy na temat istniejących zasobów dla projektantów hurtowni i data martów.

Metadane na etapie tworzenia procesów ETL

Przetwarzanie danych w swojej najprostszej formie ograniczyć się może do kilku lub kilkunastu linii kodu zmieniającego dane z jednego układu w inny. Z czasem problematyka przetwarzania danych jest rozbudowywana o zagadnienia związane z dostępem do różnych formatów, w których znajdują się dane źródłowe. Początkowo proste operacje przetwarzania na skutek stale wzrastających potrzeb odbiorców zmieniają się w coraz bardziej skomplikowane procesy. Stale wzrastająca liczba wykonywanych zadań przetwarzania musi uwzględniać także pojawiające się zależności pomiędzy nimi. Dodatkowa funkcjonalność w postaci nowych transformacji tworzona przez pojedynczych programistów powinna być udostępniana innym, a wszyst-

kie te operacje powinny także uwzględniać cechy specyficzne dla danego kraju lub regionu.

W ujęciu tradycyjnym metadane generowane w warstwie procesów ETL ograniczały się jedynie do gromadzenia informacji czysto technicznych takich jak: kolumny, tablice, lokalizacja i parametry związane z dostępem do danych. W środowiskach produkcyjnych wykorzystanie informacji zawartych w metadanych na poziomie procesów ETL nie powinno ograniczać się jednak jedynie do działań związanych z eksploracją i wykorzystaniem dostępnych danych. Model metadanych pozwala na skonstruowanie środowiska ETL, które jest niezależne od bieżącej konfiguracji i formatu danych tzn. zmiana położenia (lokalizacji) lub formatu składowania wykorzystywanych zasobów nie powinna mieć wpływu na ciągłość procesów przetwarzania. Właściwe wykorzystanie cech funkcjonalnych modelu metadanych pozwala przede wszystkim na pełną organizację wszelkich działań występujących na tym poziomie.

A zatem do poruszanych wcześniej zagadnień realizowanych w oparciu o informacje zawarte w metadanych należy dodać:

- *synchronizację procesów ETL*: wykonanie poszczególnych procesów może być przede wszystkim uzależnione od zaprojektowanego kalendarza. Kalendarz może dotyczyć podstawowych elementów takich jak: dzień tygodnia, godzina, miesiąc itp., ale także może być rozbudowywany o dodatkową funkcjonalność, jak np. ostatni pracujący dzień miesiąca. W dalszym kroku metadane opisujące procesy ETL mogą być wykorzystywane do uruchamiania zadań w zależności od stanu realizacji innych procesów;
- *administrację zmianami danych*: wykorzystanie metadanych w procesie wersjonowania i audycie zmian w strukturach danych;
- *pracę grupową*: wykorzystanie metadanych w procesie synchronizacji przy pracy grupowej;
- *wykorzystanie metadanych zawartych w systemach źródłowych*: w wykorzystywanych systemach źródłowych (w szczególności ERP) poza dostępem do części danych możliwy jest także bezpośredni dostęp do ich metadanych i pełne wykorzystanie informacji w nich zawartych;
- *współdzielenie nowych transformacji*: metadane pozwalają tworzyć dodatkową funkcjonalność środowiska ETL i mogą być wykorzystywane do organizacji schematu pracy z nimi przez programistów procesów ETL;
- *zarządzanie kodami programów*: kody podobnie jak inne zasoby mogą być także opisywane w postaci metadanych, dzięki czemu znaczna ich część może być generowana automatycznie. Metadane mogą dodatkowo zapewnić lepsze ich wykorzystanie poprzez dzielenie kodu na mniejsze części i pełne opisanie jego cech funkcjonalnych.

Metadane w procesie raportowania i analiz

Najpierw spróbujmy zastanowić się, dlaczego metadane są tak istotne w fazie eksploatacji środowiska BI. W jakimś stopniu to od ich istnienia i jakości zależy skala sukcesu wdrożenia projektu w zakresie BI. Dlaczego? Sukces tego typu projektów, mierzony wpływem hurtowni danych na procesy podejmowania decyzji, jest w uproszczeniu wprost proporcjonalny do liczby osób, które aktywnie korzystają z informacji

dostarczanej przez hurtownię, bądź to analizując ją samodzielnie, bądź korzystając z gotowych raportów. Aby ktokolwiek w praktyce skorzystał z danych czy informacji musi być spełnione kilka warunków: po pierwsze odbiorca musi wiedzieć gdzie je odnaleźć, rozumieć ich prawdziwe znaczenie, a także *musi* tej informacji ufać. Zwróćmy uwagę, że im dalej – w sensie organizacyjnym i umiejętności – odbiorca informacji znajduje się od firmowego centrum przetwarzania danych tym trudniej jest spełnić te warunki i tym mniejsza liczba osób będzie w stanie wykorzystać zasoby informacyjne organizacji. Analityk biznesowy zapewne wie, że „gdzieś" są dane o klientach czy sprzedaży za ostatni miesiąc, ale aby się do nich dostać musi zapewne zapytać kilka osób, które również zweryfikują czy aby wskazana przez administratora tabela, zawiera właściwe i aktualne informacje. Nawet, jeżeli mamy już dostęp do tabeli to jak interpretować znaczenie poszczególnych pól? Takie aktywności wymagają czasu i wysiłku, co w skali dużej organizacji oznacza również dodatkowe koszty obsługi procesów informacyjnych.

Właśnie tu kluczowe znaczenie mają metadane. Przykładowo, dzięki „ukryciu" poprzez warstwę metadanych, fizycznego położenia (serwer, baza, tabela) czy realizacji modelu danych (np. gwiazda lub kostka wielowymiarowa) analityk może samodzielnie, przy pomocy znajomego sobie interfejsu środowiska raportowego odszukać odpowiednie dla swoich potrzeb informacje. Dzięki mechanizmom wyszukiwania metadanych, może odnaleźć pojedyncze, interesujące go elementy danych posługując się biznesowymi ich definicjami (np. zysk ze sprzedaży) i zlokalizować je w pełnym kontekście (czyli identyfikując tabelę w której ta informacja jest zestawiona z innymi). Pozwala to na zrealizowanie zadań w zakresie raportowania i analiz szybciej i skuteczniej. Kolejne zastosowanie metadanych to tzw. „data lineage" czyli możliwość prześledzenia pochodzenia danej informacji polegająca na określeniu, jakim transformacjom podlegała po drodze od systemów źródłowych do raportu. Dzięki temu analityk lub konsument informacji może łatwiej zweryfikować wiarygodność oraz adekwatność danych i informacji, których używa bądź to w celu uzyskania niezbędnego do nich zaufania bądź też w celu stymulacji poprawy jakości procesów przetwarzania i dostarczania informacji.

Kluczowe dla powyższych zastosowań rodzaje metadanych to: biznesowa nazwa pola, biznesowy opis i definicja, biznesowa nazwa i opis tabeli, częstotliwość odświeżania danych, data aktualizacji, logika biznesowa, algorytm wyliczania, opis transformacji używanych podczas tworzenia, czyszczenia, agregacji czy wyliczeń.

Dostępność informacji i metadanych (bezpośrednio lub pośrednio poprzez wykorzystanie ich przez aplikacje) poprzez portal informacyjny zasadniczo zwiększa dostępność informacji w organizacji. Z drugiej strony metadane administracyjne pozwalają zapanować nad rozpowszechnianiem informacji zgodnie z ustaloną polityką bezpieczeństwa. Metadane mogą opisywać, kto i do jakich danych, raportów, aplikacji udostępnianych przez portal może mieć dostęp.

Metadane w procesie administracji systemem BI

Dobrze zaprojektowana architektura metadanych poza wsparciem codziennej pracy użytkowników pozwala także na zorganizowanie szeregu procesów i działań administracyjnych:

- konfiguracja procesów autentykacji i autoryzacji dostępu do zasobów,
- realizacja mechanizmu „single signon",
- konfiguracja środowiska Business Intelligence,
- administracja zawartością repozytorium metadanych.

Model metadanych może zapewniać wsparcie dla składowania wszelkich informacji związanych z polityką bezpieczeństwa organizacji, mechanizmami autentykacji i autoryzacji dostępu do zasobów:

- aplikacji,
- systemów baz danych,
- innych środowisk wspierających codzienną pracę użytkowników.

Obiekty, reprezentujące pojedynczych użytkowników lub ich grupy pozwalają na scentralizowanie wszelkich informacji związanych z identyfikatorami i hasłami wykorzystywanymi przez pracowników. Dzięki takiemu rozwiązaniu w oparciu o metadane możliwa jest realizacja mechanizmu „single signon", polegającego na tym, że przy dostępie do kolejnych zasobów wymagających autoryzacji niezbędne informacje (identyfikatory i hasła) są odczytywane z metadanych, co znacznie upraszcza cały proces dostępu do danych.

W oparciu o metadane można zbudować centralny mechanizm autoryzacji dostępu zapewniający jednolity standard definiujący zakres dostępnych zasobów dla grup lub pojedynczych użytkowników. Tak zdefiniowanej polityce bezpieczeństwa podlegają zarówno metadane techniczne (źródła i struktury danych) jak i metadane biznesowe opisujące istniejące raporty (ich strukturę i przeznaczenie) i analizy.

Pomimo tak rozległego zakresu gromadzonych metadanych sam proces zarządzania całym modelem metadanych powinien być z punktu widzenia administratora prosty i intuicyjny.

Niezależnie od rodzaju metadanych, kręgu ich odbiorców (użytkownicy, informatycy, analitycy) czy też aplikacji, które będą je wykorzystywać administrator metadanych może używać jednego uniwersalnego środowiska. Wykorzystując je może on monitorować „stan obecny" metadanych jak też wykonywać inne działania, jak np. konfiguracja środowiska Business Intelligence.

Wartość zintegrowanych metadanych

Typowe organizacje utrzymują wiele często wzajemnie niepowiązanych systemów, aplikacji i źródeł danych. Takie niepowiązane systemy są często niezależnymi „wyspami informacji i metadanych". Utrzymanie takiego stanu nie tylko wymaga dużych nakładów pracy i czasu, ale może też powodować, że użytkownicy biznesowi otrzymują niekompletne lub nieprecyzyjne informacje a zespoły IT spędzają niezliczone ilości czasu, tropiąc źródła informacji, konsolidując dane, ręcznie poprawiając niezależne repozytoria metadanych i w końcu asystując użytkownikom w procesie tworzenia wiedzy.

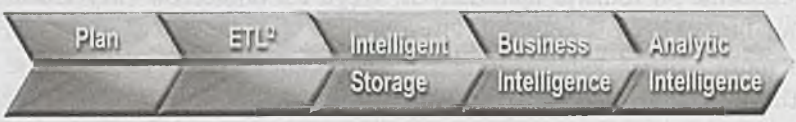
Zintegrowane metadane (czyli m.in. informacje o danych źródłowych, sposobie ich pozyskania, regułach biznesowych i prawach dostępu) są krytyczne dla wyprodukowania spójnej i dokładnej informacji. Gdyby informacje ze wszystkich aplikacji mogły być przechowywane w otwartym, centralnym i zintegrowanym repozytorium, zmiany dotyczące danych byłyby dokumentowane tylko w jednym miejscu, mniej systemów

wymagałoby wsparcia, a użytkownicy biznesowi mogliby liczyć na informacje o wysokiej jakości. Jedna wersja prawdy byłaby dostępna dla wszystkich a lepsze wykorzystanie czasu pracowników, obniżyłoby całkowite koszty posiadania infrastruktury IT.

Z wcześniejszej dyskusji widać, że metadane towarzyszą (a przynajmniej powinny) całemu procesowi dostarczania informacji w środowiskach business intelligence. Co więcej, po chwili zastanowienia widać, że te same lub bardzo podobne metadane mogą być wykorzystywane na różnych etapach cyklu życia takiego systemu przez osoby pełniące różne role. Dla przykładu projektant, definiujący model danych, tworzy metadane opisujące kolumny tabeli, wprowadzając również biznesową definicję ich znaczenia. Programista procesów ETL, wykorzystuje metadane o kolumnach do definiowania transformacji i z kolei tworzy metadane dot. przekształceń i mapowania kolumn z systemów źródłowych oraz metadane opisujące zadania ładowania. W końcu analityk biznesowy wykorzystuje opisowe informacje o tabeli przy pracy nad swoim raportem, a administrator hurtowni korzysta z opisu zadań ładowania podczas tygodniowego cyklu odświeżania danych. W takim kontekście widać, jak dużą pomocą może być zintegrowana architektura metadanych, zapewniająca ich współdzielenie przez różne klasy uczestników procesów informacyjnych oraz różne narzędzia informatyczne, a przy okazji niezbędną spójność i jednoznaczność (jedna wersja prawdy).

W skali organizacji taka architektura powinna mieć pozytywny i stymulujący wpływ m.in. na: szybkość procesów informacyjnych, łatwość (a tym samym koszty) rozbudowy systemów i danych w hurtowni o nowe obszary zastosowań, a także koszty i efektywność procesów administracji.

Taka wizja jednolitej architektury metadanych przyświecała projektantom SAS 9 Intelligence Platform, opartej o koncepcję łańcucha wartości (SAS Intelligence Value Chain, dalej IVC), której elementem jest SAS Open Metadata Architecture.



Rysunek 2. SAS Intelligence Value Chain

W skrócie SAS IVC składa się z następujących ogniw:

Plan – SAS wspiera potrzeby związane z planowaniem poprzez sprawdzone w praktyce podejścia, oparte o zintegrowane, specyficzne dla gałęzi gospodarki modele danych, zaprojektowane dla potrzeb analitycznych, metodologie projektowe i doświadczenie zespołów konsultingowych.

SAS ETL^{Q} jest wysokowydajną platformą integracji danych, syntetyzującą dane korporacyjne z wzajemnie niepowiązanych źródeł w sposób efektywny zarówno z punktu widzenia czasu, jak i kosztów.

SAS Intelligent Storage jest dedykowaną platformą repozytoriów danych zaprojektowaną od początku w celu efektywnego udostępniania informacji zarówno dla potrzeb business intelligence, jak i zaawansowanych analiz (analytic intelligence).

SAS Business Intelligence, która dostarcza zintegrowaną korporacyjną architekturę z interfejsami dedykowanymi dla różnych grup użytkowników celem wsparcia szybkiego i wiarygodnego procesu podejmowania decyzji.

SAS Analytic Intelligence, która jest wszechstronną, zintegrowaną platformą dla szerokiego zakresu analiz. Jej możliwości obejmują modele dla predykcji i opisu, prognozowanie, optymalizację, symulację, planowanie eksperymentów i inne.

Każde ogniwo (za wyjątkiem *Plan*), z technologicznego punktu widzenia reprezentowane jest przez zestaw narzędzi i interfejsów użytkownika, dedykowanych poszczególnym rolom w procesach informacyjnych, opartym o wspólną infrastrukturę techniczną, w tym wspólną dla całego łańcucha warstwę metadanych.

W jaki sposób wykorzystywane są i do czego mogą w praktyce posłużyć wspólne metadane, najlepiej zilustrować na kilku przykładach.

Współdzielenie metadanych w całym procesie dostarczania informacji

Ze względu na zmiany na rynku, niezbędna jest dodatkowa informacja dla nowych raportów i analiz. Niezbędne dane istnieją w systemach ERP ale nie zostały jeszcze uwzględnione w hurtowni danych.

Przy użyciu interfejsu *SAS ETL Studio* i *SAS Data Surveyor*, projektant hurtowni przeszukuje metadane środowiska ERP w celu odnalezienia niezbędnych danych. Zidentyfikowane tabele pojawiają się na liście danych źródłowych (metadane automatycznie tworzone w tle) i mogą być natychmiast użyte (poprzez mechanizm przeciągnij i upuść) w procesie ekstrakcji i transformacji. Docelowy model danych w hurtowni zdefiniowany przy pomocy narzędzi klasy CASE jest importowany do ETL Studio i definiuje tabele wynikowe tych procesów. Projektant (programista) procesów ETL definiuje transformacje kolumn między źródłem a docelowym modelem, niezbędne połączenia, filtry i wyliczenia. Określa częstotliwość uaktualniania i w końcu generuje zadanie. W trakcie, zupełnie automatycznie powstają i są zapisywane metadane o tym procesie, które z kolei wykorzystuje administrator hurtowni, definiując informacje dla programu szeregującego.

Idąc dalej, ponieważ docelowo chodzi nam o interaktywny raport dla użytkownika, projektant wykorzystuje metadane opisujące tak utworzone tabele, podczas definiowania miar, wymiarów i hierarchii kostki wielowymiarowej, oraz określa jej niezbędne parametry celem zdefiniowania fizycznej struktury dla *SAS OLAP Server*. Jednocześnie automatycznie definiowany jest proces, który ładuje tę kostkę na podstawie tabeli wejściowej określonej powyżej. Administrator przy użyciu *SAS Management Console* określa, którzy użytkownicy będą mieli do niej dostęp i w jakim zakresie. Środowisko raportujące będzie automatycznie uwzględniało prawa dostępu podczas pracy użytkownika. Projektant uzupełnia opis kostki o metadane biznesowe, ukrywając fizyczne położenie kostki i opisując jej elementy przy pomocy zrozumiałej dla użytkownika terminologii (tworzona jest tzw. Information Map). Analityk biznesowy wykorzystuje dostępne mapy (w ogóle nie zastanawiając się, gdzie są faktyczne dane), w celu zbudowania nowego raportu przy pomocy SAS Web Report Studio. Nie musi się zastanawiać, jak połączyć odpowiednie dane (bo to zrobił już wcześniej projektant). Nowy raport jest zapisywany (oczywiście w metadanych) i staje się automatycznie dostępny dla uprawnionych użytkowników poprzez *SAS Information Delivery Portal*.

Przenoszenie danych bez wpływu na funkcjonowanie aplikacji, raportów czy analiz

Organizacja IT wymienia sprzęt na nowszy celem sprostania wymaganiom nowych grup użytkowników. Dane przechowywane na obecnym sprzęcie będą przeno-

szone na nowy podczas trzymiesięcznego okresu czasu. Problemem są programy i aplikacje użytkowników, które używają tych danych na co dzień. W jaki sposób pracownicy IT mogą przenieść dane na nowy serwer bez komplikacji dla biznesowego użytkownika?

Dzięki centralnemu i współdzielonemu repozytorium metadanych, dane i fizyczne serwery mogą zostać bez przeszkód przeniesione. Definicje ich fizycznego położenia są przechowywane na serwerze metadanych i zarządzane przez Data Library Manager („plugin" do SAS Management Console). Wystarczy zmienić wpis w metadanych, tak, aby wskazywał na nowe położenie. Zespół IT nie musi sprawdzać i poprawiać wszystkich programów, które korzystają z danych, ręcznie kodując fizyczne nazwy nowych serwerów. Programy i użytkownicy w przezroczysty sposób „zobaczą" nowe dane zupełnie nie zastanawiając się gdzie się fizycznie znajdują.

Literatura

[1] *Materiały korporacyjne SAS Insitiute*, 1976–2004

[2] John E. Bentley: *Metadata: Everyone Talks About It, But What Is It?*, SUGI 26 Proceedings, April 22–25, 2001, Long Beach Convention Center, Long Beach, California

[3] John Singer: *Meta Data Repository Redux*, Part 1 and Part 2, artykuł opublikowany w *DM Direct Newsletter*, April 9, 2004, Issue 2004

INŻYNIERIA ETYKI

TOMASZ BYZIA

Premium Technology Sp. z o.o., Warszawa
tbyzia@premiumtechnology.pl

Uczę się więc oddawać usługi innemu człowiekowi, nie żywiąc do niego żadnej przychylności; a to dlatego, że przewiduję, iż odwzajemni się za moją usługę, oczekując innej usługi tego samego rodzaju, oraz w tym celu, iżby zachować tę samą wymianę świadczeń ze mną czy też z innymi. Toteż, gdy ja mu oddałem usługę i gdy osiągnął korzyść z mojego działania, skłonny jest on wykonać swoje świadczenie wzajemnie, ponieważ przewiduje konsekwencje swojej odmowy.

David Hume, *Traktat o ludzkiej naturze*, 1740[1]

Inżynieria etyki

Termin „inżynieria etyki" wyłonił się z mroku pewnego ciepłego weekendowego wieczoru, który, jak nietrudno zgadnąć, spędzałem w pracy, biedząc się nad warunkami realizacji pewnej ciekawej, aczkolwiek wielce ryzykownej oferty. Przyparty do muru licznymi obiekcjami i kontrargumentami klienta zacząłem usilnie poszukiwać sytuacji „win-win", dochodząc po kilku godzinach do wniosku, że taka nie istnieje. Zawsze ktoś – moja firma lub klient – przegrywał.

Zaraz. Ale skoro nie możemy razem wygrać, to kto z nas „powinien" przegrać! Moja firma czy klient? Czy klient ma dostać dobry produkt naszym kosztem? Czy to my mamy zarobić jego kosztem? Racjonalny umysł podsuwał jedyne oczywiste rozwiązanie, ale wpojone przez lata zasady działania „dla maksymalizacji satysfakcji klienta" psuły radość ulegnięcia pokusie.

Gra o sumie zerowej? „Przecież w realnym życiu takie nie istnieją" – zabrzmiał w głowie głos starego nauczyciela. I tak się zaczęło... Scott Adams twierdzi, że prawdziwy inżynier nigdy się nie poddaje i najlepszym testem jego determinacji jest postawienie mu do rozwiązania jakiegoś skomplikowanego i nikomu niepotrzebnego problemu, np. „Dilbercie, czy to prawda, że włosa nie da się podzielić na czworo?". Wiedziony tym samym instynktem inżynierskim zacząłem poszukiwać rozwiązania mojego dylematu. Oferta ofertą, ale przecież musi być jakieś wyjście z tej sytuacji.

1. David Hume, *Traktat o ludzkiej naturze*, przeł. Czesław Znamierowski, Polska Akademia Umiejętności, Kraków, 1951.

I tak zrodziła się „inżynieria etyki", zajmująca się dostarczaniem narzędzi sprawnego wykonywania zadań i osiągania etycznych celów w praktyce gospodarczej i społecznej. Etyka jako nauka filozoficzna daje solidną podbudowę teoretyczną na wzór matematyki, królowej nauk inżynierskich. Inżynieria etyki, koncentruje się na obserwacji rzeczywistości i wypracowaniu zestawu dobrych praktyk, pomagających przy dokonywaniu codziennych wyborów moralnych.

Mój punkt widzenia tych zagadnień jest oczywiście subiektywnym spojrzeniem „jakościowca", który z racji wykonywanego zawodu na wszystko patrzy z innej, „trzeciej" strony. Nie potrafię odpowiedzieć, czy przez to moje opinie nabierają charakteru bardziej obiektywnego i uniwersalnego. Z całą pewnością są jednak inne...

Firma vs klient

W każdym zawodzie wymiar etyki relacji z klientem jest oczywisty. Każdy zdrowo myślący człowiek uzna, że nie godzi się oszukiwać i naciągać klienta, że trzeba wywiązać się z umowy, być rzetelnym i terminowym, dostarczyć to, co klient zamówił, dbać o jakość itd. Obszar ten jest także tradycyjnie najlepiej chroniony przez regulacje prawne, które stają się gwarantem praw klienta wobec dostawcy.

Jeśli chodzi o dostawców systemów informatycznych, niedojrzałość inżynierii oprogramowania, brak regulacji prawnych oraz mała wiedza potencjalnych klientów na temat zastosowań informatyki dają praktycznie nieograniczone możliwości nadużyć, które nie mogą być ścigane przez prawo. Prawie każdy klient bez zastanowienia godzi się na warunki licencji oprogramowania, narzucane przez dostawców, którzy nie biorą żadnej odpowiedzialności za swoje oprogramowanie. Klient jest bezbronny, bo przecież rozrywając opakowanie z płytką CD zgodził się na warunki dostawcy umieszczone w środku.

Szukając rozwiązań swoich problemów klient zdany jest na łaskę zespołu czarnoksiężników, wypowiadających nad komputerem tajemnicze zaklęcia. Klientowi brak wiedzy, doświadczenia i modelu odniesienia, który pozwoliłby na obiektywną ocenę ryzyka i rzetelności składanych ofert. Strach i niepewność prowadzi do nieufności, a ta staje się przyczyną swoistej gry rynkowej, w której odbiorca i dostawca prezentują przeciwstawne interesy. Klient chce dobrze, szybko i tanio. Dostawca też chce dobrze, dlatego musi być długo, a więc drogo. Kłamstwo pierwsze to znacznie zawyżona cena oferty – klient i tak będzie chciał ją obniżyć. Kłamstwo drugie to cena lepsza niż u konkurencji – jest ona przecież najczęściej podstawowym kryterium oceny oferty. Kłamstwo trzecie to często zbyt daleka od rzeczywistości deklaracja wiedzy, umiejętności i doświadczenia wystarczających do sprawnej realizacji całego przedsięwzięcia. Tak więc już w pierwszej fazie negocjacji stanowiska stron są mocno spolaryzowane i bazują na nieprawdziwych przesłankach, wynikających z nieprawdziwych danych. Następujący potem proces uzgadniania celów, cen, terminów, warunków współpracy itd., to proces wygrywania małych bitew. Końcowa postać umowy jest zapisem chaotycznych potyczek o szczegóły, nie zaś opisem środowiska optymalnego z punktu widzenia przyszłej realizacji projektu. Potem komentatorzy rynku informatycznego nie zostawiają suchej nitki na dostawcach systemów informatycznych. Wytyka się ich niekompetencję, nieuczciwość, „kopertowe" załatwianie kontraktów, niemoralność prezesów, brak elementarnej etyki zawodowej itd. Czy jednak słusznie?

Każda ze stron za etyczne uważa to, co prowadzi do realizacji JEJ celów. W ten sposób każda ze stron postępuje moralnie, dążąc do osiągnięcia tego co uważa za słuszne i dobre zgodnie ze SWOJĄ etyką. Nie istnieje wspólny model odniesienia dla obu stron. Nie możemy, nie znając zasad i celów drugiej strony, zarzucać jej niemoralnego postępowania tylko dlatego, że cele te są inne niż nasze. Gdyby istniał wspólny model wartości, musiałby przede wszystkim określić reguły tworzenia oprogramowania, podobne do np. reguł sztuki medycznej. Zasady te, pozwoliłyby zawsze obiektywnie rozstrzygnąć, kto popełnił „błąd w sztuce". Czy w tak młodej dziedzinie jak informatyka jest to możliwe? Wątpię.

Klient kontra dostawca

Omawiane problemy relacji z klientem pojawiają się w sytuacji, gdy relacje te niesłusznie postrzegamy jako jednostronne. Klient nasz pan. On płaci i on wymaga. Satysfakcja klienta jest najważniejsza. Wymienione modele relacji z klientem miały bronić jego interesów, strzec przed nieuczciwymi dostawcami oraz wymuszać wysoką jakość produktów. Okazuje się, że często obracają się przeciwko klientowi.

Często już na etapie przetargu klient zachowuje się jak cesarz posyłający na śmierć oddział gladiatorów. Niech wygra najlepszy. Siła i honor. Problem w tym, że walka konkurencyjna rządzi się swoimi prawami i – jak informatycy wiedzą doskonale – rynek pełen jest rozwiązań o wątpliwej jakości, które jednak nie dały się pokonać w otwartej walce. Czy oferent, który wygląda i zachowuje się profesjonalnie, obiecał najwyższą jakość za najniższą cenę i w najkrótszym czasie, jest najlepszy? Skąd wiadomo, jakie są jego doświadczenia, jakim dysponuje potencjałem realizacyjnym, jaki jest poziom dojrzałości wewnętrznych procesów, czy wie, co to jest jakość, jak wygląda wewnętrzny system zapewnienia jakości w jego firmie? Napisanie dobrej oferty nie jest łatwe, ale nie jest też aż takie trudne. Specjalista zrobi to bez problemu, a klient nie będzie w stanie zweryfikować tego, co w niej napisano. To trochę tak jak z decyzją o małżeństwie. Atrakcyjną „ofertę" często boleśnie weryfikuje dopiero życie, a umowa umową – nie ma odwrotu.

Po rozpoczęciu realizacji projektu klient nie zapomina o swojej wiodącej roli. Przekonany o konieczności doglądania przedsięwzięcia kontroluje wszystko osobiście, tworzy nowe wymagania, zmienia koncepcje realizacyjne, stawia veto, wstrzymuje wypłaty faktur itd. Wszystko w trosce o swoje cele i szczęście. Niestety staje się wtedy tyranem narzucającym reguły gry, w którą sam nie potrafi grać. Takie przedsięwzięcie z informatycznego zmienia się w żałosną farsę. Wbudowany w dostawcę imperatyw „zadowolenia klienta za wszelką cenę" prowadzi do katastrofy. Podobnie jak bezstresowe wychowywanie dzieci prowadzi do chaosu i braku kontroli nad ich poczynaniami, obracając się w rezultacie przeciwko nim.

Spełnienie wymagań, zadowolenie klienta – tak. Zgoda na wszystko, za wszelką cenę – nie. Analizy spektakularnych niepowodzeń informatycznych pokazują, że wszystkie te katastrofy były precyzyjnie od początku zaplanowane. Ktoś zgodził się zrealizować intratny kontrakt o połowę krócej niż konkurencja. Zażądał przy tym połowę ceny oferowanej przez konkurencję, bo dysponował tylko połową środków potrzebnych do realizacji. Klasyczny „marsz śmierci". Dlaczego? Bo zabrakło odważnego, który powiedziałby: „Nie zgadzam się. Król jest nagi. Cena przy kosztorysie

wykonanym zgodnie z regułami sztuki jest dwa razy większa. Jednakże zaakceptowanie takiego kosztorysu jest gwarancją, że podjęte przez nas działania zakończą się sukcesem, choć znacznie później niż tego oczekujesz. W ten sposób zapewniamy ci bezpieczeństwo inwestycji i wysoką jakość produktu końcowego. Prosimy o uważne i rzetelne rozważenie naszego kosztorysu. Jeśli mimo wszystko nie zaakceptujesz go, będziemy musieli odstąpić od umowy, ponieważ nie możemy zagwarantować ci satysfakcji za mniejsze pieniądze i w krótszym czasie".

„Po pierwsze nie szkodzić" – to główna zasada przysięgi Hipokratesa, kanonu etycznego lekarzy. Informatyk także nie może szkodzić swojemu klientowi. To my jesteśmy odpowiedzialni za to, by chronić klienta przed samobójczymi przedsięwzięciami informatycznymi i nie przykładać do nich ręki. Profesjonalizm to nie tylko wiedza o tym, co należy robić, ale także o tym, czego i kiedy nie należy robić.

Pracownik i pracodawca

Podstawowe zasady etyki w tym wymiarze reguluje Kodeks Pracy. Wyznacza on minimalny zakres ochrony interesów pracownika oraz pracodawcy, w tym naczelną zasadę wykonywania powierzonych zadań z należytą starannością. Zasada ta jawi się jako podstawowa powinność każdego pracownika, której niewypełnienie może mieć skutki prawne. Co to oznacza w praktyce? Pracownik firmy przygotowuje ofertę dla klienta i umieszcza w niej kosztorys realizacji systemu informatycznego. Pracodawca weryfikując kosztorys stwierdza, że cena jest za mała i każe go poprawić. Co powinien zrobić pracownik? Pracodawca może też stwierdzić, że cena jest za duża. Co wtedy ma zrobić pracownik, aby postąpić z należytą starannością? Podobnych dylematów można znaleźć wiele. Najbardziej kontrowersyjną sytuacją jest taka, w której pracownik widzi, że pracodawca działa na szkodę klienta, np. zaniżając oszacowanie pracochłonności, rezygnując z wewnętrznej kontroli jakości, wyznaczając na kierownika projektu młodego i niedoświadczonego „znajomka". Gdzie jest granica lojalności względem pracodawcy, a gdzie zaczyna się uczciwość wobec klienta?

Pojęcie lojalności jest często wypaczane i trywializowane. W firmach wymaga się lojalności od pracowników po to, by zabezpieczyć interes firmy podczas konfliktów i negocjacji z klientem, wobec stróżów prawa, niezależnego audytora prześwietlającego nasze procesy organizacyjne itd. Lojalność w powszechnym rozumieniu to umiejętność nabrania „wody w usta" we właściwym momencie. Kto opanuje tę sztukę, zajdzie w firmie daleko. Lojalny jest ten, kto przymyka oko na mniejsze lub większe grzeszki pracodawcy. Ten, kto działa w interesie firmy, przedstawiając uczciwy plan projektu, nie jest lojalny. Jest naiwnym sabotażystą, a w najlepszym razie nieszkodliwym „człowiekiem z księżyca".

Problemy z lojalnością pracowników są nieusuwalne z organizacji. Organizacja jako całość realizuje cele, podczas gdy pracownik przede wszystkim kieruje się swoim własnym interesem i systemem wartości. Dlatego firmom zależy przede wszystkim na pracownikach przeciętnych, bez inicjatywy, wykonujących bez zastanowienia każde polecenie, elastycznych i konformistycznych, bez honoru i poczucia dumy z wykonywanego zawodu informatyka. Gdy firmie uda się pozyskać pracownika, który odpowiednio „umotywowany" utożsami swe życie z firmą, pozyskuje w swoim rozumieniu lojalnego sługę. Prosty zabieg wysłania pracownika na drogie szkolenie, za które będzie musiał sam zapłacić, jeśli odejdzie za wcześnie, zamienia sługę w niewolnika.

Wszystko pod pozorem inwestowania w pracownika, jego dobra, gwarantowania długotrwałego i stabilnego zatrudnienia itd.

W kodeksach etycznych zapis o lojalności, uczciwości wobec „braci" wykonujących ten sam zawód jest obowiązkowy. Jest to sposób budowania tożsamości zawodowej i konsolidacji środowiska. Ale lojalność nie jest wartością absolutną. Prawda i uczciwość ma pierwszeństwo przed lojalnością. To ona chroni grupę zawodową przed skostnieniem i zmową milczenia w obronie kłamstwa i nierzetelności w imię „interesu grupy". Czy w naszych firmach są jeszcze tacy pracownicy?

Zamiast podsumowania

Wspólnym mianownikiem opisującym większość powyższych relacji jest problem wyboru między interesem wspólnym a własnym. Doskonałą ilustracją tego znanego od zarania dziejów problemu jest tzw. *dylemat więźnia*. Spróbuję opowiedzieć go w jednej z bardziej znanych wersji z roku 1950. Dwóch znanych policji przestępców złapano w kradzionym aucie. Podejrzewa się ich o popełnienie dużo poważniejszego przestępstwa (zbrodni), na co jednak nie ma żadnych dowodów. Przestępcy są przesłuchiwani w osobnych celach bez możliwości komunikowania się ze sobą. Przesłuchujący ich komisarz wpada na pewien pomysł – proponuje im nagrodę za wsypanie wspólnika i dostarczenie dowodów na jego udział w zbrodni. Możliwe są trzy sytuacje:

- obaj milczą – wówczas obaj otrzymują niewielki wyrok za kradzież auta,
- jeden się przyznaje i wsypuje wspólnika – zdrajca wychodzi na wolność, jego kompan (tzw. frajer) otrzymuje wyrok za kradzież i za popełnienie zbrodni,
- obaj się przyznają i wsypują wspólnika – obaj otrzymują karę za kradzież i popełnienie zbrodni, nieco złagodzoną ze względu na współpracę z wymiarem sprawiedliwości.

W teorii gier zapisalibyśmy tę sytuację w postaci macierzy wypłat. Strategia C (cooperation) oznacza współpracę (między graczami), czyli milczenie. Strategia D (defection) oznacza zdradę, czyli wsypanie wspólnika. Oto nasza macierz:

	C	D
C	R, R	S, T
D	T, S	P, P

Znaczenie poszczególnych liter w macierzy gry jest następujące:

- R (reward) – nagroda za współpracę (czyli kara wyłącznie za kradzież auta),
- S (sucker's payoff) – kara za kradzież i zbrodnię (wypłata frajera),
- T (temptation to defect) – kara za kradzież auta złagodzona za wsypanie wspólnika (zachęta do zdrady),
- P (punishment) – kara za obopólną zdradę – czyli kara kradzież i zbrodnię złagodzona ze względu na współpracę z wymiarem sprawiedliwości.

Aby taką grę uznać za dylemat więźnia muszą być spełnione pewne zależności między wysokościami poszczególnych wypłat w macierzy: $T > R > P > S$. Wysokość poszczególnych wypłat jest nieistotna. Ważne, by spełnione były powyższe warunki.

Patrząc na macierz wypłat trudno ukryć zdziwienie prostotą tej gry. A przecież tkwi tu poważny haczyk... Nieprzypadkowo dylemat więźnia stał się podstawą wielu

badań prowadzonych w naukach społecznych. Z łatwością można zauważyć, że dla obu graczy strategia zdrady (**D**) jest dominująca. Prowadzi to do równowagi **DD** – a przecież na pierwszy rzut oka widać, że obaj lepiej wyszliby zgodnie współpracując ze sobą! W ten sposób doszliśmy do sedna sprawy – do konfliktu pomiędzy troską o własny zysk, a troską o „dobro grupy".

W przypadku rozgrywki jednorazowej praktycznie wszystko jest jasne – grający „na współpracę" jest frajerem. Jedynym rozsądnym wyjściem jest zdrada. Jest to sytuacja dosyć absurdalna, zważywszy na fakt, że powoduje ona przegraną obu graczy, w sytuacji, w której w prosty sposób mogliby zarobić (solidarnie współpracując).

Co się dzieje, gdy gracze mają grać ze sobą wielokrotnie, jak to często w życiu bywa? Po pierwsze zwróćmy uwagę na fakt, że w przypadku gdy gracze znają liczbę rozgrywanych gier, rozgrywka wielokrotna przestaje się różnić od jednokrotnej!

Rozgrywka wielokrotna ma sens jedynie wówczas, jeżeli gracze nie wiedzą, kiedy nastąpi jej koniec. Odpowiedź na pytanie o najkorzystniejszą strategię w grze w *iterowany dylemat więźnia* jest trudne. Zdradzić czy współpracować? A jeżeli napotkamy zdradę drugiego gracza, to jak zachować się w kolejnym spotkaniu? Odpłacić mu, czy też puścić płazem?

Omawiając ten problem nie sposób pominąć znanego doświadczenia Roberta Axelroda, który przeprowadził następującą próbę. Otóż poprosił on osoby interesujące się teorią gier o zaproponowanie różnych strategii gry w dylemat więźnia. W sumie zgromadził w ten sposób 63 graczy, a raczej 63 strategie – od prostych po bardzo skomplikowane. Następnie zorganizował turniej, w którym programy te grały przeciwko sobie. Turniej wygrała prosta strategia autorstwa kanadyjskiego politologa Anatola Rapoporta. Strategię ochrzczono mianem *Tit-for-Tat* (wet za wet), a jej schemat postępowania wygląda następująco:

- każdą rozgrywkę zaczynaj od współpracy,
- w następnych kolejkach wybieraj tę, którą drugi gracz zagrał w poprzedniej kolejce.

Po dokładnym omówieniu wyników, Axelrod zorganizował drugą rundę turnieju. Biorący w niej udział zawodnicy nadesłali kolejne programy czerpiąc z doświadczeń rundy poprzedniej i... „wet za wet" znów okazał się najlepszy! Axelrod przeprowadził dokładną analizę nadesłanych programów i doszedł do wniosku, że wśród plasujących się w czołówce większość wyróżniała się następującymi cechami:

- przyjazna – nie zdradza jako pierwsza,
- odwetowa – za zdradę karze zdradą,
- przebaczająca – po ukaraniu zdrady wyciąga rękę na zgodę,
- przejrzysta – jej decyzje powinny być spójne i łatwe do przewidzenia.

Przedstawione powyżej elementy teorii gier mają na celu jedynie pobudzenie wyobraźni czytelnika do samodzielnego poszukiwania właściwych strategii przetrwania w etycznym chaosie XXI wieku. Doświadczenia Axelroda i licznych jego naśladowców wskazują, że generalnie strategie uprzejme mają znaczącą przewagę nad strategiami „zdradzającymi". Oznacza to, że postawa współpracy w realnym świecie też długofalowo może się opłacać. Warto o tym pamiętać, negocjując kolejny kontrakt. Życie nie jest grą o sumie zerowej.

WDROŻENIA Z DZIEDZINY USŁUG MULTIMEDIALNYCH I ZASTOSOWAŃ CYFROWEGO PRZETWARZANIA SYGNAŁÓW

ANDRZEJ CZYŻEWSKI

Politechnika Gdańska
Katedra Systemów Multimedialnych

Streszczenie wystąpienia

Obecne możliwości technologiczne w dziedzinie komputeryzacji i sieciowej wymiany informacji wyprzedzają w wielu przypadkach istniejący rozwój koncepcyjny w zakresie sposobów realizacji i funkcji przekazu multimedialnego. W związku z tym, możliwości nowych technologii informatycznych nie zawsze są w pełni wykorzystywane a nawet niekiedy nie są należycie uświadamiane. Z tego względu przemysł informatyczny oraz firmy telekomunikacyjne starają się promować rozwój koncepcyjny multimediów, z którego wynikać powinny nowe koncepcje zastosowań technik multimedialnych (stosowane określenie anglojęzyczne: *new multimedia content*). Niestety, mecenat przemysłu teleinformatycznego wciąż niezwykle rzadko obejmuje swoim zasięgiem polskie zespoły naukowe. Tymczasem, wskutek upowszechnienia się tego typu technologii i stosunkowo dobrego poziomu kształcenia w ośrodkach akademickich, krajowe zespoły naukowe, zajmujące się tą dziedziną są w pełni przygotowane, aby skutecznie rozwijać badania i tworzyć nowe, użyteczne i oryginalne rozwiązania, które z łatwością znajdują możliwości szerokiego wykorzystania. Aby zilustrować tę tezę, wystarczy posłużyć się choćby przykładami z działalności zespołu Katedry Systemów Multimedialnych w ostatnich latach, choć należy mieć świadomość, że w Kraju istnieje także inne zespoły badawcze, które mają istotne osiągnięcia na tym polu.

Zainteresowanie technikami multimedialnymi wśród członków zespołu było naturalną konsekwencją wieloletniego rozwoju badań w dziedzinie inżynierii dźwięku i obrazu. Dziedzina ta, od początku jej wyodrębnienia się przed wielu laty, miała charakter interdyscyplinarny. Punktem zwrotnym w rozwoju prowadzonych badań było wprowadzenie do obszaru tej dyscypliny metod sztucznej inteligencji i tzw. soft computingu oraz sposobów rozwiązywania algorytmów trudnych obliczeniowo. Od tej pory następuje prawdziwy przełom, w rozwiązywaniu w naszym zespole wielu istotnych problemów z dziedziny zastosowań informatyki i technologii społeczeństwa informacyjnego. Wynika to bezpośrednio z natury rzeczywistych sygnałów fonicznych i wizyjnych, będącym ważnym składnikiem przekazu multimedialnego, zwłaszcza ich

dłuższych sekwencji, które są na ogół niepowtarzalne, trudne do ścisłego zdefiniowania, nie poddające się w pełni modelowaniu matematycznemu, powiązane ze sobą wzajemnie w sposób niejawny, odbierane i oceniane przez słuchacza lub widza w sposób niemożliwy do ścisłego zdefiniowania. Tymczasem, algorytmy przetwarzania danych i sygnałów osiągnęły w ostatnich latach tak wysoki stopień rozwoju, który zresztą odbywał się przy istotnym udziale polskiej myśli naukowej, że obecnie mogą być one z powodzeniem stosowane do pokonywania w pełni praktyczny sposób wielu problemów, które były zaliczane do trudnych lub niemożliwych do rozwiązania przy użyciu tradycyjnych metod obliczeniowych.

Przykładowo, jednym z bardziej istotnych problemów, wcześniej nie w pełni zadowalająco rozwiązywanych, jest rekonstruowanie nagrań fonicznych, zwłaszcza nagrań utworów muzycznych. Redukowanie szumu i zakłóceń opierało się tradycyjnie na filtracji adaptacyjnej i predykcji, a zatem na algorytmach opartych na uproszczonym, globalnym i na ogół odbiegającym od rzeczywistości modelowaniu sygnału muzycznego i zakłóceń, które to modelowanie nie uwzględnia preferencji estetycznych słuchaczy. Tymczasem, możliwość zastosowania algorytmów uczących się i prowadzenie ich treningu przez operatora, który odpowiednio dobiera przykłady dźwiękowe, biorąc pod uwagę subiektywnie ocenianą jakość dźwięku, pozwala na uwzględnienie subiektywnych preferencji słuchaczy w toku filtracji adaptacyjnej. Badania w tej dziedzinie, które rozwijano również w zespole Katedry Systemów Multimedialnych, zostały ostatnio poszerzone o elementy związane z modelowaniem procesów słyszenia. W ten sposób zostały opracowane oryginalne zastosowania perceptualnego kodowania dźwięku do redukowania szumu obecnego w sygnale źródłowym. Stanowi to rozszerzenie powszechnie stosowanych metod perceptualnej kompresji dźwięku w kierunku dokonywanych jednocześnie: kompresji i odszumiania sygnału. Procesem wyznaczania położenia tzw. masek perceptualnych przy tego typu kodowaniu sterują algorytmy uczące się. W najbliższym czasie planowana jest rozbudowa udostępnionego już w sieci Internet systemu rekonstruowania nagrań muzycznych oraz mowy, opartego na wykorzystaniu tej nowej, efektywnej metody redukowania szumu. W związku z tym, można uważać za w pełni uzasadnione kontynuowanie prac nad zorientowaną psychoakustycznie, zaawansowaną filtracją adaptacyjną, której efekty mają charakter zarówno poznawczy, jak i liczne aspekty aplikacyjne w telekomunikacji i informatyce. Wyniki uzyskane dotychczasowo w ramach tego zagadnienia badawczego zostały wdrożone m.in. w postaci systemu umożliwiającego rejestrację i rekonstruowanie sygnału mowy pilotów wojskowych (projekt celowy na zlecenie Wyższej Szkoły Oficerskiej Sił Powietrznych w Dęblinie). Opracowany system stanowi zestaw zaawansowanych narzędzi do rejestracji, analizy i rekonstruowania mowy, zrealizowany w formie oprogramowania komputerowego. Narzędzia te pozwalają na szybkie wyszukiwanie pożądanych fragmentów nagrań oraz poprawę ich zrozumiałości na drodze redukcji szumów, zniekształceń i zakłóceń. Ogólną strukturę systemu można przedstawić jako zbiór stacji bazowych (rejestratorów), dokonujących rejestracji korespondencji radiowej, komunikujących się za pośrednictwem serwera. Jest to rozproszony system rejestrujący korespondencję radiową, prowadzoną na określonym obszarze.

Opracowany system umożliwia rejestrację korespondencji radiowej, która jest prowadzona w określonych stacjach bazowych. Zarejestrowany materiał stanowi ma-

teriał źródłowy, wykorzystywany dalej w procesie analizy. Podczas analizy korespondencji silnie zakłóconej niezbędna jest dodatkowa obróbka sygnału, zmierzająca do poprawy zrozumiałości mowy. Do tego celu stworzono wyspecjalizowane narzędzie, udostępniające pakiet zaawansowanych algorytmów z dziedziny cyfrowego przetwarzania sygnałów. Aplikację tę nazwano *Rekonstruktor* – umożliwia ona rekonstruowanie zniekształconego sygnału mowy.

Sposobem wdrożenia wyników badań w tym zakresie na skalę ogólnoeuropejską jest pozyskany projekt zintegrowany FP6 pod nazwą PRESTOSPACE – Preservation towards storage and access. Standardised Practices for Audiovisual Contents in Europe. Projekt dotyczy opracowania i wdrożenia technologii archiwizacji, rekonstrukcji i dostępu do nagrań archiwalnych. Standaryzacja sposobów dostępu do archiwów foniczno-wizyjnych w Europie. Celem projektu jest opracowanie systemu archiwizacji i rekonstrukcji filmów i nagrań archiwalnych, który stałby się standardem europejskim w zakresie archiwizacji i rekonstrukcji zbiorów audiowizualnych. Zadaniem Katedry Systemów Multimedialnych PG jest opracowanie wybranych algorytmów rekonstruowania dźwięku.

Kolejnym przykładem wykorzystania inteligentnych algorytmów decyzyjnych i zaawansowanych technik obliczeniowych w multimediach jest *problem automatycznej identyfikacji dźwięków muzycznych, fraz muzycznych i obiektów w obrazie wizyjnym lub w grafice komputerowej*. Rozwiązywanie tych zagadnień ma trudne do przecenienia znaczenie dla rozwoju technologii multimedialnych baz danych. Obecnie można utwory muzyczne lub elementy obrazów cyfrowych wyszukiwać w takich bazach jedynie na podstawie ich opisu tekstowego, natomiast ze względu na lawinowy przyrost liczby nagrań dostępnych w sieciach komputerowych, bardzo pomocną funkcją staje się ich wyszukiwanie na podstawie określonych cech dźwięku oraz poprzez podawanie w procesie wyszukiwania fragmentu linii melodycznej lub obrazu poprzez definiowanie jego opisu parametrycznego. Waga tego zagadnienia jest już obecnie na tyle doceniana, że stworzeniem norm do tego celu zajął się komitet standaryzacyjny MPEG (*Motion Picture Engineering Group*), który opracowuje standardy dotyczące identyfikacji plików fonicznych i wizyjnych MPEG-7, MPEG-21 uwzględniające nie tylko ich opis tekstowy (metaopis, metadane), ale również parametryzację sygnałów, pod kątem jej wykorzystywania przez inteligentne narzędzia wyszukiwania. Rozwiązywanie tych zagadnień ma podstawowe znaczenie dla upowszechnienia interaktywnej radiofonii i telewizji.

Istotnym obszarem zastosowań nowych koncepcji z dziedziny multimediów są aplikacje medyczne i telemedyczne. We wspomnianej dziedzinie, techniki multimedialne oparte na odpowiedniej podbudowie naukowej, okazują się bezpośrednio przydatne lekarzom i pacjentom, a więc mają szczególne znaczenie z punktu widzenia społecznego. Tak się składa, że w związku z realizacją części projektów zamawianych Ministerstwa Zdrowia w Katedrze Systemów Multimedialnych (w ścisłej współpracy z warszawskim Instytutem Fizjologii i Patologii Słuchu), członkowie zespołu mieli okazję w ostatnich latach znacznie rozwinąć badania stosowane w tej dziedzinie. Rezultatem tego jest opracowanie i wdrożenie (dokonane za pośrednictwem Instytutu Fizjologii i Patologii Słuchu) szeregu oryginalnych rozwiązań multimedialnych systemów do badania słuchu, szumów usznych, wzroku, płynności wymowy (wraz z możliwością

korygowania niepłynnej wymowy na drodze elektronicznej i wspomaganego komputerowo treningu logopedycznego). Stworzenie wyżej wymienionych narzędzi, które na mocy decyzji Ministerstwa Zdrowia i Ministerstwa Edukacji Narodowej i Sportu są od pewnego czasu wprowadzane do szkól w celu ich wykorzystania w badaniach przesiewowych dzieci i młodzieży, wymagało rozwiązania wielu problemów koncepcyjnych. W oparciu o przeprowadzone badania naukowe udało się opracować nowe metody prowadzenia badań medycznych przy wykorzystaniu komputerów multimedialnych, które z powodzeniem zastępują kosztowne przyrządy diagnostyczne. Uzyskane wyniki zostały w 2000 r. uhonorowane Pierwszą Nagrodą Prezesa Rady Ministrów za wybitne krajowe osiągnięcia naukowo-techniczne oraz nagrodą Polskiego Klubu Biznesu w kategorii „Produkt Roku". Wdrożenie wyników oryginalnych badań naukowych w tym zakresie zostało także dostrzeżone za granicą poprzez nominację w konkursach „Stockholm Challenge Award" i „Europe's Best in Multimedia" a także zostało nagrodzone złotym medalem i dwoma pucharami Grand Prix na Światowej Wystawie Wynalazków w Brukseli w 2002 r., złotym medalem Międzynarodowych Targów Poznańskich w 2003 r., było zgłoszone (przez Fundację na rzecz Nauki Polskiej) do Nagrody Gospodarczej Prezydenta RP, również w 2003 r. Wymienione wcześniej osiągnięcia z zakresu zastosowań technik multimedialnych w medycynie zostały podsumowane przyznaniem Nagrody Przewodniczącego Komitetu Badań Naukowych w kategorii „info" w 2003 r. (jedna z trzech nagród przyznanych za osiągnięcia w działalności badawczo-wdrożeniowej). Najnowszym sukcesem w tej dziedzinie jest nominacja w maju 2004 r. tego projektu do konkursu na najlepszy projekt Europy w kategorii *e-Health*.

W związku z przystąpieniem do Unii Europejskiej powstaje obowiązek tworzenia dokładnych map hałasu. Polskie prawo przewiduje odpowiednie regulacje w tym zakresie już od kilku lat – na razie brak jednak jednolitych rozwiązań, które pozwalałyby na stosowanie tego prawa. Opracowywanie map hałasu w sposób tradycyjny jest kosztowne. Wymaga zatrudnienia specjalistów i zakupu urządzeń pomiarowych. Dlatego w zespole podjęto pracę nad dużo tańszym rozwiązaniem, którym jest *ogólnopolski multimedialny system monitorowania hałasu*, działający przy bezpośrednim wykorzystaniu komputerów, Internetu i telefonów komórkowych. Multimedialny System Monitorowania Hałasu, nad którym długofalowe prace rozpoczęto w 2003 r. pomyślany jest tak, by każdy użytkownik Internetu mógł zmierzyć poziom hałasu w miejscu, w którym mieszka lub pracuje i przesłać informacje do ogólnopolskiej bazy danych. Zamiast bardzo kosztownych urządzeń specjalistycznych, system wykorzystuje opracowaną prototypową przystawkę do komputera lub opracowane urządzenie pomiarowe mające możliwość transmisji danych poprzez sieć komórkową oraz oprogramowanie multimedialne dostępne na serwerach i na komputerach lokalnych.

Problematyka planowanego projektu badawczego odnosi się, zatem, do nowych rozwiązań koncepcyjnych i praktycznych, dotyczących profilaktyki i diagnostyki zagrożeń hałasowych na terenie całego kraju. W chwili obecnej nie ma w Polsce jednolitego rozwiązania z zakresu prowadzenia pomiarów hałasu, nie jest w kraju wytwarzany odpowiedni sprzęt i oprogramowanie do szacowania zagrożeń hałasem z uwzględnieniem nowego rodzaju wskaźników uciążliwości hałasu, zdefiniowanych w dyrektywie Unii Europejskiej, nie został wdrożony ujednolicony system tworzenia map akustycznych, nie zostały zdefiniowane procedury dostępu do danych na temat hałasu, ani

określone sposoby wykorzystania tego typu danych. Ponadto w metodyce prowadzenia pomiarów hałasu i przekazywania wyników tego typu pomiarów, nie zostały dotychczas wykorzystane na odpowiednią skalę zdobycze współczesnej informatyki i telekomunikacji, takie jak rosnące upowszechnienie komputerów multimedialnych i serwerów sieciowych, rozwój sieci komputerowych oraz możliwość prowadzenia pomiarów akustycznych oraz przekazywania danych za pomocą sieci komórkowych, wykorzystania satelitarnych systemów informacji przestrzennej, tworzenia baz danych wyposażonych w funkcje inteligentnego wyszukiwania i prezentowania informacji oraz przekrojowej syntezy danych dla celów planowania i podejmowania decyzji.

Pomimo dostępności wyników indywidualnych badań dotyczących szkodliwości oddziaływania hałasu na organizm ludzki, zwłaszcza na narząd słuchu, brak wiarygodnych danych tego rodzaju, odnoszących się do szerszej populacji, regionów oraz do rodzajów hałasu, który jest źródłem zagrożeń.

Tymczasem, w lipcu 2000 r. ukazała się końcowa wersja propozycji dyrektywy dotyczącej oceny i zarządzania hałasem środowiskowym: COM(2000)468. Ideą przewodnią tej dyrektywy jest opracowanie we wszystkich państwach członkowskich UE map akustycznych, według wspólnej metody, ujednoliconych wskaźników oceny hałasu, w sposób umożliwiający ich rozpowszechnianie i porównanie. Zalecenia dyrektywy będą wiążące w Polsce po ostatecznym przystąpieniu do UE, czyli w bliskiej perspektywie. Już obecnie, w ramach dostosowania prawa krajowego do standardów obowiązujących w UE, zapisy dotyczące wykonywania map akustycznych zostały uwzględnione w nowelizacji Ustawy o ochronie środowiska, gdzie wprowadza się obowiązek wykonywania map akustycznych co 5 lat. Mapa akustyczna ma zawierać wszystkie niezbędne informacje o stanie środowiska akustycznego oraz powinna stanowić podstawę do opracowania planów działań i strategii redukcji hałasu na poziomie lokalnym, narodowym oraz Unii Europejskiej.

Sprawom rosnących zagrożeń środowiskowych, związanych z hałasem, od pewnego czasu wiele uwagi poświęca Światowa Organizacja Zdrowia (WHO), według danych której obecnie ponad 20% mieszkańców państw należących do tej organizacji tj. 80 mln. osób jest narażonych w miejscu pracy lub zamieszkania na hałas o poziomie przekraczającym wartość uznawaną za bezpieczną dla słuchu i ogólnego stanu zdrowia. Nakazem chwili jest, zatem, podjęcie rodzimych działań w zakresie monitorowania hałasu i ich takie ukierunkowanie, aby rozwiązanie problemu ogólnopolskiego monitorowania zagrożeń hałasowych nie stało się wyłącznie przedmiotem importu myśli naukowo-technicznej i komercyjnych rozwiązań technologicznych. Byłoby to szczególnie niepożądane w sytuacji, w której rozwiązania tego typu są dopiero tworzone od podstaw za granicą.

Kolejny element programu badawczo-wdrożeniowego, który jest realizowany w Katedrze Systemów Multimedialnych dotyczy zagadnienia rozwoju koncepcji nowych usług w terminalach mobilnych. Badania realizowane wspólnie z Wydziałem Telekomunikacji Politechniki w Madrycie oraz firmą Ericsson doprowadziły do opracowania koncepcji nowego sposobu udostępniania usług w telefonii komórkowej, który to sposób wykorzystuje w większym stopniu zasoby obliczeniowe sieci komputerowych w miejsce zasobów terminali mobilnych i jest w większym stopniu, niż dotychczas niezależny od konkretnej platformy i związanego z nią systemu operacyjnego. W związku

z planowanym wykorzystaniem tej koncepcji w praktyce, zdefiniowano nowe usługi, takie jak tłumaczenie tekstu skanowanego przez kamerę terminala mobilnego, czy wykorzystanie terminala do osobistej dozymetrii narażeń hałasowych. Ponadto planowane jest w sieciach komórkowych wprowadzenie personifikacji ustawień sposobu przetwarzania mowy, dzięki czemu można będzie poprawić warunki komunikowania się przez telefon osób o słabym słuchu i wiele innych.

Przytoczone przykłady realizowanych wdrożeń miały za zadnie zilustrować zakres opracowań z dziedziny usług multimedialnych i zastosowań cyfrowego przetwarzania sygnałów, realizowanych aktualnie w zespole Katedry Systemów Multimedialnych Politechniki Gdańskiej. Zakres nadal niewykorzystanych możliwości z tej dziedziny pozostaje ogromny i stwarza pole do działania licznych zespołów twórczych w kraju i za granicą.

PO PIERWSZE – NIE MOTYWUJ!
jak każdy rodzaj motywowania bezpośrednio prowadzi do demotywacji

HELMUT V. GLÄSER

IMV Hamburg
imv.maus@t-online.de

Założeniem, z którego wychodzimy, jest krytyczne spojrzenie na „techniki" motywowania i sprawdzenie, czy i w jakim stopniu mogą one pomagać firmie w osiąganiu jej celów. Będziemy rozpatrywali takie aspekty wpływające na wytwarzanie zysku przez przedsiębiorstwa, jak:

- wydajność, rentowność,
- fluktuację (rotację kadr), wewnętrzne wypowiedzenie,
- jakość pracy, kreatywność.

Wykład podzielony jest na trzy części:

1. Pułapka motywacji:
   - o błędnych założeniach, oczekiwaniach i zachowaniach związanych z motywowaniem.
2. Łańcuch dewaluacji:
   - jak wyglądają systemy motywacyjne w praktyce?
3. Kierowanie dzisiaj:
   - jeżeli nie motywowanie, to co jest ważne?

Pułapka motywowania

Większość zatrudnionych skarży się, że jakość życia jest niska. Polega to na tym, że spędzając 2/3 swojego życia w miejscu pracy, marzą o czymś, co tą pracą nie jest, np. o wygranej w Toto-Lotka, która pozwoli im zacząć „nowe" życie.

Taka postawa zatrudnionych jest doskonale znana kierownictwom firm. Na brak motywacji pracowników reagują one najczęściej wprowadzając systemy motywowania.

Tymczasem to właśnie wspomniane przed chwilą i stosowane powszechnie systemy wyzwalają w pracownikach brak zainteresowania pracą i ucieczkę w marzenia o innym, nowym życiu:

- skutecznie zabijają motywację,

- uniemożliwiają efektywne kierowanie firmą,
- są drogie,
- prowadzą do stresu,
- stają się przyczyną choroby, na którą miały stanowić lekarstwo.

Aby uzasadnić, dlaczego tak się dzieje, przyjrzyjmy się, skąd czerpią swoje źródło systemy motywowania.

Otóż polegają one na *taylorystycznym obrazie organizacji*. Firma postrzegana jako maszyna, nie działa jednak jak maszyna. Stąd pochodzi błędne założenie, iż istnieje jakaś *motywacyjna luka* w zatrudnionych ludziach. Chcąc ją zapełnić, spowodować odpowiednie zaangażowanie pracowników, organizacje tworzą systemy, w których (świadomie?) *myli się motywację z motywowaniem.*

Inaczej mówiąc, źródeł tworzenia systemów motywacyjnych możemy szukać w sposobie postrzegania pracowników przez kierownictwa firm. Czyż systemy motywacyjne nie wynikają z założenia, iż *ludzie nie robią dobrowolnie tego, co powinni*? Zdają się mówić:

DEFINICJA 1: *Człowiek ma przyrodzoną niechęć do pracy i gdzie może, unika jej. Szuka satysfakcji osiąganej bez wysiłku. Dlatego trzeba go przymusem, groźbą kary i kontrolą doprowadzić do tego, aby przyczynił się do osiągnięcia przez organizację jej celów.*

Powyższa definicja nie brzmi przyjaźnie dla ucha. Posłuchajmy więc innej:

DEFINICJA 2: *Fizyczny i umysłowy wysiłek w pracy są dla człowieka tak samo naturalne, jak zabawa czy sen. Kiedy człowiek widzi sens w swojej pracy i zgadza się ona z jego celami, gotów jest z własnej inicjatywy ponosić wysiłek i poddawać się samokontroli. Nie tylko gotów jest przejmować odpowiedzialność, ale wręcz sam jej szuka. Człowiek z natury jest pomysłowy i kreatywny, jeśli tylko mu się na to pozwoli.*

Oczywiście każdy z nas, zapytany która z definicji mówi prawdę o nim, podpisałby się z pewnością pod tą drugą.

Dlaczego więc tak często kultura organizacyjna firm zdaje się mówić, że jesteśmy leniwi i trzeba nas poganiać? W jaki sposób taki obraz pracownika jest wpisywany w system wartości przedsiębiorstw?

Podejrzewanie jako kultura firmy – „motywacyjna luka"

Zakładając, że należy w przedsiębiorstwie stosować system motywacyjny, kierujący nią wysyłają następujący komunikat: *Podejrzewamy (albo obserwujemy), iż między rzeczywistym a możliwym zaangażowaniem istnieje luka. System motywacyjny ma tę lukę zapełnić.* Inaczej mówiąc: *Uzgodniliśmy w umowie z pracownikiem określoną wydajność na 100%. Jednak z góry zakładamy że będzie jej tylko np. 80%. Aby uzyskać całe 100%, musimy wobec niego stosować dodatkowe bodźce.*

Możemy też ten komunikat nieco przeredagować: *Wszyscy pracownicy to złodzieje! Umawiamy się z nimi na określoną wydajność. Płacimy im za to. Cóż jednak zrobić, abyśmy od nich owe uzgodnione 100% wydajności uzyskali? Musimy wprowadzić system motywacyjny!*

To oczywiście tylko interpretacje – obie jednakowo uprawnione. Za każdym razem komunikat brzmi bowiem tak samo: *Nasza firma oferuje pracownikom system motywacyjny.*

Jakie są skutki tak odczytanego komunikatu?

- jest samospełniającą się przepowiednią,
- buduje klimat organizacyjny na nieufności,
- nieufność ma tendencję do samoudowadniania się,
- powstaje błędne koło.

System motywowania jest więc zmetodyzowanym podejrzewaniem.

Badania przeprowadzone wśród menedżerów potwierdzają, że kultura podejrzewania jest typowa dla większości przedsiębiorstw. Wystarczy przytoczyć przykłady opinii z dwóch.

Ch. Freilinger – 400 menedżerów wyraziło opinię, że:

- nawet najbliżsi współpracownicy są leniwi i pracują tylko pod wpływem materialnych bodźców oraz kontroli,
- wydajność menedżerów (tzn. ich samych) jest wielokrotnie wyższa, niż ocenianych przez nich współpracowników.

Haire – przeprowadził ankietę z 3641 menedżerami z 14 krajów i stwierdził:

- leżący u podstaw kierowania brak zaufania do innych
- menedżerowie uważają, że wielokrotnie przewyższają swoich szefów w kreatywności, elastyczności i innowacyjności.

Z zacytowanych badań wynika, że zarządzający inaczej traktują siebie, a inaczej swoich podwładnych (*jakich to ja mam głupich ludzi w dziale*). Natomiast jako podwładni, uważają, że są lepsi od własnych szefów (*Stary się do wszystkiego wtrąca, a o niczym nie ma pojęcia*).

Punkt widzenia zależy więc od miejsca siedzenia? – Wracając do przytoczonych wcześniej określeń pracownika, w kulturze podejrzewania można mieć o sobie dobre mniemanie (definicja 2), ale uznawać za obiektywne stwierdzenie z definicji 1.

Motywacja – motywowanie

Angielskie słowo „motivation" możemy przetłumaczyć zarówno jako polskie „motywacja" oraz jako „motywowanie". Zwróćmy uwagę na tę różnicę:

1. Motywacja w pierwszym znaczeniu stanowi obiekt zainteresowania psychologii. Zajmuje się wyjaśnianiem zachowań, bada w nas i wokół nas to, co nas skłania, aby tak a nie inaczej się zachować. Stawia pytanie: „Dlaczego?"
2. Motywowanie to obszar zainteresowań menedżera. Rozumie on motywowanie jako tworzenie, podtrzymywanie i podwyższanie gotowości do działania u zatrudnionych przez przełożonego lub bodźce. Menedżer pyta: „Jak?"

W pierwszym przypadku motywacja jest absolutnie indywidualna – określa to, co stanowi o samosterowaniu jednostki i tylko jej dotyczy. W drugim przypadku mamy do czynienia ze świadomym działaniem kierującego albo z systemem bodźców, a więc ze sterowaniem z zewnątrz. Zwróćmy uwagę, jak blisko *od motywowania do manipulowania.*

W literaturze menedżerskiej motywowanie przedstawiane jest jako jeden z najważniejszych aspektów kierowania. W tym sensie motywować oznacza:

- wyposażyć kogoś w motywację, której wcześniej nie posiadał,

- rozpoznać czyjeś motywacje i zaoferować mu odpowiednie możliwości niezbędne do ich zrealizowania,
- nadać pewnym zachowaniom ważność,
- rozpalać entuzjazm, dopingować,
- dostarczać odpowiednich bodźców.

Podstawowym niebezpieczeństwem dla tak myślącego menedżera jest mylenie celu z drogą. Utożsamiając nastawienie pracownika ze świadomym działaniem kierującego stwarza bowiem wrażenie, że wystarczy jedynie coś pobudzającego zrobić, by u pracownika powstała automatycznie motywacja.

Jakichkolwiek nie używalibyśmy eufemizmów, ta manipulacja pozostanie przykro odbieraną manipulacją. Nawet jeśli menedżerowie mylą motywację z motywowaniem, zatrudnieni czują różnicę.

Motywowanie jako sterowanie z zewnątrz również odbierają jako manipulację.

Łańcuch dewaluacji – systemy motywacyjne w praktyce

Ktoś mógłby spróbować wybiegu i działać wedle zasady: *Motywowanie to umiejętność skłonienia człowieka, żeby robił to, czego ja chcę, kiedy ja chcę i jak ja chcę – i to w taki sposób, iż ma on wrażenie, że sam tak chce.*

Ważne, żebym przedtem dokładnie wiedział o najskrytszych życzeniach, potrzebach i słabych stronach swojej ofiary. W końcu żadna organizacja nie może istnieć bez kierowania, więc może spróbujmy motypulacji (motywowanie + manipulacja).

Aby zamknąć „motywacyjną lukę" stosuje się zazwyczaj kilka strategii:

1. Strategia „przemoc"

Wyraża się w postawie: *Rób, co ci każę, bo inaczej zostaniesz ukarany. Jeżeli będziesz działał, jak ja chcę, wtedy nic ci się nie stanie.* Instrumentami przy stosowaniu tej strategii są np.: blokowanie urlopu, awansu, zwolnienie...

System taki funkcjonuje tylko wtedy, gdy zatrudniony nie ma możliwości ucieczki i gdy jego pracy nie da się dokładnie mierzyć.

2. Strategia „przynęta"

Wyraża się w postawie: *Rób, co ci każę, bo inaczej zaszkodzisz sobie! Wysil się, to dostaniesz, co ci się należy.* Instrumentem jest tu system „negatywnych bonusów".

Z punktu widzenia użytkowników wadami tego rozwiązania jest to, że w systemie bonusów zatrudniony może oszukiwać, nie wszyscy są tak samo wrażliwi na ten sam bodziec, a praca nie daje się tak łatwo kwantyfikować. Dodatkowo strategia ta prowadzi do powstawania niesprawiedliwości.

3. Strategia „nagrody"

Postawa stosującego ją polega na wysyłaniu komunikatu: *Rób co ci każę i to chętnie.* Pozytywne sformułowanie: *Jeśli będziesz miał właściwe podejście, otrzymasz nagrodę.* Jako instrumenty wykorzystuje się tu premie, bonusy, systemy kafeteryjne, incentives.

Ponieważ jest to najbardziej propagowana strategia motywowania – wiele artykułów mających w założeniu popieranie humanizacji zarządzania podkreśla wyższość „marchewki" nad „batem" – warto prześledzić, jak działa i jakie przynosi realne efekty.

Jako przykład weźmy incentives, których popularność na Zachodzie osiągnęła swoiste apogeum. Niezwykłe nagrody stały się codziennością wielu przedsiębiorstw, dlatego coraz trudniej znaleźć coś nowego, co byłoby dla wyróżnionych atrakcją. Podczas targów NIPS w Chicago w 1997r. znalazły się m.in. takie oferty, jak budowanie szałasów na Bahamach, gaszenie pożarów na Manhattanie, kurs prawa jazdy na tramwaj w San Francisco, obiad z kanibalami na wyspach Fidżi, śniadanie w hotelu w Singapurze w towarzystwie żywego goryla, pływanie w basenie z krokodylami, a następnie przyrządzanie szaszłyków z ich mięsa (Australia).

W Niemczech spotkałem się też z anty-incentives. Wyróżnieniem, tym razem dla tych pracowników, którzy nie odnieśli sukcesu była wycieczka rowerowa w deszczowy listopadowy dzień.

Jeśli chodzi o efekty, najbardziej wymierne polegają na zwiększaniu obrotów w branży turystycznej.

W przypadku firm z pewnością incentives prowadzą do coraz większych kosztów, bowiem nagrody szybko stają się oczekiwaną częścią wynagrodzenia.

Natomiast u zatrudnionych powodują podwyższenie ich oczekiwań, nieuniknionych przypadków niesprawiedliwości.

Co ciekawe, nagrody otrzymują ci, którym nie są one potrzebne – najlepiej bowiem pracują ci, którzy mają wewnętrzną motywację. Ci, którzy ich nie dostają, są przez to zniechęceni. A takich jest wielokrotnie więcej!

Także kierownicy tracą w wyniku strategii „nagrody" część swoich kompetencji – skoro mają system motywacyjny, traktują go jako wyrękę. Organizacja „kieruje się sama", albo kierują nią systemy.

Najpoważniejszą jednak konsekwencją stosowania systemów nagród jest zabijanie przez nie motywacji. Wtedy rzeczywiście powstaje „motywacyjna luka". Im bowiem silniejszy jest bodziec (np. incentives), tym bardziej przyćmiewa indywidualne, wewnętrzne zaangażowanie.

Człowiek zaczyna gonić za nagrodą, a nie za rezultatem pracy. Najczęściej ze szkodą dla tego rezultatu.

4. Strategia „wizja"

Komunikat przekazywany zatrudnionym ma ich przekonać, że ich organizacja jest wspaniała, najlepsza, z dużym rozmachem wprowadza w życie śmiałą wizję i dlatego trzeba wyzwolić w sobie entuzjazm dla wykonywanych działań.

5. Pull, no push – nie popychaj, pociągnij

Narzędziem tej strategii jest wewnętrzny public relations ukierunkowany na tworzenie wyidealizowanego obrazu firmy w umysłach zatrudnionych.

Skutki takiej strategii mogą okazać się tragiczne przede wszystkim dla samej organizacji. Traci ona bowiem zdolność do samokrytyki, gdy zaczyna pobożne życzenia świetlanej przyszłości stworzone na potrzeby motywowania brać za rzeczywistość. Sztucznie narzucona wizja nie idzie w parze ze zmianą kultury firmy – więc obok pobożnych życzeń powracają w kierowaniu strategie gróźb, kar czy obietnice nagród.

Praktyka stosowania powyższych strategii – osobno czy łącznie – związana jest z pasywnością kierowników. To nie kierownik mówi pracownikowi *jestem niezadowolony z twojej pracy*, tylko system premiowania przez brak bonusu albo karę.

Menedżer przenosi swoje zadania przewodzenia ludziom na system motywowania. Także swoją odpowiedzialność. W przypadku błędów wina będzie przypisana systemowi. Za błędy odpowiada zamiast menedżera albo „zły" system motywowania, albo dział HR, który go stworzył. Dla menedżera bardzo wygodna sytuacja. Jednocześnie menedżer traci swoją rację bycia jako przywódca!

Tak naprawdę systemy motywowania odzwierciedlają klęskę menedżerów w prowadzeniu ludzi.

Kierowanie dzisiaj – jeśli nie motywowanie, to co?

Każda efektywna praca zależy od trzech podstawowych czynników:

(1) gotowości do działania,
(2) zdolności do działania,
(3) możliwości działania.

Pierwszy można by wiązać z wewnętrzną motywacją jednostki, drugi z posiadanymi przez nią kompetencjami (umiejętnościami i wiedzą). Chociaż oba są indywidualne, wsparcie ze strony kierownika odgrywa niemałą rolę. W przypadku możliwości działania udział menedżerów w jej zapewnieniu jest fundamentalna.

Jakie zadania dla przełożonych można wymienić w kontekście tych czynników? Powinni oni:

- stawiać wymagania,
- uzgadniać wyniki (konkretyzować oczekiwania),
- kontrolować ich realizację,
- otwarcie rozmawiać z podwładnym o wynikach tej kontroli,
- zapewniać sens wykonywanej pracy,
- zapobiegać demotywacji,
- dzielić się władzą (uprawniać).

Ponadto do kluczowych zadań menedżera należy kwestionowanie aktualnego stanu w firmie, postawa ukierunkowana na przyszły rozwój i poprawę organizacji.

Do ich realizacji wszyscy w funkcjach kierowniczych potrzebują wiele umiejętności osobistych i interpersonalnych.

Wśród nich najliczniejsze związane są z profesjonalnym komunikowaniem się (słuchanie, pytanie, argumentowanie, negocjowanie, dyskutowanie, moderowanie, prezentowanie, przeprowadzanie zebrań).

Większość menadżerów to niestety komunikatywne kaleki. Często żyją w obawie, co by się stało gdyby im zabrano „kule" premiowania. A najgorzej gdyby tak odebrać „premie uznaniowe".

Niezbędna jest także znajomość psychologii grup. Znajomość psychologii pomaga też w wybieraniu współpracowników, umiejętnym wykorzystywaniu konfliktów, reagowania na zachowania nie-fair.

Inne łączą się z zarządzaniem (ustalanie celów, uzgadnianie wyników, kontrolowanie, ocena, organizowanie czasu i struktur, delegowanie i zapobieganie redelegacji), uczeniem (przełożony jako trener i coach swoich ludzi).

Nie można też zapominać o typowych umiejętnościach przedsiębiorczych, jak np. podejmowanie decyzji, ryzyka, systematyczne rozwiązywanie problemów, innowacyjność i kreatywność.

Kluczową umiejętnością jest moderowanie sytuacji konfliktowych. Moderowanie oznacza, że to nie przełożony rozstrzyga sytuację konfliktową, karząc jako sędzia „złych" i „nagradzając" dobrych, tylko że odpowiedzialny team sam rozwiązuje sytuację konfliktową, szukając nowego (w danym momencie najlepszego rozwiązania).

Najlepszego, czyli maksymalnie przybliżającego do uzgodnionego celu. Myślenie w tej kategorii jest podstawą zarządzania według doktryny jakości. Zarządzania w systemach procesowych.

Zarządzanie według doktryny jakości to nie nowa „technika" zarządzania, to całkowita zmiana filozofii, nastawienia do zarządzania.

Zarządzanie według doktryny jakości to: szukanie, wykrywanie, dopuszczanie i wręcz prowokowanie sytuacji konfliktowych oraz ich owocne dla wszystkich (zarówno w firmie jak i dla klientów) rozwiązywanie.

Konflikt to nasza najbardziej twórcza forma komunikowania się. Tak czy siak zużywamy na nią największą ilość czasu. Rozwój jest możliwy tylko przy owocnym wykorzystywaniu (a nie unikaniu) sytuacji konfliktowych.

Każde odchylenie od standardu to sytuacja konfliktowa. Mówiąc o konfliktach, większość myli sytuację konfliktową ze sposobem jej rozwiązywania.

Zarządzanie według doktryny jakości, to owocne wykorzystywanie konfliktów w zarządzaniu. Wymaga ono innych, nowych umiejętności w zarządzaniu, a przede wszystkim zmiany nastawienia do konfliktu.

Mogę postrzegać zgłaszającego reklamację klienta jako kogoś, z kim muszę sobie poradzić („radzenie" sobie w sytuacjach konfliktowych), a tak naprawdę, to *klient zgłaszający reklamację jest bezpłatnym doradcą*, który sygnalizuje mi możliwość polepszenia mojej pozycji na rynku. Z tego powodu w centrum zainteresowania w zarządzaniu według doktryny jakości stoi zadowolenie klienta (zewnętrznego i wewnętrznego). Ja uważam, że *klient zgłaszający reklamację jest cenniejszy niż klient zadowolony* i z tego powodu życzę Państwu wielu reklamujących klientów!

LIFE-INSPIRED SYSTEMS: WHY AND HOW?

LECH JÓŹWIAK

Eindhoven University of Technology,
EH 9.24, PO Box 513, 5600
MB Eindhoven, The Netherlands
L.Jozwiak@tue.nl

Abstract: The recent spectacular progress in modern microelectronics and information technology that enabled implementation of a complete complex information processing system on a single chip, global networking and wire-less communication created new important opportunities. The traditional applications can be served much better and numerous new sorts of systems are became technologically feasible and economically justified, especially for applications that require miniaturization, high performance, low power dissipation, and wire-less or distant communication. On the other hand, this revolutionary progress created new serious difficulties. These difficulties mainly result from the huge and rapidly growing silicon and system complexity, and from the remarkable and growing role of the information technology systems. Many previously ignorable phenomena have now a great impact on the system correctness and other quality aspects.

An adequate solution of these new problems can only be achieved through development and application of new more suitable system and design methodologies and means. This paper aims at a brief analysis of the situation, trends and problems in the field of the modern information technology systems, and discussion of a new system paradigm of the life-inspired systems.

Keywords: embedded systems, design methodology, life-inspired systems, silicon and system complexity, system on chip

1. Introduction

What system design is about is a definition of the required quality, in the sense of a satisfactory answer to the following two questions: *what (new or modified) quality is required?* and *how can it be achieved?*

Design, being a definition of the required quality, expresses the values, aims and quality of life of a particular society, and particular individuals or groups from the society, having a particular culture, technology, and functioning in a particular space, time, and conditions. What is perceived as quality depends therefore on social, cultural and other aspects, and in particular, on value systems and material culture, specifically: on technology.

Since societies, their value systems, cultures, and technologies evolve, the quality of the future is different than the quality of the past. The products of the past – that are mainly based on the matter and energy processing (as buildings,

objects of daily use, mechanical and electromechanical systems etc.) – are quite concrete and static objects exposing a quite simple behavior. In contrast, the systems of today and future – that are more and more based on the information processing – are much more ideal, virtual, intelligent and dynamic. They expose more and more complicated and intelligent behavior, and interact with their creators and users more like persons or subjects than objects. To ensure that these new systems will represent the actually required quality, their designers must well understand and appropriately exploit the rapidly changing information based technology, but they also must well understand and seriously account for the more and more complex and quickly evolving organization, culture, value systems and actual needs of the society.

In particular, the recent spectacular progress in modern microelectronics and information technology that enabled implementation of a complete complex information processing system on a single chip, global networking and wire-less communication, created new important opportunities, but also new serious difficulties. These difficulties mainly result from the huge and rapidly growing silicon and system complexity, and from the remarkable and growing role of the information technology systems. The adequate solution of these new problems can only be achieved through development and application of new more suitable system and design methodologies and means.

This paper aims at a brief analysis of the situation, trends and problems in the field of the modern information technology systems, and a proposal and discussion of a new system paradigm of the life-inspired systems.

2. Situation and trends in modern information technology and their consequences

Progress in microelectronic technology is extremely rapid:

- a new technology generation replaces the current one every two to three years,
- the circuit complexity grows three to four times from generation to generation.

Recent spectacular advances in microelectronic technology enabled implementation of a complete complex information processing system on a single chip (SoC) or in a single package (SiP). They also made possible the development of the global networking and mobile wire-less communication and fast progress in these areas.

The SoC and SiP technology, global networking and wire-less communication:

- opened new better ways to serving existing applications, and
- made technologically feasible and economically justified numerous new applications.

In particular, new sorts of systems are feasible and affordable for applications that require miniaturization, high performance, low power dissipation, and wire-less and/or distant communication. For instance, systems that process or transmit the measurement or control information in, to or from mobile objects or distant, poorly accessible or dangerous places, installations, machines or devices. Various diagnostic, measurement, control and stimulation systems that can be put on or embedded in such (mobile) objects, installations, machines or devices, or even implanted in human or animal body can serve as an example.

This created a big stimulus towards development of various kinds of application-specific embedded systems, including ambient intelligence, augmented reality, and ubiquitous, wearable or even implanted systems.

The world's first intelligent shoe of Adidas can serve as an example. In this shoe, an electromechanical construction (that can be compared to a muscle) and an application specific embedded processor (that can be compared to a brain) take care of the measurement of the current (soil and shoe user) conditions and of the subsequent support and damping adaptation of the shoe to the current situation.

An embedded system (unlike a stand-alone computer) is an inseparable part (i.e. intrinsic sub-system) of a certain larger system. It is built in this larger system and designed especially or adapted to serve a specific aim in this system. Usually, embedded systems are some measurement, supervisory, control or communication sub-systems of the larger systems they are built in. They can be characterized as reactive real-time systems. They are currently used and will be more and more extensively used in virtually all fields of human activity: at home (in every day appliances and home electronics), in the office (inside of the office automation equipment), hospital (inside of medical devices), in factories (in robots and machines), planes (as parts of various control and navigation systems), cars (in suspension, engine, brakes and doors controllers) etc.

Embedded systems play a remarkable role in today's life, but they are very difficult to develop and validate. Since they are application specific, their design is highly innovative. They must continuously communicate with their surroundings and appropriately react in real-time to the signals from the surrounding. They often are very complex: they implement a number of concurrent processes and complex communication, both between the processes and with their surroundings. They are heterogeneous, i.e. composed of various sub-systems, implemented with different combinations of various processors, co-processors, accelerators, converters, sensors, actuators, and embedded software of these various programmable sub-systems. They must satisfy application specific constraints and objectives related to such attributes as functional behavior, geometrical dimensions, computation speed, power dissipation, price etc. Moreover, many of them are used in critical applications that impose extremely high quality requirements (e.g. measurement, control, and other systems built in various machines, robots, assembly lines, planes, cars, telecommunication equipment, military systems, safety systems, medical instruments or human body).

Summing up, availability of SoC and SiP technology, global networking and wire-less communication, and further rapid progress in these technologies created a large discrepancy between what is possible and what is used nowadays. This discrepancy results in both:

- a very strong technology push, and
- a very strong market pull,

to develop new or modified products and services, and in consequence, results in the ongoing information technology and information-based systems revolution.

On the other hand however, the spectacular advances in semiconductor and information technology introduced unusual complexity:

- Silicon Complexity, in the sense of large numbers, diversity, small dimensions and huge density of devices and interconnects, huge length of interconnects, and new materials and mixed technologies,
- System Complexity, in the sense of a huge number of possible system states, large numbers and diversity of subsystems, and extremely complex interactions and interrelations between the subsystems.

Due to the Silicon Complexity, and especially:

- extremely high device densities,
- extremely small devices' physical dimensions,
- power supply reduction, and
- very high operating frequencies,

many previously ignorable phenomena have now a great impact on the system correctness and other quality aspects. This results in a number of new difficult to solve hardware issues, such as:

- power and energy crisis, and fluctuations in the on-chip power density distribution,
- exponential increase in leakage power,
- delay variation due to substrate coupling and cross-coupling,
- on-chip communication and synchronization problems,
- decreased reliability, due to numerous reasons (noise, interference, signal integrity problems, increased defect density, manufacturing process variability, gate insulator tunneling, joule heating, electromigration, single event upsets and transients etc.),
- decreased design predictability (due to similar and some extra reasons),
- manufacturability problems and decreased yield,
- high manufacturing (mask and probe card) non-recurrent engineering (NRE) and production costs, etc.

One of the important consequences is that the future chips will inherently be less reliable and more sensitive to noise and interferences with their environment.

The System Complexity also results in a number of system and design challenges:

- design of complex and highly heterogeneous systems (involving various mixtures of digital and analog hardware and software on a single chip) with exponentially growing number of states,
- ensuring quality and validation (simulation, formal verification, testing etc.) of such complex and heterogeneous systems with exploding number of states and components,
- ensuring the systems' responsiveness, reliability and safety in the light of changing, noisy and unreliable environment and interior,
- reducing the design productivity gap and shortening time-to market,
- reducing the design NRE costs.

More details and explanations can be found in [1, 2].

Moreover, the information technology (IT) and microelectronics play a more and more remarkable role in today's life. The microelectronic-based information technology systems have important applications in virtually all fields of human activity

and have a global influence on the performance of society. They are commonly used not only in technical systems, but in various organizations and social systems as well, including numerous governmental, public and business services (e.g. e-government, e-education, e-banking, e-finance, e-commerce, transportation and recreation, communication, entertainment, and multi-media services, etc.). The quality of organizations, products, processes and services is increasingly dependent on the application of IT and the quality of the applied information processing systems. IT plays a vital role in innovation of products, processes and services, enhances their quality, and decides their success more and more frequently. Technological means of IT, and in particular microelectronic hardware and software, have become a basic strategic material for advanced products, processes and services, and often are more important than all other materials.

The microelectronic-based systems are used more and more commonly in virtually all sorts of technical and social systems, in more and more important and demanding applications. They are even implanted in our bodies. Our life is to a higher and higher degree dependent on their adequate operation. Therefore, the individual and society expectations regarding the systems quality grow rapidly, but also the information technology systems become more and more often goal of various attacks and unauthorized manipulations by individuals and organizations.

In consequence, responsiveness, robustness and dependability of the IT systems, as well as, reliable and secure storage, transmission and processing of information are critical to governments, organizations, businesses and individuals all over the world. Numerous cases are known of the political, financial or personal losses suffered when governmental, corporate or personal data are stolen or misused, or systems are broken. Moreover, as explained above, the future chips will inherently be less reliable and more sensitive to noise and interferences with their environment. However, we can certainly not tolerate that the future systems will be less reliable. In contrary, since they will be used in more and more important and demanding applications, we will expect their higher reliability. Unfortunately, even the best laws, regulations, police and lawyers alone (although important) cannot protect us against criminal individuals or organizations, nor guarantee the responsiveness and dependability of systems under changing, noisy and unreliable environment and interior. The adequate organizational and technical solutions must be developed and used. Many solutions are available, but many of them are not effective enough in confrontation with a sophisticated attacker or new phenomena and problems resulting from the silicon and system complexity.

Moreover, many systems, and especially, many social systems, were not developed having in mind the (extensive and/or intrinsic) usage of IT, and therefore, they often cannot sufficiently profit from the IT application. Even worse, the application of the IT means may result in their worse functioning, or even corrupt their stability due to a drastic decrease of their inertia (as a result of the change of their time constants to close to zero comparing to the people reaction time, due to tremendous speed-up of information processing and transmission) and positive feedback. Technical systems are typically developed anew or re-designed to maximally profit from the IT application. The same should be done with the social systems, despite that this involves much larger inertia and costs in most cases. The future social systems should assume the

unavoidable and growing usage of the IT technology means in their basic infrastructure, and adequately account for the IT application effects. Similar ideas regarding the necessity of the social system redesign were formulated by Hiroto Yasuura [3] after an interesting discussion that I had with him on this topic during the Summer School on Application-Specific Microprocessor SoC, Chateau de Pizay, France, 8–12 July 2002.

Summing up, analogously to the co-development of the technical systems and the IT systems embedded in them or controlling them, the co-development of the social systems and the IT systems used in the social systems is necessary. Since many of the contemporary and future systems have/will have a mixed (social, technical or even biological) character, to solve many of the serious issues of the today's and future systems the total systemic solutions are necessary, involving adequately composed mixed social, technical and biological system sub-solutions. For numerous reasons explained above and further commented below, the co-development of the future mixed social, technical and/or biological systems should aim at the quality maximization of the total systemic solution, in the sense of its total effectiveness and efficiency, with a special focus on the stability, robustness, responsiveness, dependability, safety, security, adaptability, and validation aspects. Since all this sort systems involve people and/or influence people to a high degree, they should be developed taking into account not only the purely systemic, technological and economical aspects, but the social, psychological, philosophical, ethical, biological and other aspects as well. In this context, the environmental aspects, and specifically, the energy and material usage should also be seriously accounted for. The terms responsiveness, robustness, adaptability, safety, security and dependability are used here in the sense explained in [4].

The above mentioned important aspects are not new and many of them were already taken into consideration in the past. What is thus new or different now?

The new or different character of the current situation includes the following:

- due to the huge and growing silicon and system complexity, more and more demanding system applications, their growing importance and growing danger of attacks and manipulations, it will be more and more difficult to guarantee the system quality, and particularly, their responsiveness, dependability, robustness, safety, security and validation;
- due to the common usage of systems in variety of applications, in various kinds of social, technical and biological systems, the whole life on the Earth more and more depends on them; their quality, and specifically responsiveness, dependability, safety and security are critical to governments, organizations, businesses and individuals; therefore, the individual and society expectations regarding the system quality grow rapidly; also the social, psychological, philosophical, ethical, biological and other aspects become more and more important;
- in the past, the critical applications of systems were much limited to some space, flight, military, nuclear, medical and similar special applications; in the future, there will be more critical applications: the future systems will be used in virtually all sorts of critical applications, but also applications considered previously as non-critical will become more and more critical, because we more and more rely on them (e.g. (mobile) communication, networking).

Consequently, such system features, like high responsiveness, dependability, robustness, safety or security, that were considered in the past in relation to some special critical systems, must now become much more common, even in relation to the so called consumer applications (e.g. mobile phones or PCs). Since in the past only the special critical systems were equipped in these features, one could afford for their costly implementation. In the future systems, due to common application, reasonably low-cost solutions must be used. This means that these features cannot anymore be added on the top of the designed or even implemented system, when using simple, but expensive solutions (as for instance, the fault-tolerance implementation by system replication or testability implementation by adding some external circuitry), but all these features must be:

- accounted for from the very beginning of the system specification and design process,
- implemented using sophisticated effective and efficient solutions,
- considered in parallel with all other important system aspects to possibly share the implementation costs for various features (e.g. usage of the same redundancy technique for fault-tolerance and testing) and account for the consequences of their implementation for another required system characteristics (e.g. the consequences of usage of a particular safety or security enhancing technique on the power-dissipation, speed, area etc.), to allow this way for an adequate tradeoff exploitation and multi-objective optimization, and in consequence, to result in more coherent, compact, comprehensive, reliable, robust and lower-cost solutions.

Summing up, the transition:

- from the multi-chip systems to systems-on-a-single-chip,
- from the general-purpose stand-alone computers to application-specific embedded systems,
- from the separated systems to networked systems,
- from the wire-based communication to wireless communication,
- from the static systems to mobile and dynamic systems

is not a gradual change, but a real paradigm shift:

- it opens new possibilities,
- but it also creates new very serious difficulties

that cannot be adequately resolved without an adequate system and design methodology adaptation.

The adequate system and design paradigms to solve the problems seem to be the paradigms of:

- life-inspired systems, and
- quality-driven design.

Ten years ago, predicting the current situation, I proposed the paradigm and methodology of the quality-driven design of the microelectronic-based systems [5–7]. Together with my Ph. D. students and other collaborators, we applied the quality-driven design methodology to the architecture synthesis, multi-objective hardware/software partitioning and mapping of the generic platform-based hard real-time embedded SoCs [2, 8, 9], and to the multi-objective optimal circuit synthesis [10, 11].

Subsequently, I supplemented the paradigm of the quality-driven system design with the paradigm of the life-inspired systems that I am currently researching.

3. Life-inspired systems

The paradigm of the life-inspired systems originates from the observation that:

- the complexity, operation domains and roles of the microelectronics-based systems

more and more resemble

- the complexity, operation domains and roles of (parts of) the (intelligent) life organisms or organized populations of such organisms.

Based on this parallel, I formulated the hypothesis that: the future microelectronic-based systems should have characteristics that resemble the characteristics of (parts of) the (intelligent) life organisms or their organized populations.

Consequently, the basic concepts, principles, functional and structural organization etc. of the microelectronic-based systems should resemble these of the (intelligent) life organisms or their populations.

Since the microelectronics-based systems are used in various (often life-critical) applications in numerous social, technical and biological systems, including life organisms, the life of people, and also the whole life on the Earth, more and more depends on them. These systems must be thus life-inspired also for this reason.

Like a real organism, a life-inspired system should be a largely autonomous, self-contained, robust, self-organizing, self-adapting, self-regulating dynamic evolutionary system. Like a real organism or brain, it should be composed of largely autonomous, diverse, having their own particular aims and optimized for these aims sub-systems (organs or centers).

However, the sub-systems should be:

- adequately (hierarchically) organized,
- interconnected with an appropriate network of communication channels,
- properly coordinated and adequately collaborating with each other

to synergistically achieve the global system aims.

This is thus a very different view of the self-organizing, self-adapting, self-regulating evolutionary system than the simplistic view of the flat, homogeneous and structurally limited neural networks. It assumes a hierarchical, highly heterogeneous, and functionally and structurally complex system organization. It resembles much better the life organism organization, and accommodates as its special cases many recent microelectronic system concepts and architecture trends. Neural networks may of course be some specific sub-systems of the life-inspired system when appropriate.

Similarly to the real brain, the life-inspired system should:

- not limit itself to the traditional basic functions of an information technology system of collecting, transmitting, storing, processing, and presenting information in relation to some external systems, but in addition to these functions
- solve complex problems, take and implement difficult decisions, learn, discover new ideas, etc. when appropriate, also in relation to itself.

It has to achieve these diverse aims effectively and efficiently:

- in relation to complex applications, and

- in the light of changing, noisy and unreliable environment and own interior.

Therefore, the life-inspired system must:

- be highly decentralized and composed of largely autonomous, robust, dynamic and diverse sub-systems, highly specialized to particular aims,
- involve properly (hierarchically) organized and coordinated sub-systems, adequately collaborating and communicating using an appropriate network of communication channels,
- include adequate self-protection, self-testing, self-diagnosis, self-repair, fault-tolerance and other self-organization, adaptation and regulation mechanisms,

analogously to the life organisms.

Complex information processing is typically realized as a combination of simpler processing tasks that often can be performed simultaneously (in parallel). The particular (parallel or sequential) tasks can be performed either using some single powerful task-specific operators or composed of (often long) sequences of some „general-purpose" operators from some limited (close to minimal) functionally complete system of operators. It is clear that the traditional von Neuman like sequential computers, by using the close to minimal functionally complete system of operators and performing the processing basically sequentially in time, use a lot of time and energy comparing to the massively parallel systems that use the efficiently (in hardware) implemented task-specific operators. Moreover, the sequential character and CPU-centric processing results in memory and communication bottlenecks.

To avoid the memory and communication bottlenecks, processing time and energy inefficiency etc.:

- information, intelligence and computational resources of the life-inspired system should be properly distributed over all its sub-systems
- effective application-specific operators should be used, and
- parallel processing should be applied for tasks involving parallelism.

This requires:

- local distributed memories for the sub-systems,
- (more) global multi-port memories for sharing data and communication between the sub-systems,
- memory-centric processing for massive data – the computations must come to the data and not the data to the computations: re-configurable computing,
- (massively) parallel processing sub-systems involving application-specific in hardware implemented operators
- re-configurable hardware to implement the application-specific (parallel) processing and memory-centric processing effectively and efficiently.

Since re-configuration and generic system solutions will serve numerous important purposes in the life-inspired systems, they are discussed somewhat more precisely below.

4. Importance of the re-configurable and generic system solutions

In the life inspired-systems, re-configuration will play a very important role and serve numerous purposes, including the following:

- memory-centric processing for massive data,
- computation speedup, as well as, power and energy usage reduction in comparison to software solutions, due to the distributed processing and effective implementation of the application/program-specific operations and (massively parallel) computation patterns directly in the re-configurable hardware,
- computational resource sharing,
- flexibility, adaptability, and product differentiation in relation to applications (e.g. to cover large application domain, differentiate within a product family, react to changing standards, enhance, update or improve the product after fabrication, etc.)
- self-organization, self-regulation and adaptation in reaction to changing and/or unreliable surrounding and own interior, including reaction to changing operation conditions (e.g. adaptive control, filtering, interfacing, etc.), but also fault-tolerance, self-protection, self-testing, self-diagnosis and self-repair,
- development and fabrication effort re-use that results in the reduction of the design productivity gap, shortening of the time-to market, and reduction of the design and fabrication NRE costs.

Actually, what is re-configurable system? Since adequate definitions are very important for good understanding, let us focus for a while to adequately answer this question.

Many various definitions of re-configurable system can be found in literature, of which perhaps the most cited are the two following:

- programmable piece of silicon (S. Goldstein, Carnegie Melon Univ.), and
- device that computes by using post-fabricated spatial connections between compute elements (A. DeHon, UC Berkeley).

Unfortunately, all definitions that I was able to find in the literature, including the above two, are not general and precise enough to enable any serious discussion of the re-configurable systems. They only cover some particular (implementation) aspects of the re-configurable systems. Therefore, I proposed the following definition of re-configurable system: a system whose sub-systems and/or sub-system configurations can be changed or modified after fabrication (L. Jóźwiak, Eindhoven University of Technology).

Re-configuration can be implemented using various reconfiguration mechanisms and their supporting technologies. However, the re-configurable computing (RC) systems have recently become economically justified due to the recent progress in microelectronic technology that resulted in the availability of the high-density field re-programmable hardware devices (as FPGAs and EPLDs) and of the SoC technology. The high-density field re-programmable hardware devices, that can be temporarily (partly) customised for a specific application, program or part of a program, are used as the core component of a re-configurable system that implements various changing configurations. The SoC technology enabled both:

- embedded FPGAs, being parts of various SoCs, and
- reconfigurable SoC Platforms, being complete re-configurable systems composed of FPGAs/EPLDs/FPFAs, software-programmable processors, on chip

memories and various h/w accelerators (e.g. Xilinx' Virtex-II Pro^{TM}, Altera's Excalibur^{TM}).

RC systems composed of re-programmable hardware modules and software-programmable micro-processors (μPs) and digital signal processors (DSPs) take advantage of combined strengths of hardware and software at an acceptable cost. They enable effective and efficient reuse of development effort and silicon, eliminating some time-consuming and costly hardware design and fabrication phases and allowing for sharing the same piece of silicon for various hardware modules implementing different computations. RC systems make possible effective and efficient implementation of the in space distributed processing and the application/program-specific operations and (massively parallel) computation patterns directly in the re-configurable hardware, what results in a big computation speedup, as well as, power and energy saving in comparison to software solutions based on traditional CPU-based sequential processors. In particular, they enable effective exploitation of the bit-level and loop-level parallelism.

Re-configurable resources can be differently placed in the RC system and can play different roles in relation to the software-programmable processors of the RC system. In particular, the re-configurable part can be a Functional Unit (FU) inside a processor, Data and/or Control Path (DCP) segment inside a processor, DCP closely coupled to the processor, loosely coupled external DCP, co-processor, re-configurable/programmable fabric, re-configurable array of re-configurable/programmable processors, or re-configurable/programmable SoC platform.

RC systems can be classified in relation to:

- re-configuration granularity into the fine-grained (configuration at the level of individual bits) and coarse-grained (configuration at the level of words); the coarse-grained architectures require less configuration resources, but are less flexible;
- extent of reconfiguration into the fully and partially re-configurable;
- (re-)configuration method and time at which the (re-)configuration takes place into static: performed once prior to use of a system), or dynamic: performed during the use of a system, either prior to the run-time of a certain application or task or during the run-time of a certain application or task;
- placement and granularity of the data memory and configuration memory: internal or external (distributed, local or global) memories;
- topology, organization and programmability of interconnections (some of them are of cellular type with almost only local interconnections between the neighbouring logic blocks another implement global interconnects, e.g. linear arrays, mesh-based, crossbar based etc.)

Thus, many various re-configurable architectures are possible and there is no single ideal universal architecture: various re-configurable architectures are suitable to various applications involving different application-specific computation patterns and operators. Therefore, one of the main ideas supporting an effective and efficient re-configurable computing consists of generalization to a class of similar problems having similar computation patterns and involving similar operations, and development of a generic solution (re-configurable computation platform) for this class of problems

that is able to implement the required computation patterns and operations in a selected re-configurable hardware fabric.

Generic system solutions and architectures, not necessary related to re-configurable systems, and particularly the generic system platforms, can serve similar aims as re-configuration, and additionally, they enable the (semi-)automatic architecture synthesis, partitioning and mapping. If the problem at a hand belongs to a certain well-defined abstraction class containing problems with similar requirements, a general form of a solution can be developed for the whole class and reused. The general form constrains the solution search space to such a degree that the construction of particular solutions can be performed by well-defined mapping, search and decision process, based on evaluation of the solution's quality model. The mapping, search and decision processes can be automated to increase the designer's productivity and the overall quality of the design process. This sort of reuse is very important for the complex SoCs and SiPs. For a certain application field, an appropriate generic architecture template can be developed and the nature of the core system's modules and interfaces between the modules and with the external world can be defined. Some families of (programmable and/or (re-)configurable) core system modules and interfaces suitable for different sub-fields of the application field can be developed. For a particular application, the generic architecture template and some selected core system's modules and interfaces are reused and adequately instantiated to appropriately serve the application, but also some new problem-specific modules and interfaces may also be added. Summing up, to efficiently develop the complex life-inspired systems, an adequate mixture of the design reuse at the system and/or sub-system levels with the automatic synthesis from the system and/or sub-system levels is needed.

It should be observed, that the concept of generic system solution is strictly parallel to genotype in the life organisms:

- the generic solution is adequately instantiated to better serve a particular application,
- the genotype is mutated to better fit to particular conditions.

Also the system re-configuration is strictly parallel to the adaptation in life organisms.

5. Promises and challenges of the life-inspired systems

The life-inspired systems seem to be able to tackle the current and future problems of the emerging and future technologies and of the new demanding applications that made obsolete the traditional approaches. At the same time they will create some new technological, social and other challenges. For instance, the new systems should/will behave and interact with their surrounding, including people, more like life organisms or even like people.

To efficiently develop the new systems closely resembling the (intelligent) complex life organisms, new system architecture concepts and design methods and tools must be developed. However, the life-inspired systems are not just purely about technology, but more about the interaction between the nature (including human being) and the human civilization that, in particular, includes technology and technical systems. The life-inspired systems should be therefore considered taking

into account not only the technological and economical aspects, but the social, psychological, ethical and other questions as well.

Summing up, the life-inspired systems will require for their realization:

- largely autonomous, robust, dynamic and heterogeneous sub-systems, highly specialized to particular aims,
- distributed information, intelligence and computational resources,
- massively parallel multi-processor architectures involving efficiently, in (re-programmable) hardware, implemented task-specific operators,
- distributed memory, and for massive data – the memory centric re-configurable processing,
- appropriate network of communication channels,
- adequate self-organization, self-regulation and adaptation mechanisms, including fault-tolerance, self-protection, self-testing, self-diagnosis and self-repair,
- (re-)configuration that will serve multiple purposes,
- generic system platforms and other generic (architectural) solutions,
- an adequate mixture of the design reuse at the system and sub-system levels with automatic synthesis from the system or sub-system levels,
- new more effective and efficient design methods and EDA tools, based on the quality-driven design paradigm, involving the quality-driven multi-objective design decision making, and adequately addressing the new important design issues.

With the total quality-driven design approach [7], the responsiveness, robustness, dependability, safety, security, adaptability, validation and other important system attributes can be explicitly accounted for, given necessary attention, modeled, analyzed, and traded off against or combined with another aspects just from the beginning of the design process, instead of being added to the almost ready design as in the traditional design approaches. This will result not only in their better satisfaction, but in a more efficient design as well, due to the possibility of sharing some resources used for simultaneous implementation of different attributes, and better design space exploration, trade-off exploitation, and multi-objective optimization.

6. Conclusion

The system paradigm of the life-inspired systems seems to be adequate to tackle the current and future problems of the emerging and future technologies and of the new demanding applications. This paradigm combines many valuable concepts and approaches of the modern IT system engineering and explains what purposes these concepts and approaches can serve and why they are useful. It forms a solid base for the development of the future complex microelectronics-based systems. Its application in combination with the quality-driven design should result in higher quality of systems, due to the usage of systemic solutions more adequate in the context of the emerging and future technologies and new demanding applications, and due to a more coherent, systematic, effective and efficient design process. I hope that the discussion of the current situation, future trends and problems in the field of the modern microelectronic-based systems and of the paradigm of the life-inspired systems presented in this paper will suggest some interesting research subjects and

work directions important for the successful development and application of the future systems.

Acknowledgements

This paper has been prepared in the private capacity of Lech Jóźwiak, without using resources of the Eindhoven University of Technology.

References

[1] *The 2003 International Technology Roadmap for Semiconductors*, Semiconductor Industry Association, San Jose, CA, USA, 2003.

[2] L. Jóźwiak, S.A. Ong: *Quality-Driven Decision Making Methodology for System-Level Design*, 22nd EUROMICRO Conference, Prague, Czech Republic, September 2–5, 1996.

[3] H. Yasuura: *Towards the Digitally named World – Challenges for New Social Infrastructures Based on Information Technologies*, Keynote Speech, DSD 2003 – Euromicro Symposium on Digital System Design, September 1–6, 2003, Belek-Antalya, Turkey, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 17–22.

[4] L. Jóźwiak: *Life-inspired Systems, DSD 2004 – Euromicro Symposium on Digital System Design*, August 31st – September 3rd, 2004, Rennes, France, ISBN 0–7695–2004–0, IEEE Computer Society Press, Los Alamitos, CA, USA

[5] L. Jóźwiak: *Modern Concepts of Quality and Their Relationship to Design Reuse and Model Libraries, Current Issues in Electronic Modeling*, Chapter 8, Issue 5, Kluwer Academic Publishers, 1995.

[6] L. Jóźwiak: *Quality-Driven System-on-a-Chip Design, IEEE International Symposium on Quality of Electronic Design*, March 20–22, 2000, San Jose, California, USA, IEEE Computer Society Press, Los Alamitos, CA, USA, 2000.

[7] L. Jóźwiak: *Quality-driven Design in the System-on-a-Chip Era: Why and How?*, Journal of Systems Architecture, April 2001, ISSN 1383–7621/01165–6074, Elsevier Science, Amsterdam, The Netherlands, 2001, Vol 47/3–4, pp 201–224.

[8] S.A. Ong, L. Jóźwiak, K. Tiensyrja: *Interactive Co-design for Real-Time Embedded Control Systems*, Proc. ISIE-97, IEEE International Symposium on Industrial Electronics, Guinaraes, Portugal, July 7–11, 1997, IEEE Press, 1997.

[9] S.A. Ong, L. Jóźwiak, K. Tiensyrja: *Interactive Co-design for Real-Time Embedded Control Systems: Task Graphs Generation from SA/VHDL Models*, Proc. EUROMICRO-97, 23rd Conference „New Frontiers of Information Technology", Budapest, Hungary, Sept. 1–4, 1997, IEEE Computer Society Press, Los Alamitos, CA, USA, 1997, pp. 172–181.

[10] L. Jóźwiak, A. Chojnacki: *Effective and Efficient FPGA Synthesis through General Functional Decomposition*, Journal of Systems Architecture, ISSN 1383–7621, Elsevier Science, Amsterdam, The Netherlands, Vol. 49, No. 4–6, September 2003, pp. 247–265.

[11] L. Jóźwiak: *Advanced AI Search Techniques in Modern Digital Circuit Synthesis, Artificial Intelligence Review*, Kluwer Academic Publishers, Dordrecht, The Netherlands, Vol. 20, No 3–4, December 2003, pp. 269–318.

[12] A. Avizienis, J.-C. Laprie und B. Randell: *Fundamental Concepts of Dependability*, Research Report N01145, LAAS-CNRS, April 2001.

QUALITY OF SERVICE THROUGH INFRASTRUCTURE ON REAL-TIME DEMAND

SUKU KRISHNARAJ, MURTHY SRINIVAS, JACEK WALICKI

HP SGBU Management Software Organization

sukumar.krishnaraj@hp.com, murthy.srinivas@hp.com, jacek.walicki@hp.com

Abstract: This paper describes an integrated model-driven automation approach for delivering infrastructure on demand in an optimal fashion and thus achieving the necessary quality of service provided to customers.

1. Problem Statement

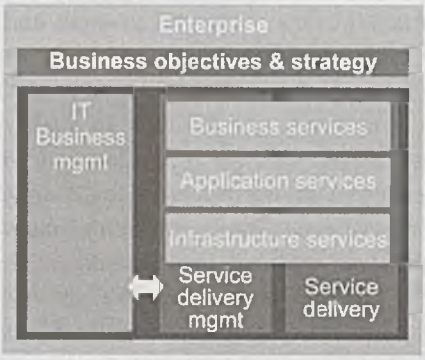
Our enterprise service-provider customers place an enormous emphasis on meeting, if not exceeding, the quality of service for their customers. Declining quality of service means declining customer satisfaction in addition to loss of revenue and market share. Acquiring additional compute capacity to resolve increased business demands is no longer an option (due to resulting increase not only in IT asset costs, but also in IT administrative costs). Additional resources are difficult to justify when untapped resources exist within the organization but are pre-allocated to critical applications in the event they may be needed. Manual processes for change management are time-consuming, expensive and error-prone. Hence, more and more customers are demanding the ability to leverage their existing investments in IT for addressing demand fluctuations in an optimal and timely fashion. We are not talking about an instantaneous change and adaptation. Necessary adjustments might actually take many hours. However, that needs to be contrasted with days and even weeks required when a human driven project is instituted to create a business driven change. Reliablity of action is also greatly increased through automated means that allow playing "what if" scenarios.

By building an infrastructure that will adapt itself in response to changing business needs and external factors, quality of service can be met effectively even in most demanding times while making effective use of existing IT resources. Although, we in HP have stand-alone products that provide capabilities for individual building blocks for an adaptive solution, an effective technical integration between them do not exist today thus preventing us from offering a holistic automation solution to our customers. This paper describes a technical approach for designing an adaptive IT closed-loop system – a model-driven, policy-based provisioning system – that provides automated provisioning of service based on real-time demand, in order to meet/exceed quality of service.

2. Approach

Infrastructure on demand warrants timely, automated (re-) provisioning of resources when demand for compute capacity doesn't match the current allocation – provisioning additional capacity for a service when demand exceeds supply and removing excess capacity when supply exceeds demand. To achieve these use cases, end-to-end management systems need to be in place. Service Delivery Controllers allow for a simple yet powerful way of orchestrating the various resources and management systems in the IT environment to achieve automated provisioning of a business service.

Several logical domains may be providing the infrastructure capabilities for end-to-end execution of a service (think 3-tier architecture) depending on how the service is designed. Examples of such domains are Oracle enterprise database, BEA application server layer, load balancing & network routing, etc. A service delivery controller (SDC) is responsible for controlling each such domain and operates on a standardized model of resources that it controls. SDCs have specialized knowledge of the domains they operate upon, and hence have the ability to maintain configuration integrity in case of change in the infrastructure. By designing a hierarchy of service delivery controllers, automation can be achieved throughout the life cycle management of a service (across different layers of the infrastructure), hence maintaining the quality of service provided.



Our approach follows architectural principles of Management for Adaptive Enterprise (Darwin Architecture) and it is based on the concept of a service delivery controller.

The Darwin architecture is the blueprint for the Adaptive Enterprise. It guides the evolution of your enterprise toward business agility one solution at a time, dynamically linking IT with business needs.

In Darwin, business processes, applications and the infrastructure are delivered as services throughout your organization.

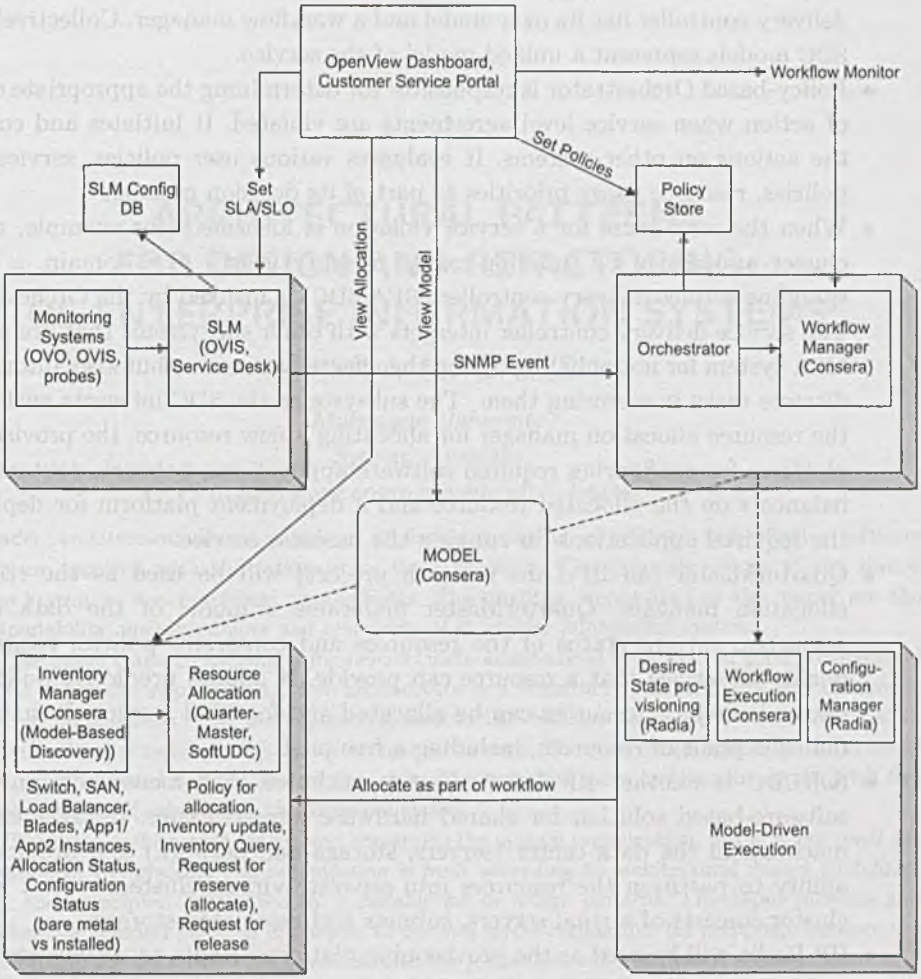
You model your enterprise from the top, starting with your business objectives and strategy, and at each level you apply the same design principles – simplification, standardization, modularity, and integration.

IT business management is the capability that creates an efficient supply-demand relationship between IT resources and business needs. When IT is run as a business, people, processes, and technology are integrated across the organization to deliver services at the business, application, and infrastructure levels.

Darwin is a service-oriented architecture that helps you tap the power of virtualization, where resources are pooled and shared. Model-driven automation helps keep IT practices simple and costs low. By using the Darwin architecture to guide your investments, you can balance agility with your need to increase performance, reduce risk and maximize financial return.

3. Solution Opportunity

The solution is achieved by integrating the following available products from HP:



- OpenView operations, OV Service Level Manager, OV Internet Services, OV Service Desk, and other management systems are used for assessing (monitoring) and advising the health of IT infrastructure. These tools allow us to set and monitor service level agreements (SLA). When an SLA is violated, the policy-based provisioning system is notified for further action. The protocols used for notification could be SNMP traps, web service notification protocol, or others.
- A centralized model is essential for accomplishing consistent change throughout the underlying infrastructure. The model is a description of a service, its associated configuration and constituent components that participate in successfully delivering the service capabilities. The business service is modeled (along with underlying dependencies and relationships) using the AgileOne model. Workflows for automation of tasks are developed by using AgileOne's workflow

designer and the model. Workflows capture the various tasks that need to be carried out in an automated fashion. By leveraging the model and driving operations through it, we ensure that the changes initiated in the environment are fully automated and apply holistically to all dependent resources. Each service delivery controller has its own model and a workflow manager. Collectively, the SDC models represent a unified model of the service.

- Policy-based Orchestrator is responsible for determining the appropriate course of action when service level agreements are violated. It initiates and controls the actions on other systems. It evaluates various user policies, service level policies, resource usage priorities as part of its decision making.
- When the root cause for a service violation is identified (for example, a BEA cluster appears to be suffering excess memory usage) in a domain, a corresponding service delivery controller (BEA SDC) is invoked by the Orchestrator. The service delivery controller interacts with other subsystems that are part of MBA system for not only identifying the affected resources, but also automating discrete tasks in restoring them. The subsystems the SDC interacts with are – the resource allocation manager for allocating a new resource, the provisioning platform for configuring required software applications, network, routers, load balancers on the allocated resource and a deployment platform for deploying the required applications for running the business service.
- QuarterMaster (an HP-Labs research project) will be used as the resource allocation manager. QuarterMaster maintains a model of the data center resources, current status of the resources and constraint policies relating to quality of service that a resource can provide. It applies predictive policies to determine which resources can be allocated and for what service. It maintains multiple pools of resources, including a free pool.
- SoftUDC is another HP Labs research initiative that focuses on providing software-based solution for shared hardware infrastructure. It takes existing machines in the data center (servers, storage and network) and provides the ability to partition the resources into separate virtual-clusters – each virtual cluster consists of virtual servers, subnets and associated storage.
- HP Radia will be used as the provisioning platform. Radia provisions resources at various levels – OS, patches, applications, and services, based on policies and desired state management. Radia applies differencing between actual configuration and ideal configuration as part of its desired state management, resulting in consistent replication of ideal configuration on resources. Radia provides ability to (re)configure hardware devices, network subnets, routers, load balancers and switches.

The interaction between various sub-systems is achieved by designing the integration layer using Service Oriented Architecture. Management Integration Platform (MIP) framework will allow sub-systems to exchange management events, share models and invoke actions based on policies through SOA interfaces.

ARCHITECTURAL PATTERNS FOR DESIGN AND REFACTORING OF ENTERPRISE INFORMATION SYSTEMS

LESZEK A. MACIASZEK

Macquarie University
Sydney, Australia
http://www.comp.mq.edu.au/~leszek/

Abstract: *Architectural design* is the set of decisions aiming at efficient and effective software architecture together with the rationale for these decisions. The rationale relates to the *system qualities* known as non-functional requirements. The qualities emphasized in this paper are the understandability, maintainability and scalability of enterprise information systems.

The paper starts by defining a *framework* (*meta-architecture*) in support of good architectural design for enterprise applications. Meta-architecture is a construction that guides and constrains the system organization – its decomposition, structuring decisions, and dependency firewalls. In object technology, a meta-architecture is a *reuse technology*. The technology applies to the reuse of the design, as opposed to the reuse of the code. The proposed meta-architecture works with five hierarchical layers of subsystems (packages) of classes.

The meta-architecture guides and constrains the system organization, but it is not itself the organization of the system. The organization is built according to architectural design guidelines, policies, and principles, legitimized by a suitable set of *design patterns*. The paper presents and exemplifies few selected patterns applicable to the two layers responsible for mappings between the application and the database – the *mediator* and the *foundation* subsystems.

The foundational knowledge for this paper is given in the author's latest book [8] http://www.comp.mq.edu.au/books/psc/index.htm.

1. Introduction

Software architecture can be defined as the organization of structural and behavioral components of the system together with statically visible relationships and externally visible interfaces of these components. The visibility aspect of an architecture influences many aspects of its quality. By making *relationships* between components "statically visible", a software engineer can understand dependencies between components. By making *interfaces* "externally visible", a software engineer can understand the services provided by components.

As software systems grow in complexity, the software developers have been learning valuable lessons about building complex enterprise information systems. As

observed in the seminal paper by Brooks [2], software is *complex by its essence*. The only thing that we can do is to ensure that we do not add to that inherent complexity through bad software development practices.

Software complexity has to be controlled and managed. Studies of the most complex systems that we can observe – the natural systems – reveal that all systems are layered *hierarchies of components* (Maciaszek, [6]). The lesson must be that any human-made systems must also be hierarchies, not networks. Only hierarchical structures can keep the complexity tractable and can evolve over time. It is the responsibility of a software architect to propose a system architecture that would keep complexity under control and would ensure the critical quality of the system known as *supportability* (Maciaszek and Liong, [7]). A supportable system has the properties of understandability, maintainability and scalability (evolve-ability).

Although intellectual freedom of software developers is an admirable thing, the truth of the matter is that the production of *supportable systems* (as opposed to the *legacy systems*) places undisputable demands for restricting this freedom. Large systems must be "architected" before they are built. Software architecture determines the *system qualities* (non-functional requirements), of which the *supportability* must be of overriding importance. The qualities define the general goodness of the system (as opposed to its fit-to-purpose).

There are two aspects to software architectural design. The first aspect is the definition of a framework (*meta-architecture*) that determines the *layers* of the (necessary) hierarchical structure and specifies allowed *dependencies* between the layers. The meta-architecture rules certain high-level implementation choices out and guides various selection decisions and trade-offs. The second aspect focuses on guiding and constraining low-level implementation choices by applying more specific policies and principles. These policies and principles are documented as *design patterns* (Gamma *et al.*, [5]; Fowler, [4]) and are indispensable to produce an architecture-compliant system.

This paper presents and exemplifies a small number of patterns applicable to the two layers responsible for mappings between the application and the database – the *mediator* and the *foundation* subsystems. The patterns are sourced from Fowler's *PEEA patterns* and relate to Gamma's *Gang of Four (GoF) patterns*. The relevant Fowler's patterns include Data Mapper, Identity Field, Identity Map, Unit of Work, Service Layer, and Lazy Load. The relevant GoF patterns include Façade, Abstract Factory, Chain of Responsibility, Observer, and Mediator.

The paper is structured as follows. The next section describes our meta-architecture called *PCMEF (Presentation-Control-Mediator-Entity-Foundation)*. Section 3 presents the principles that minimize *dependencies* between software objects in the PCMEF architecture. Section 4 discusses a couple of *patterns* for building PCMEF-compliant systems with the supportability quality. The Summary section points out that the *proactive approach* to architecture-driven development (as presented in this paper) must be supplemented with a *reactive approach* to uncover and rectify any architecture violations.

2. The PCMEF Meta-Architecture

There is no one unique or best meta-architecture that can result in supportable systems. The meta-architecture, which we advocate, is called *PCMEF* (Maciaszek and Liong, [8]). The *PCMEF* framework "layers" the software into five subsystems (in the UML sense) of classes – presentation, control, mediator, entity and foundation. It turns out that PCMEF is nicely aligned with the *Core J2EE tiers* (Alur *et al.*, [1]), as shown in Figure 1 (Maciaszek and Liong, [8], p. 562).

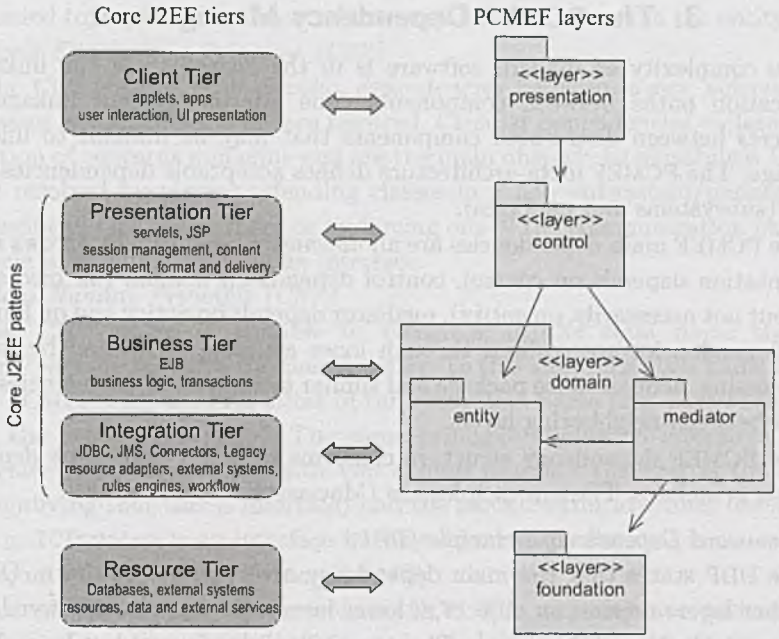


Figure 1. J2EE tiers and PCMEF layers (Source: Maciaszek and Liong, [8], p. 562)

The PCMEF *presentation* layer contains classes that define GUI objects. The user communicates with the system via presentation classes. Accordingly, the class containing the program's `main` function is typically housed in the `presentation` subsystem (alternatively it can reside in the control subsystem).

The *control* layer handles the presentation layer's requests. It consists of classes responsible for processing user's interactions (passed to control from presentation objects). As a result, control is responsible for bulk of the program's logic, algorithmic solutions, main computations, and maintaining session state for each user.

The *entity* subsystem of the domain layer handles the control layer's requests. It contains classes representing "business objects". They store (in the program's memory) objects retrieved from the database or created in order to be stored in the database. Many entity classes are container classes.

The *mediator* package of the domain layer establishes a channel of communication that mediates between entity and foundation classes. Mediation serves two main purposes. Firstly, to isolate the two subsystems so that changes in any one of them can be introduced independently. Secondly, to eliminate a need for the control classes to directly communicate with the foundation classes whenever new entity objects need to

be retrieved from the database (such requests from control classes are then channeled via mediator classes).

The *foundation* layer is responsible for all communications with the database and web services, which manage persistent data required by the application program. This is where the connections to the database and web servers are established, queries to persistent data are constructed, and the database transactions are instigated.

3. The PCMEF Dependency Management

The complexity of modern software is in the "wires" – in the linkages and communication paths between components. The inter-component linkages create *dependencies* between distributed components that may be difficult to understand and manage. The PCMEF meta-architecture defines acceptable dependencies between its layers (subsystems and packages).

The PCMEF main dependencies are all *downward*, as shown by arrows in Figure 1 – presentation depends on control, control depends on domain (on mediator and, possibly but not necessarily, on entity), mediator depends on entity and on foundation. Upward dependencies are realized through loose coupling facilitated by interfaces, event processing, acquaintance package and similar techniques. Dependencies are only permitted between neighboring layers.

The PCMEF dependency structure conforms to seven important *dependency-minimizing principles*. These principles are (Maciaszek and Liong, [8]):

1. *Downward Dependency Principle (DDP)*
   The DDP states that the main dependency structure is top-down. Objects in higher layers depend on objects in lower layers. Consequently, lower layers are more *stable* than higher layers. They are difficult to change, but (paradoxically) not necessarily difficult to extend (Martin, [1]). Interfaces, abstract classes, dominant classes and similar devices should encapsulate stable packages so that they can be extended when needed.
2. *Upward Notification Principle (UNP)*
   The UNP promotes low coupling in a bottom-up communication between layers. This can be achieved by using *asynchronous communication* based on *event processing*. Objects in higher layers act as subscribers (*observers*) to state changes in lower layers. When an object (*publisher*) in a lower layer changes its state, it sends notifications to its subscribers. In response, subscribers can communicate with the publisher (now in the downward direction) so that their states are synchronized with the state of the publisher.
3. *Neighbor Communication Principle (NCP)*
   The NCP demands that a package can only communicate directly with its neighbor package. This principle ensures that the system does not disintegrate to an incompressible network of intercommunicating objects. To enforce this principle, message passing between non-neighboring objects uses *delegation*. In more complex scenarios, a special *acquaintance subsystem* can be used to group interfaces to assist in collaboration that engages distant subsystems.

4. *Explicit Association Principle (EAP)*
   The EAP visibly documents permitted message passing between classes. The principle demands that *associations* are established on all directly collaborating classes. Provided the design conforms to PCMEF, the downward dependencies between classes (as per DDP) command the associations.. Associations resulting from DDP are unidirectional (otherwise they would create circular dependencies). It must be remembered, however, that not all associations between classes are due to message passing. For example, both-directional associations may be needed to implement referential integrity between classes in the entity package.
5. *Cycle Elimination Principle (CEP)*
   The CEP ensures that *circular dependencies* between layers, subsystems and classes within subsystems are resolved. Circular dependencies violate the separation of concerns guideline and are the main obstacle to reusability. Cycles can be resolved by placing offending classes in a new subsystem/package created specifically for the purpose or by forcing one of the communication paths in the cycle to communicate via an interface.
6. *Class Naming Principle (CNP)*
   The CNP makes it possible to recognize in the *class name* the subsystem/package to which the class belongs. To this aim, each class name is prefixed in PCMEF with the first letter of the subsystem name (e.g. `EInvoice` is a class in the entity subsystem). The same principle applies to interfaces. Each interface name is prefixed with two capital letters – the first is the letter "I" (signifying that this is interface) and the second letter identifies the subsystem (e.g. `ICInvoice` is an interface in the subsystem package).
7. *Acquaintance Package Principle (APP)*
   The APP is the consequence of the NCP. The acquaintance package consists of *interfaces* that an object passes, instead of concrete objects, in arguments to method calls. The interfaces can be implemented in any PCMEF subsystem. This effectively allows communication between non-neighboring packages while centralizing dependency management to a single acquaintance package.

4. The PCMEF Architectural Patterns

An *architectural pattern* names and explains the best and widely acknowledged practice to solve a design problem such that the solution ensures a supportable system that conforms to a pre-defined architecture. The supportability aspect of a pattern makes it possible to be applied during the system *design* (and prior to the implementation) or during *refactoring* (Fowler *et al.*, [3]) of the code so that its internal structure can be further improved.

Maciaszek and Liong [8] present and exemplify the whole range of patterns in support of the PCMEF architecture. In here, we will concentrate on the patterns applicable to large enterprise information systems, of which Fowler's *PEAA (Patterns of Enterprise Application Architecture)* are most influential (Fowler, [4]). The PEAA patterns are presented in the context of the PCMEF architecture. The PCMEF-motivated extensions of PEAA patterns, and indeed some new patterns, are also explained.

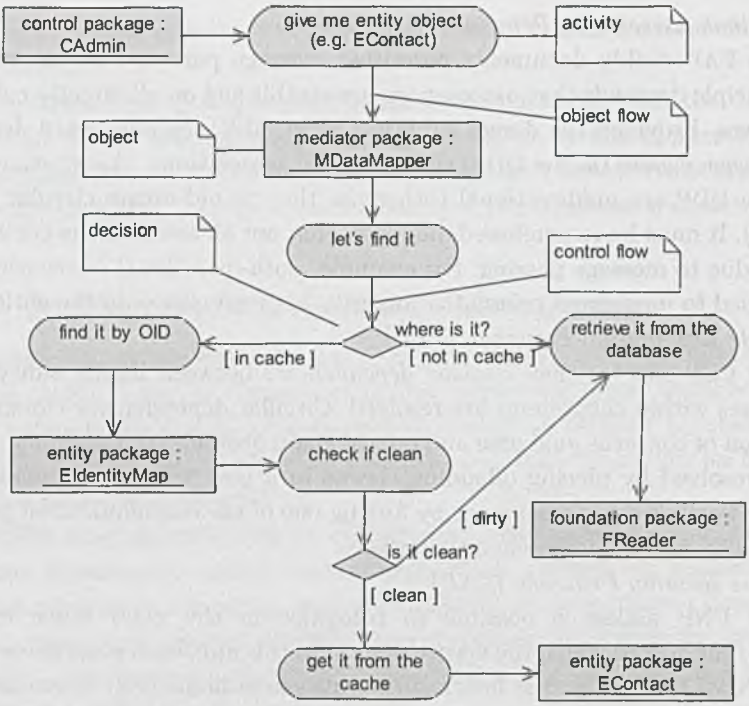


Figure 2. Activity diagram for MDataMapper (Source: Maciaszek and Liong, [8], p. 488)

PCMEF architectural patterns target those aspects of the system that make it an *enterprise application*, as opposed to a small-scale desktop application. They address such issues as communication with the database, in-memory caches of persistent objects, transaction and concurrency management, web presentation, working with distributed objects, etc. Accordingly, they mostly apply to the last two PCMEF layers – mediator and foundation subsystems. Because of the space restriction, only two patterns are explained – Data Mapper and Lazy Load (OID Proxy). Many other patterns are addressed in Maciaszek and Liong [8].

4.1. Data Mapper

The *Data Mapper* pattern depends on another pattern known as the *Identity Map*. An *Identity Map* object maintains in the program's memory one or more maps (e.g. hash maps) that map an object identifier to an object. This pattern ensures that "each object gets loaded only once by keeping every loaded object in a map and looks up objects using the map when referring to them." (Fowler, [4], p. 195). *Identity Map* has two principal objectives:

- to ensure that the same database record is not loaded many times into different application objects (this would create havoc if the application modified one of these objects, but not the others),
- to avoid unnecessary loading of the same data many times (this would impact on the application's performance).

The *Data Mapper* pattern is defined as "a layer of *Mappers* that moves data between objects and a database while keeping them independent of each other and

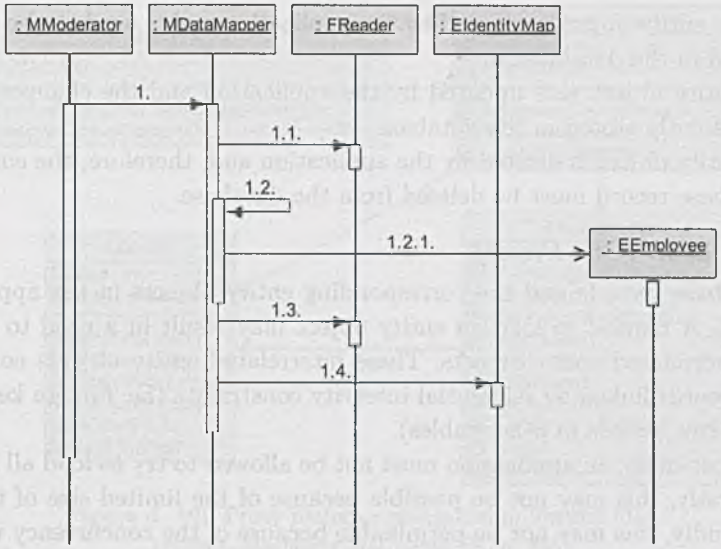


Figure 3. Sequence diagram for object loading.

1. `retrieveEmployeeByName(username: String)`
1.1. `query(sql: String)`
1.2. `createEmployee( )`
1.2.1. `EEmployee(ref: IEDataSupplier)`
1.3. `closeResult(rs: java.sql.ResultSet)`
1.4. `registerEmployee(oidObject: domain.entity.IEObjectID)`

(Source: Maciaszek and Liong, [8], p. 489)

the mapper itself." (Fowler, [4], p. 165). The generic *Mapper* pattern, referred to in the definition, is "an object that sets up a communication between two independent objects." (Fowler, [4], p. 473).

In the PCMEF framework, *Data Mapper* belongs to the mediator subsystem. Figure 2 is an *activity diagram* that shows how the `MDataMapper` class decouples the entity subsystem, (responsible for the cache) from the foundation subsystem (responsible for accessing the database).

Once data records are retrieved from the database, `MDataMapper` makes them into objects (i.e. `MDataMapper` creates new entity objects). The new objects are then registered in `EIdentityMap`. This is a *load* operation – database records are retrieved (loaded) and transformed to memory objects. From the database perspective, the load is a *check-out* operation (records are checked-out to the program).

A fragment of the sequence diagram in Figure 3 illustrates a possible scenario for object loading. The diagram shows how `MDataMapper` retrieves and creates a new `EEmployee` object when asked to get it by employee name. This operation includes the generation of an OID for `EEmployee`, which is then passed to `EIdentityMap` in the parameter of `registerEmployee()`.

Naturally enough, if *Data Mapper* is responsible for loading objects from the database, it should also be responsible for the opposite activity of unloading objects to the database. From the database perspective, this is a *check-in* operation. The unload operation is required when:

- a new entity object is created by the application and it needs to be persistently stored in the database, or
- an entity object gets updated by the application and the changes need to be persistently stored in the database, or
- an entity object is deleted by the application and, therefore, the corresponding database record must be deleted from the database.

4.2. Lazy Load (OID Proxy)

Database records and the corresponding entity objects in the application are interrelated. A request to load an entity object may result in a need to load a long series of interrelated entity objects. These interrelated entity objects correspond to database records linked by referential integrity constraints (i.e. foreign key mappings to primary key records in other tables).

In most cases, an application must not be allowed to try to load all interrelated objects. Firstly, this may not be possible because of the limited size of the memory cache. Secondly, this may not be permissible because of the concurrency requirement of the database, which must allow many applications to use the same records in a conflict-free way. Thirdly, loading objects that may not be needed by the application has a negative impact on performance.

The *Lazy Load* pattern is defined as "an object that doesn't contain all of the data you need but knows how to get it" (Fowler, [4], p. 200). There are a few published approaches to implement *Lazy Load*. The pattern presented here, called *OID Proxy* has been developed specifically for the PCMEF framework (Maciaszek and Liong, [8]).

Object programming environments have internal mechanisms of ensuring the *identity of objects* loaded to memory and use these mechanisms to link objects. The mechanisms vary between languages. None of these mechanisms is a replacement for *OIDs (object identifiers)* assigned explicitly by the program to entity objects when they are instantiated in memory. In all but trivial situations, complex transfers of entity objects between the application and the database call for programs to explicitly assign OIDs to in-memory objects. This introduces duplication between the internal mechanisms of programming languages to ensure identity of objects and the explicit allocation of OIDs by programmers.

Duplication is not always bad. The fact that both techniques link in-memory objects can be taken advantage in *Lazy Load*. The approach can be documented in a pattern – the *OID Proxy* pattern. The *Identity Map* knows if an entity object is loaded. Upon loading, the object is provided with an OID and all its data members are initialized. Initialization includes foreign key (FK) values obtained from the database. OID-based association links corresponding to foreign keys are initialized to null if the linked object has not been loaded yet.

An advantage of *OID Proxy* is the ease with which the *dirty/clean status* of an entity object can be determined. When first loaded, the entity object is flagged as clean. If the data content of the object gets out of sync with its corresponding content in the database, the flag is set to dirty. At all times, the entity object knows its status and can trigger re-loading to refresh its content from the database.

There are two variations of *OID Proxy* that differ in how the program navigates between associated entity objects:

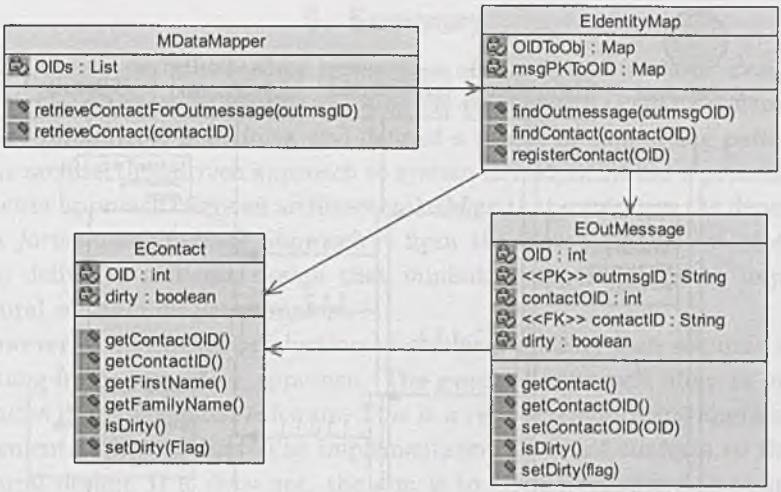


Figure 4. OID Proxy pattern – navigation in Identity Map
(Source: Maciaszek and Liong, [8], p. 498)

- Navigation in Identity Map
- Navigation in Entity Classes

Because of the space restriction, this paper explains only the *navigation in identity map* approach. An *Identity Map* keeps every loaded entity object in a map. The mapping maintains relationships between an OID and its object. Other maps that relate primary key values to OIDs can also be maintained in order to support requests to search for objects based on their field values. These maps are in general sufficient to navigate between objects that are linked in the database by referential integrity constraints.

The navigation requires that an *Identity Map* interrogate entity objects to establish *association links*. An association link in an entity object is represented in two ways:

- by a foreign key (FK) value obtained from the database, and
- by an OID to a related object in memory.

Both FK and OID data members can be `null`. A `null` in the foreign key has the same meaning as in the database – the object is not linked. A `null` in the OID means that the associated object is not loaded.

Figure 4 shows a class model that uses the *OID Proxy*. The class `EOutMessage` contains a data member `contactOID` that serves as a proxy for the `EContact` object it links to. As per the *Lazy Load* pattern, `EOutMessage` does not contain the data requested when `getContact()` is called on it, but it knows how to get it.

`MDataMapper` keeps a list of OIDs of all objects loaded in memory. When passed an OID of any object, it can immediately verify if such object is loaded. When passed a primary key (PK) value of an object (e.g. `outmsgID`), it can ask `EIdentityMap` for corresponding OID.

Figure 5 explains a possible sequence of messages. With *Identity Map* in place, each entity object knows its state (if it is clean or dirty). The state is checked by `isDirty()`. Assuming that `EOutMessage` is clean, `EIdentityMap` asks `EOutMessage`

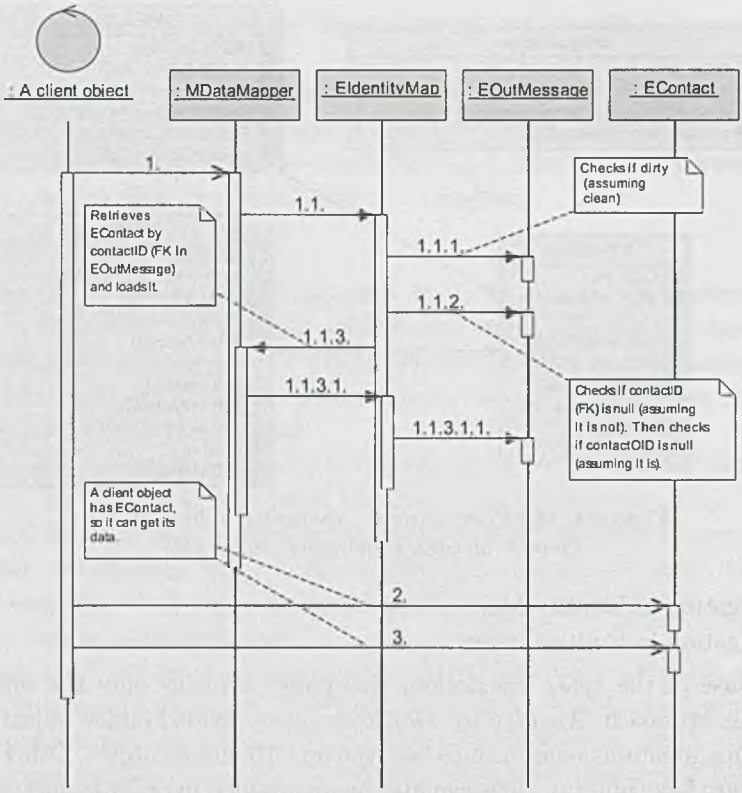


Figure 5. OID Proxy pattern – navigation in Identity Map (sequence diagram).

1. `retrieveContactForOutmessage(outmsgID)`
1.1. `findOutmessage(outmsgOID)`
1.1.1. `isDirty( )`
1.1.2. `getContact( )`
1.1.3. `retrieveContact(contactID)`
1.1.3.1. `registerContact(OID)`
1.1.3.1.1. `setContactOID(OID)`
2. `getFirstName( )`
3. `getFamilyName( )`

(Source: Maciaszek and Liong, [8], p. 499)

to `getContact()`. `EOutMessage` checks if the foreign key value to `EContact` is not `null`. It then checks if `contactOID` is not `null`.

`EContact` may or may not have been loaded in memory. `EOutMessage` has this knowledge in `contactOID`. If `contactOID` is null, then `EContact` has not been loaded. The sequence diagram in Figure 5 assumes that this is the case. `EIdentityMap` asks `MDataMapper` to retrieve the `EContact` based on the foreign key value.

The outcome will be that `EContact` is loaded (lazily). `MDataMapper` will update its list of OIDs, `EIdentityMap` will update its maps, and `contactOID` in `EOutMessage` gets set. `EContact` is now loaded and returned to the client object, which can now ask it directly for data (`getFirstName()`, `getFamilyName()`, etc.).

`EIdentityMap` is in charge of all navigations between entity objects. In this approach, entity objects do not have to contain associated objects. Entity objects are contained in maps of `EIdentityMap`. Entity objects themselves have only OIDs to other objects.

5. Summary

This paper described some principles of good *architectural design* aimed at *supportable systems*. It introduced a *meta-architecture* (PCMEF), explained its *dependency-minimizing principles*, and defined a couple of supporting *patterns*.

The architecture-driven approach to system development has a proactive flavor. The *proactive* approach offers an architectural design that minimizes the dependencies. This is a *forward-engineering* approach – from the design to implementation. The aim is to deliver a software design that minimizes dependencies by imposing an architectural solution on programmers.

However, a successful production of a supportable system requires a parallel contribution from a reactive approach. The *reactive* approach aims at measuring dependencies in implemented software. This is a *reverse-engineering* approach – from the implementation to design. The implementation may not conform to the desired architectural design. If it does not, the aim is to compare the metric values in the software with the values that the desired architecture would have delivered. The troublesome dependencies can then be pinpointed.

Measuring supportability of designs and programs cannot be done manually. Maciaszek and Liong [7] describes a tool, called *DQ* (Design Quantifier), which is able to analyze any Java program, establish its conformance with a chosen supportable meta-architecture, compute complete set of dependency metrics, and visualize the computed values in UML class diagrams.

References

[1] D. Alur, J. Crupi, and D. Malks: *Core J2EE Patterns: Best Practices and Design Strategies*, 2 ed., Prentice Hall, 2003, 528 p..

[2] F.P. Brooks: *No Silver Bullet: Essence and Accidents of Software Engineering*, IEEE Software, 4, 1987, pp. 10–19; reprinted in: *Software Project Management. Readings and Cases*, ed. C.F. Kemerer, Irwin, 1997, pp. 2–14.

[3] M. Fowler, with K. Beck, J. Brant, W. Opdyke, and D. Roberts: *Refactoring. Improving the Design of Existing Code*, Addison-Wesley, 1999, 431 p.

[4] M. Fowler: *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2003, 531 p.

[5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides: *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995, 395 p.

[6] L.A. Maciaszek: *Managing Complexity of Enterprise Information Systems*, ICEIS 2004 Sixth Int. Conf. on Enterprise Information Systems, Volume I, INSTICC, April 2004, Porto, Portugal, pp. IS-9–IS-13 (keynote paper; extended version to appear in a book by Kluwer Academic).

[7] L.A. Maciaszek and B.L. Liong: *Designing Measurably-Supportable Systems*, Advanced Information Technologies for Management, Research Papers No 986, ed. by E. Niedzielska, H. Dudycz, M. Dyczkowski, Wroclaw University of Economics, 2003, pp. 120–149

[8] L.A. Maciaszek and B.L. Liong: *Practical Software Engineering. A Case-Study Approach*, Addison-Wesley, 2005, 864 p.

[9] R.C. Martin: *Agile Software Development, Principles, Patterns, and Practices*, Prentice-Hall, 2003, 529 p.

STRATEGIA BEZPIECZEŃSTWA, ZARZĄDZANIE RYZYKIEM: TRENDY I PRZYKŁADY NAJLEPSZYCH PRAKTYK

(streszczenie wystąpienia)

DAREK MILEWSKI

Oracle Corporation

Katalizator

Mimo zaawansowania technologii, w tym technologii bezpieczeństwa, świat dziś jawi się jako bardziej niebezpieczny niż pięć lat temu. Tragedia 11 września 2001 pokazała, jak Stany Zjednoczone patrzą na świat i traktują bezpieczeństwo wewnętrzne. Gwałtowny wzrost wydatków na nowe środki bezpieczeństwa w USA (z 5 miliardów dolarów w 2000 roku do 85 miliardów dolarów w 2004 roku) spowodował, że technologie i strategie bezpieczeństwa, dotąd ezoteryczne, stały się faktem. Czy tak się jednak stało naprawdę?

Niewiele się zmieniło jeśli chodzi o fundamentalne elementy bezpieczeństwa informacji. Szeroki dostęp do Internetu, technologie bezprzewodowe (802.11x), nieustająca powódź patchy, a przede wszystkim brak powszechnej świadomości nadal są bolączką. Nie wystarczy mieć PKI z czytnikiem biometrycznym połączonym z alarmami i strażnikami pod bronią, jeśli aplikacja czy system operacyjny pozwolą na zainstalowanie keylogera przez nieświadomego (lub nie) sprzedawcę czy asystenta. Nawet szyfrowanie AES nam wtedy nie pomoże. Sęk tkwi w edukacji i strategii bezpieczeństwa, które wychodzą poza paradygmat technologii.

Kto jest dziś klientem, kto dostawcą, kto pracownikiem? Granice w tradycyjnym łańcuchu dostaw się rozmywają. Dziś tradycyjny firewall do naszego ekstranetu wygląda jak sito. Nie sposób tym zarządzać. Czy czas poważnie pomyśleć o Jericho?

Prawo i interwencja państwa w sprawy bezpieczeństwa informacji powodują często dość kosztowne skutki dla wielu instytucji. Być może już wkrótce w USA raportowanie stanu bezpieczeństwa stanie się obowiązkiem firm w obrocie publicznym (The Putnam Act). Już dziś nasi klienci domagają się atestów, ISO17799, itp.

Mimo wszystko, rośnie dysonans pomiędzy bezpieczeństwem „technicznym" (rozwiązanie problemu przez wykorzystanie nowej technologii), a tradycyjnym biznesem, który domaga się rozsądku w wyważeniu strat i korzyści. I jak udowodnić ROI na bezpieczeństwo?

Outsourcing i offshoring to gorące tematy. Już upublicznili się pierwsi poparzeni (brak praw własności intelektualnej, niemożliwość ścigania, brak wiarygodności, itp.). Co zrobić, by nie dołączyć do grona poszkodowanych?

Trendy i (nie)konwencjonalne strategie

Informatyka a zarządzanie ryzykiem – do tej pory całkowicie rozdzielano lub, przeciwnie, utożsamiano te dwa obszary. Ani jedno ani drugie podejście dziś się nie sprawdza. Informatycy muszą wziąć kilka kursów z MBA, a zarządzający ryzykiem księgowi zrozumieć, jak technologia może ułatwić i odmienić pracę. Wymagania Basel II w zakresie ryzyka operacyjnego w bankach są pomocne. Ale trzeba wyjść poza mierzenie ryzyka.

Nowy paradygmat „od bezpieczeństwa informacji do zabezpieczenia biznesu" (jak zostało to ujęte przez Working Council for Chief Information Officers) zakłada, że szerszy zakres zagrożeń, skomplikowana struktura organizacji, zasady dobrego nadzoru i zarządzania spółkami (corporate governance), standardy, płynność granic pomiędzy organizacjami – spinają bezpieczeństwo informacji z innymi działaniami zabezpieczającymi funkcjonowanie, przetrwanie i rozwój firm.

Zarządzanie ryzykiem staje się istotnym czynnikiem ustalania priorytetów. Nie można zapewnić bezpieczeństwa wszystkim w takim samym stopniu. Musimy wiedzieć, co mamy najcenniejszego, jak długo możemy funkcjonować bez tego i co się stanie jak, nie będziemy tego mieli. Rolą nowej funkcji zarządzania bezpieczeństwem jest dostarczenie biznesowi narzędzi i metod do dokonania świadomego i nieskomplikowanego wyboru i podjęcia decyzji.

Kluczowa rola w podejmowaniu strategicznych decyzji niesie ze sobą potrzebę umieszczenia funkcji zarządzania bezpieczeństwem/ryzykiem bliżej zarządu. Dziś obszar bezpieczeństwa informacji podlega zwykle (w 40%) CIO – szefowi ds. informatyki. Równie często leży w gestii CFO – dyrektora finansowego. Nowym trendem jest podporządkowanie stworzonej funkcji Chief Information Security Officer samemu CEO – dyrektorowi zarządzającemu. To pozwala na większą wiarygodność (i efektywność) w dostarczaniu i egzekwowaniu polityk, standardów, architektury, wyborze partnerów, przetwarzaniu danych.

Choć rola zarządzania bezpieczeństwem informacji jest centralna na poziomie strategii, polityki i governance, to większość funkcji operacyjnych ciągle znajduje się w rękach innych jednostek organizacyjnych. Funkcje wykrywania ataków (intrusion detection), zarządzania incydentami, zapewnienia ciągłości operacyjnej wymagają szybkości i dostosowania do lokalnego otoczenia. Gdy funkcje te realizowane są centralnie, nie są w stanie spełnić takich wymagań.

Zapewnienie ciągłości operacji (BCP) jest dziś kluczową strategią, zwiększającą szanse przetrwania organizacji w sytuacji kryzysowej, a jednocześnie umożliwiającą zwiększenie przewagi rynkowej. Integrowanie systemów informatycznych z zastosowaniem wszelkich technik zabezpieczeń jest oczywiście istotne. Jednak równie istotne jest organizacyjne przygotowanie do zarządzania kryzysem. Doświadczenie pokazuje, że metodologie oparte na współwłasności BCP pomiędzy jednostkami organizacyjnymi i lokalnym dostosowaniu planu są dziś najefektywniejsze. Podstawą są systemy komunikacyjne i ich dostępność.

Business Protection, następny szczebel w ewolucji zarządzania bezpieczeństwem informacji staje się nowym paradygmatem.

Tradycyjna architektura opierająca się na ochronie perymetru nie spełnia wymagań nowej rzeczywistości i nowych technologii. Choć wiele firm ucieka przed technologią bezprzewodową (802.11x), ona jest i zostanie. Plaska topologia sieci, brak rozsądnej segmentacji, nie pozwoli nam na wykorzystanie najnowszych technologii dostępu.

Najlepsze praktyki w nowatorskim zarządzaniu bezpieczeństwem informacji

Na koniec garść przykładów praktyk godnych bliższego przyjrzenia się:

- portfelowe podejście do zarządzania bezpieczeństwem i przełożenie na strategię – firma petrochemiczna i PeopleSoft;
- kampania edukacyjna – PeopleSoft;
- rozsądna segmentacja sieci – HP;
- dwie metody zarządzania tożsamością – Microsoft i Oracle;
- przykładowe użycie SAML w bezpiecznej współpracy – Boeing i Southwest Airlines;
- narzędzie do zarządzania ryzykiem – Oracle Threat Manager;
- spoiwo i narzędzie compliance – Oracle Internal Control Manager.

SYMULACJA KOMPUTEROWA W PROJEKTOWANIU SYSTEMÓW CZASU RZECZYWISTEGO

TADEUSZ NOWICKI

Wojskowa Akademia Techniczna
Instytut Systemów Informatycznych, Wydział Cybernetyki
nowicki@isi.wat.waw.pl

Streszczenie: W pracy przedstawiono stan, perspektywy i problemy modelowania symulacyjnego. Zarysowano chronologicznie rozwój metod, technik i narzędzi oraz aktualny stan prac w dziedzinie symulacji komputerowej systemów czasu rzeczywistego, w szczególności Interaktywnej Symulacji Rozproszonej. Przedstawiono również podstawowe języki i środowiska wytwórcze oprogramowania symulacyjnego stosowane na świecie w heterogenicznym środowisku sprzętowo-programowym. Zaprezentowano przykład zastosowania technik symulacji do modelowania wybranego systemu rzeczywistego. Praca zawiera również uzasadnienie tezy, dlaczego symulację komputerową można nazwać technologią przyszłości.

1. Wprowadzenie

Symulacja jest procesem projektowania modelu rzeczywistego lub wyimaginowanego systemu i przeprowadzenia eksperymentów z tym modelem (Encyclopedia of Computer Science). Przyczyną stosowania eksperymentów symulacyjnych jest chęć poznania (często jedynie zrozumienia) zachowania się systemu lub konstrukcji algorytmów sterowania tym systemem. Problemy występujące w rzeczywistym świecie są na tyle złożone, że stosowane modele matematyczne (lub inne formalne modele) zjawisk związanych z tymi problemami nie wystarczą do ich odwzorowania. W tym przypadku zachowanie się systemu musi być oddane dzięki zastosowaniu symulacji. Ze względu na popularne obecnie środowisko realizacji eksperymentów symulacyjnych przyjęto nazwę symulacji komputerowej lub symulacji cyfrowej.

Modelowanie symulacyjne uwzględnia: szeroki zakres i wielką skalę problemów (w tym problem życia ludzkiego), dużą różnorodność elementów, powikłane zależności, złożone charakterystyki elementów, skąpe źródła miarodajnych i aktualnych informacji, losowy charakter zjawisk, błędy i zniekształcenia celowe informacji. Modelowanie symulacyjne polega na: zdefiniowaniu problemu, zdefiniowaniu modelu konceptualnego, zebraniu danych wejściowych, konstrukcji oprogramowania symulacyjnego. Z kolei realizacja symulacji komputerowej obejmuje: opracowanie planu eksperymentów, wykonanie eksperymentów symulacyjnych, zebranie danych wynikowych eksperymentów, analizę uzyskanych danych, dokumentowanie rezultatów badań, prace nad rozszerzeniem własności modelu.

Projektowanie procesów symulacji związane jest ściśle z następującymi problemami: zarządzanie zdarzeniami, kolejkowanie zdarzeń, generowanie zdarzeń losowych, zarządzanie czasem systemowym, sieci komputerowe i obliczenia równoległe, konserwatywne i optymistyczne algorytmy synchronizacji, modelowanie behawioralne, zarządzanie modelem.

Do technologii komputerowych związanych silnie z rozwojem symulacji komputerowej należą:

- sieci komputerowe – naturalne środowisko sprzętowo-programowe, w którym realizowane są złożone eksperymenty symulacyjne;
- obliczenia równoległe – korzystanie z własności sieci komputerowych prowadzi w sposób naturalny do wykorzystania technik zrównoleglenia obliczeń symulacyjnych, umożliwiających przyśpieszenie obliczeń i rozproszenie aplikacji, co wspomaga proces symulacji;
- sztuczna inteligencja – do reprezentacji w symulacji komputerowej zachowań ludzi wykorzystuje się modele i techniki sztucznej inteligencji: systemy ekspertowe, sieci neuronowe, algorytmy genetyczne, itd.;
- grafika komputerowa – symulacja komputerowa pozwala wykorzystać techniki zobrazowania graficznego wyników symulacji, co jest niejednokrotnie wymogiem stawianym przez użytkownika symulatorów;
- bazy danych – eksperymenty symulacyjne generują zwykle duże ilości informacji, które są później analizowane, co powoduje konieczność budowy stowarzyszonych z symulatorami baz danych – relacyjnych lub obiektowo zorientowanych – lecz również wymagana jest łączność z istniejącymi już bazami danych stanowiącymi dane źródłowe procesów symulacji;
- WWW – ekspansja Internetu i technologii WEB-owej spowodowała konieczność uwzględnienia tej technologii dla interakcyjnej realizacji eksperymentów symulacyjnych przez użytkowników „zdalnych".

2. Podstawy symulacji komputerowej systemów czasu rzeczywistego

W literaturze przyjmuje się następujące etapy procesu symulacji: projektowanie modelu, realizowanie eksperymentu, analizowanie wyników. Zauważmy, że tylko w fazie realizowania eksperymentu występuje możliwość wyboru konkretnej metody. Może być to uzależnione od wielu czynników, m.in. natury zjawisk w modelowanym systemie, preferowanych technik i posiadanych narzędzi aplikacyjnych do realizacji eksperymentu. Wydaje się, że największą popularność zyskały metody uwzględniające dyskretny upływ czasu, zorientowane na zdarzenia. Do znanych technik wytwarzania oprogramowania symulacyjnego systemów czasu rzeczywistego należą: techniki sekwencyjne, techniki równoległe PADS, techniki rozproszone DIS oraz najnowszy standard HLA.

Techniki sekwencyjnie (*Serial algorithms*) – technika stosowana w przypadku nieskomplikowanych symulacji realizowanych w postaci programu na jednym komputerze. Z jej pomocą można wykonać prawie każdy eksperyment, ale nie zawsze uzyskuje się akceptowalną jakość.

Techniki równoległe (*Parallel Discrete Event Simulation PDES*) – wykorzystywana do realizacji w ograniczonym czasie złożonych obliczeń, wymagających jednostek o dużych mocach obliczeniowych. Jej podstawą jest podział zadań pomiędzy wiele komputerów użytych do synchronicznego wykonania przydzielonych fragmentów zadań. Podkreślić należy, że podział zadań wynika tylko z zastosowanych algorytmów przydziału a nie z właściwości modelowanego systemu.

Eksperymenty rozproszone (*Parallel And Distributed Simulation-PADS*) – w symulacji rozproszonej dokonuje się odwzorowania rzeczywistego systemu w posiadane zasoby sprzętowo-programowe tak, by zachować podstawowe cechy elementów systemu i zachodzących w nim zjawisk. Asynchroniczne rozproszenie zadań na wiele komputerów wynika przede wszystkim z konieczności zachowania wzajemnych powiązań elementów.

Mianem DIS („Distributed and Interactive Simulation") określa się symulację prowadzoną na różnych platformach sprzętowo-programowych, w niezależnych i odległych miejscach, z wykorzystaniem sieci komputerowych, symulatorów oraz włączającą interaktywne oddziaływanie człowieka na systemy lub jednostki w celu dokonania zmian w ich zachowaniu.

Z kolei HLA, którego korzeniem jest DIS, jest zbiorem zaleceń, unormowanych definicji oraz wymagań jakie powinny być spełnione na wszystkich etapach procesu wytwarzania systemu symulacyjnego. Jest więc technologią a nawet metodologią realizacji systemów realistycznego, kompleksowego, rozległego, interaktywnego „świata" symulacji. Jednoczy w sobie różne cele, bazując na przeróżnych technikach symulacji, niejednorodnych środowiskach sprzętowych i programowych. W rezultacie wymiany informacji pomiędzy aplikacjami, sprzętem i ludźmi poprzez wydzielone interfejsy i łącza uzyskuje się syntetyczne środowisko symulacji.

W wystąpieniu przedstawione zostaną podstawowe cechy wszystkich technik stosowanych w modelowaniu symulacyjnym oraz dokonane zostanie porównanie właściwości technologii PADS, DIS oraz HLA.

3. Języki i środowiska programowe symulacji

Zakres stosowanych na świecie języków symulacji i środowisk wytwórczych oprogramowania symulacyjnego, wbrew pozorom, nie jest szeroki. Historycznie rzecz biorąc do najważniejszych języków symulacyjnych należą: Simula, GPSS/H, SIMSCRIPT II.5, SIMAN/Cinema, SLAM II, MODSIM. Z kolei do popularnych na świecie pakietów symulacyjnych należą: Worbench, TAYLOR II, COMNET III, BONeS Designer, SimPack, oraz MultiGenII. Wszystkie one zostaną krótko omówione w prezentowanym referacie.

4. Perspektywy symulacji komputerowej

W pracy Strategic Directions in Simulation Research (Fujimoto, Fishwick i inni) autorzy przedstawili podstawowe strategiczne problemy rozwoju w badaniach symulacyjnych. Niektóre z nich, to:

- Problemy skali – pomimo zwiększających się możliwości sprzętu i metod wytwarzania oprogramowania budowane modele symulacyjne odpowiadają coraz

to zwiększającym się systemom w sensie rozmiaru i złożoności. Problemy symulacji dużej skali są nadal problemem niezwykle ważnym.

- Weryfikacja, ocena i adekwatność modeli symulacyjnych – aplikacje symulacyjne z punktu widzenia praktyków, badaczy i osób zarządzających muszą być weryfikowalne, co nadal stwarza olbrzymie problemy projektantom.
- Problemy równoległej i rozproszonej symulacji – w minionych latach badacze lamentowali nad zbyt małą pojemnością pamięci, zbyt małą szybkością procesorów, słabymi procedurami graficznymi itd., ta sytuacja zmieniła się diametralnie w ostatnich latach. Jednak istnieją jeszcze kierunki, które koniecznie należy rozwijać: HLA jako standard dla DIS, przezroczystość oprogramowania symulacyjnego, metody synchronizacji procesów symulacyjnych, komunikacja z bazami danych, komunikacja z zewnętrznymi aplikacjami wykorzystywanymi w symulacji oraz użycie heterogenicznych środowisk obliczeniowych dla aplikacji symulacyjnych.
- Problem zdalnego dostępu – pomimo stosowanych powszechnie technik zdalnego łączenia aplikacji z użytkownikiem pozostają jeszcze liczne nierozwiązane w tym zakresie zagadnienia.
- Od modelowania do programowania – koniecznym jest rozwój metod i technik pozwalających na systematyczne przejście od procesu modelowania rzeczywistości do uzyskania ostatecznego kodu programowego symulatorów. Stosowanie współczesnych technik, takich jak UML czy RUP staje się wyzwaniem stojącym przez zespołami projektantów oprogramowania symulacyjnego.
- Efektywność aplikacji symulacyjnych – zakres i złożoność modeli symulacyjnych powoduje konieczność uwzględnienia problemów efektywności aplikacji powstałych w wyniku projektowania oprogramowania symulacyjnego.
- Inne strategiczne kierunki w symulacji rozproszonej – systemy operacyjne, podejmowanie decyzji w czasie rzeczywistym, świat wirtualny, modelowanie zachowań ludzkich i praktyczne metody zarządzania zdarzeniami.

5. Podsumowanie

Rozwój sprzętu komputerowego i technik wytwarzania oprogramowania implikuje zarówno postęp w tworzeniu różnego rodzaju aplikacji komputerowych, jak i rozwój dziedziny modelowania i symulacji. Symulacja komputerowa napotykała stosunkowo niedawno na ograniczenia w jej stosowaniu, wynikające głównie z możliwości środowisk wytwórczych i realizacyjnych oprogramowania symulacyjnego. Obecnie modele symulacyjne są uruchamiane „całościowo' przez specjalnie przygotowany personel na dedykowanych systemach komputerowych. Osiągnięty został stan, gdy złożone procesy symulacji mogą być realizowane na komputerach osobistych przez ekspertów różnych dziedzin bez pomocy specjalistów z symulacji komputerowej. Badania w ramach rozwoju metod i technik symulacji pozwalają na stosowanie wszelkich metod formalnych w definiowaniu modeli, interoperacyjność, interaktywną symulację, meta-modelowanie, modelowanie zachowania człowieka oraz równoległą i zdalną symulację.

Należy spodziewać się, że w przyszłości procesy produkcji, różnego typu badań, treningu zespołów decydentów, itd. nie będą mogły obyć się bez wspomagania oferowanego przez użycie technik symulacji, jako szybkich i ekonomicznych. Jako ze

świat dąży do społeczeństwa informacyjnego, zatem wszelkie przejawy aktywności miedzy innymi w biznesie, nauce czy w sferach rządzących będą definiowane w formie cyfrowej informacji, która musi być zorganizowana, analizowana i prognozowana za pomocą symulacji.

Literatura

[1] K.M. Chandy, J. Misra: *A case study in design and verification of distributed programs*, IEEE Transactions on Software Engineering: SE-5(5): 440–452, September 1979.

[2] K.M. Chandy, J. Misra: *Asynchronus Distributed Simulation via a Sequence of Parallel Computations*, Communications of the ACM, 24(11): 198–206, November 1981.

[3] *Encyclopedia of Computer Science*, 4th Edition, Macmillan References Press, New York, November 1999.

[4] A. Ferscha: *Parallel and Distributed Simulation of Discrete Event Systems*, McGraw-Hill, 1995.

[5] P.A. Fishwick i in.: *Strategic directions in simulation research*, Proceedings of the Winter Simulation Conference, 1999.

[6] P.A. Fishwick: *Simulation Model Design And Execution*, Prentice Hall, New Jersey, 1995.

[7] US DoD: *High-Level Architecture Object Model Template Specification*, IEEE, 1998.

JAKIE UNIWERSYTETY?*

LESZEK PACHOLSKI

http://www.ii.uni.wroc.pl/~pacholsk

W styczniu „The Economist" opublikował artykuł o szkolnictwie wyższym. Zawiera on między innymi informacje o opublikowanym przez Komisję Europejską rankingu najlepszych uniwersytetów na świecie i o wydatkach na szkolnictwo wyższe.

Oczywiście mowa jest o pieniądzach, czesnym i państwowych dotacjach, ale główna teza brzmi: „unikajmy systemu państwowego, jednorodnego, popierajmy różnorodność i niezależność" Artykuł ostrzega przed systemem, który „bardziej pasuje do sowieckiego systemu centralnego planowania niż do współczesnej demokracji". Czytając to, trudno było nie pomyśleć o złożonych w polskim Sejmie projektach nowej ustawy o szkolnictwie wyższym.

Według „The Economist", rząd francuski przeznacza na szkolnictwo wyższe prawie 6% produktu krajowego brutto. Więcej niż USA, Wielka Brytania, Niemcy, Japonia i Szwajcaria. Wprawdzie w USA fundusze na naukę są znacząco uzupełniane ze źródeł prywatnych, więc całkowite wydatki są tam nieco większe, ale i tak analiza rankingu uniwersytetów szokuje.

Najlepsze uniwersytety są w USA – Harvard, Stanford, Caltech, UC Berkeley. Najlepszy uniwersytet francuski jest na 65 miejscu. Wyprzedza go 37 uniwersytetów amerykańskich i 14 europejskich z krajów wydających na szkolnictwo mniej niż Francja. W gronie 50 najlepszych uniwersytetów europejskich są oczywiście Oxford i Cambridge, ale także inne uniwersytety brytyjskie: Imperial College, University College i uniwersytet w Edynburgu, a także uniwersytet i politechnika w Zurychu oraz uniwersytety w Sztokholmie, Utrechcie i Monachium. Inne badania wykazują, że Stany Zjednoczone przodują nie tylko w rankingach uniwersytetów, ale także w liczbie oraz finansowej wartości patentów, w eksporcie nowoczesnych technologii – właściwie we wszystkim, z wyjątkiem atrakcyjności zabytków i jakości kuchni.

W kwietniu we Francji wybuchły strajki uczonych. Głównym ich powodem było to, że rząd przekształcił 540 stałych etatów w etaty czasowe. W USA strajki tego typu byłyby niemożliwe, bo tam rząd takich decyzji nie podejmuje. I tu dochodzimy do prawdziwej przyczyny tego, że za niemal takie same pieniądze w USA jest dobrze,

* Skrócona wersja artykułu, zatytułowana *Zacznijmy lepiej uczyć studentów*, ukazała się w „Tygodniku Powszechnym", http://www.tygodnik.onet.pl, 24 października 2004.

a we Francji nie. Źródłem ogromnych różnic pomiędzy efektywnością nauki i szkolnictwa wyższego we Francji i w USA są francuski centralizm i amerykańska autonomia uniwersytetów.

Znaczna część wydatków na szkolnictwo i naukę we Francji to wydatki na utrzymanie niesłychanie rozbudowanej biurokracji. Szkolnictwo w Wielkiej Brytanii, które w porównaniu z francuskim jest znacznie gorzej finansowane, ma lepsze wyniki, bo tamtejsze szkoły są mniej spętane biurokracją.

Unia Europejska jest zaniepokojona amerykańską supremacją. Niestety, niewiele z tego niepokoju wynika. Mamy wprawdzie deklaracje polityków: strategie lizbońskie i procesy bolońskie, apele, międzynarodowe akademie i uroczystości, obietnice utworzenia programów, powołania komitetów, zespołów i biur. Zbudujemy w Europie najlepszą, najsprawniejszą, najnowocześniejszą i najdroższą administrację zajmującą się nadzorowaniem badań naukowych i szkolnictwa. Zapewne pozwoli nam to wyprzedzić USA w objętości formularzy grantowych i długości sprawozdań naukowych, tak jak dawniej w RWPG wyprzedzaliśmy imperialistów w produkcji surówki i lokomotyw.

Wąskie cele

Polska wciąż stoi przed wyborem modelu edukacji wyższej i modelu organizacji badań naukowych. Istniejący system i prawo coraz bardziej hamują rozwój i prowadzą do patologii. Niestety, niezbędnych reform nie przeprowadzono wtedy, gdy reformowano system polityczny i ekonomiczny. Teraz będzie trudniej, nie ma bowiem entuzjazmu do zmian ani w klasie politycznej, ani wśród naukowców. My jesteśmy ze wszystkiego oprócz pensji zadowoleni, a politycy mają na głowie sprawy, które łatwiej przeliczyć na głosy wyborców.

A od tego, jaki model edukacji wyższej zostanie wybrany w Polsce, zależy bardzo dużo. Przykład Irlandii wskazuje, jak istotny ma to wpływ na tempo rozwoju gospodarczego. Musimy podjąć decyzję, czy tak jak Francja będziemy wyrzucać pieniądze w błoto, czy też zainwestujemy je w przyszłość.

W ostatnich dniach czerwca Sejm skierował do komisji dwa konkurencyjne projekty ustawy regulującej sprawy szkolnictwa wyższego. Jeśli ustawa zostanie uchwalona, będzie prawdopodobnie obowiązywać przez kilkanaście lat. Mimo że dotyczy najżywotniejszych spraw szkolnictwa i nauki w Polsce, nie wzbudziła zainteresowania środowiska naukowego. Senaty szkół wyższych, z wyjątkiem jednego, nie dyskutowały nad projektami. Nie było też prawdziwej dyskusji na łamach prasy.

Jeden projekt – prezydencki, przygotował powołany przez Prezydenta RP zespół złożony z obecnych i byłych rektorów, drugi – poselski, jest modyfikacją projektu Krajowej Sekcji Nauki NSZZ „Solidarność". Projekty te, prócz tego, że petryfikują obecny, zdecydowanie zły system, przynoszą dość istotne nowe zagrożenia. Trudno nie zgodzić się z profesorem Łukaszem Turskim, który we „Wprost" z 8 sierpnia napisał: „Oba projekty zostały napisane po to, by pewna część środowiska akademickiego zrealizowała swoje wąskie cele".

Projekt poselski kładzie nacisk na zniesienie habilitacji, projekt prezydencki – na ustawowe umocowanie licznych ciał zarządzających życiem akademickim i centralizację decyzji dotyczących programów nauczania. Oba projekty w zasadzie nie

podnoszą innych spraw i nie rozwiązują istotnych problemów szkolnictwa wyższego i nauki w Polsce. Tam, gdzie nie dotyczy to „wąskich celów", nie proponuje się żadnych zmian w istniejącej ustawie, nawet w tych miejscach, gdzie istniejące zapisy są ewidentnie złe. Na przykład oba projekty proponują pozostawienie dotychczasowego, absurdalnego, dwustopniowego sposobu wyboru członków Rady Głównej Szkolnictwa Wyższego, nie zapewniającego ani reprezentatywności, ani wysokiego poziomu członków Rady. W trakcie dotychczasowych prac nad ustawą nie było żadnej dyskusji, która pozwoliłaby na sporządzenie listy niezbędnych lub pożądanych zmian oraz pomogłaby znaleźć możliwie najlepsze rozwiązania. Przy całym szacunku dla wysokich gremiów, które nad projektami pracowały – mogły one niejedno przeoczyć.

Precz z habilitacją?

W „Polityce" z 24 lipca ukazał się artykuł Cezarego Wójcika i Józefa Wieczorka, postulujący zniesienie habilitacji. Artykuł żarliwy, pełen troski o stan nauki polskiej, ale jednocześnie trochę bałamutny i bardzo naiwny. Zaczyna się od spacyfikowania potencjalnych polemistów. Ewentualni obrońcy habilitacji to zwolennicy tytułomanii niespotykanej w cywilizowanym świecie i sprzymierzeńcy profesorów z nadania Łukaszenki, uczeni na poziomie Kuby, Wietnamu i Korei Północnej. Aż strach się odezwać. Artykuł jest bałamutny, bo już tytuł sugeruje, że dożywotnie zatrudnienie jest czymś wyjątkowym – elementem polskiej patologii. A przecież obaj panowie doskonale wiedzą, że na przykład na uniwersytetach amerykańskich, profesor po sześciu latach pracy albo zostaje zwolniony, albo dostaje „tenure", co oznacza, że ma, podobnie jak prof. dr hab., posadę do emerytury.

Zamierzam bronić habilitacji, ale z wieloma tezami autorów się zgadzam – w szczególności z tym, że nauka w Polsce znajduje się w stanie zapaści. Oczywiście są ośrodki, gdzie prowadzi się badania naukowe na przyzwoitym światowym poziomie, i są uczeni, którzy są światowymi liderami. Są też, niestety, ogromne rzesze naukowców, publikujących w wydawnictwach, których nikt nie czyta, naukowcy, którym nawet do głowy nie przyjdzie, że można napisać coś, co zainteresuje kogokolwiek na świecie.

Niestety, zniesienie habilitacji nie tylko nie uzdrowi nauki w Polsce, ale może zniszczyć ostatnie oazy dobrej naukowej roboty. Istotną cechą życia naukowego w Polsce jest fizyczny bezruch. Młody uzdolniony człowiek, podejmujący studia na polskim uniwersytecie, jeśli tylko wybierze karierą naukową, prawdopodobnie przejdzie na tym samym uniwersytecie na emeryturę. Jeśli będzie bardzo dobry i będzie w dobrym ośrodku, zapewne po drodze zaliczy staż za granicą i to wszystko. Jego rówieśnicy w USA i w Europie Zachodniej wielokrotnie zmienią miejsce pobytu, wielokrotnie przystąpią do konkursu na stanowisko, o które ubiegać się będą dziesiątki lub nawet setki osób. Będą oceniani przez różne gremia, złożone z osób, które nie są z nimi i z ich promotorami powiązane układami. W Polsce w wielu instytucjach zrobienie doktoratu gwarantuje zatrudnienie w macierzystej placówce, a tam, gdzie etatów jest mało, o zatrudnieniu często decyduje siła promotora lub nawet koneksje rodziców, a nie dorobek naukowy. Zniesienie habilitacji tego nie zmieni.

Panowie Wójcik i Wieczorek publikują nie tylko w „Polityce". W Internecie można znaleźć ich artykuły, przedstawiające obraz polskiego środowiska pełnego ba-

ronów, którzy dzięki tytułom naukowym i układom zdobywają wielką władzę i trzęsą swoim otoczeniem ze szkodą dla nauki. Autorzy wierzą, że likwidacja habilitacji i tytułu profesora odbierze baronom władzę. Niestety, są bardzo naiwni. To właśnie habilitacja i tytuł utrudniają baronom życie. Aby powierzyć stanowisko profesora swemu pupilkowi, muszą wcześniej znaleźć co najmniej trzech recenzentów, którzy napiszą pozytywne recenzje, i muszą się liczyć z opinią Centralnej Komisji. Likwidacja habilitacji doprowadzi do tego, że środowisko naukowe zostanie całkowicie zamulone miernotami.

Inny dowód naiwności autorów to propozycja „podniesienia poziomu doktoratów", który jest zdecydowanie za niski. Autorzy nie precyzują, czy powinno to nastąpić poprzez wymuszenie środkami administracyjnymi, czy też byłoby skutkiem presji środowiska. Metody administracyjne źle się nadają do kontrolowania poziomu badań naukowych i byłyby bardzo kosztowne. Apele o podniesienie poziomu mogłyby ewentualnie odnieść skutek tam, gdzie poziom już teraz jest wyższy niż gdzie indziej. Tam, gdzie jest źle, na pewno się nie poprawi. A jak rozwiązać problem praw nabytych? Czy wymagać więcej już od tych, którzy kończą pisanie rozprawy doktorskiej, czy tylko od tych, którzy właśnie rozpoczęli studia doktoranckie? Gdy kilkanaście lat temu zlikwidowano tytuły profesora zwyczajnego i nadzwyczajnego oraz wprowadzono jeden tytuł profesora, zakładano, że wymagania do nowego tytułu będą wyższe od wymagań do tytułu profesora nadzwyczajnego. W praktyce okazało się, że obecny profesor to tylko dawny profesor nadzwyczajny. Na całym świecie poziom doktoratów zależy od uniwersytetu. Wiem, że inny jest w École Normale Supérieure, a inny w Université de Provence, inny w Stanford, a inny w Kent State. Każdy może to sprawdzić.

Zmienić habilitację!

Bronię habilitacji, uważam jednak, że można i trzeba ograniczyć jej znaczenie i zmienić jej charakter. Jedynym przywilejem doktorów habilitowanych powinno być prawo uczestniczenia w procedurach nadawania stopni naukowych. Po to, by decydować o nadaniu stopnia doktora i doktora habilitowanego, trzeba mieć wiedzę i doświadczenie nieco szersze niż ma świeżo upieczony doktor. Nie wiem natomiast, dlaczego tylko doktor habilitowany może być opiekunem magistranta, i nie znam żadnych przyczyn, dla których szkoła wyższa kształcąca magistrów (a tym bardziej szkoła oferująca studia licencjackie) powinna zatrudniać co najmniej ośmiu profesorów na każdym kierunku.

Jestem belfrem prawie czterdzieści lat i nie zetknąłem się z żadnym dowodem na to, że – z wyjątkiem zajęć, których celem jest przygotowanie najlepszych studentów do pracy naukowej – fakt posiadania habilitacji miał pozytywny wpływ na jakość pracy dydaktycznej. Oczywiście jest pewna korelacja pomiędzy inteligencją i sukcesami w pracy naukowej, więc wśród dobrych nauczycieli są także doktorzy habilitowani, ale znam bardzo wiele osób, które doskonale uczą i wcale nie zamierzają się habilitować. Znam też profesorów zwyczajnych, którzy uczyć nie powinni.

Zniesienie wymagań dotyczących kadry samodzielnej na studiach licencjackich, a nawet magisterskich, co najmniej częściowo zlikwidowałoby wieloetatowość, jedną

z najbardziej dotkliwych patologii w życiu szkół wyższych. Oba projekty ustaw nawiązują do tego problemu, ale proponowane w nich rozwiązania niczego nie zmienią.

Niektórzy zwolennicy zniesienia habilitacji argumentują, że na czas „robienia habilitacji" zawiesza się badania naukowe. Nie wykluczam, że jest tak w niektórych środowiskach i dlatego należy zdecydowanie zmienić formułę habilitacji, podobnie jak zrobiono już we Francji i w Niemczech. Nie jest bowiem prawdą, że zniesiono tam habilitację. Zmieniono tylko jej formę. We Francji i w Niemczech od wielu lat nie ma już rozprawy habilitacyjnej. Decyzję o nadaniu odpowiednika habilitacji podejmuje się na podstawie oceny dorobku naukowego. Żadna rozprawa nie jest potrzebna. Jedyny dodatkowy wysiłek to przygotowanie przekonującego opisu dorobku naukowego. Tam, gdzie prawdziwe konkursy są standardem, umiejętność pochwalenia się swoimi wynikami jest cenna. Oczywiście wymagania dotyczące dorobku mogą być różne, w zależności od dyscypliny. Wymaganie od historyka literatury, żeby napisał monografię wydaje się dość naturalne, bywa jednak absurdalne w stosunku do chemika.

Uczony w USA poddawany jest surowej ocenie po pierwszych sześciu latach pracy na stanowisku profesora. Jeśli ocena jest pozytywna, oferuje mu się stałą pracę, jeśli nie, musi odejść. Habilitacja bez konieczności pisania rozprawy byłaby pewną namiastką procedury przyznawania stałej pracy w USA. Różnica polegałaby tylko na tym, że w Polsce decyzję zatwierdzałaby CK.

To, że stopień doktora habilitowanego byłby nadawany na podstawie dorobku, prawdopodobnie wyeliminowałoby żenujące wymagania dotyczące oświadczeń o wkładzie autorów prac wspólnych. Warto takie wymagania wyeliminować także w przypadku rozpraw doktorskich. W ciągu ostatnich kilku lat starannie przejrzałem ponad setkę wyróżniających się rozpraw doktorskich z informatyki, napisanych głównie w czołowych uniwersytetach amerykańskich. Znakomita większość z nich zawierała wyniki uzyskane we współpracy z promotorem. Rezultaty wielu z tych prac były znane i cytowane jeszcze przed zakończeniem przewodu doktorskiego. Nikogo to nie gorszy, a znakomicie ułatwia młodym doktorom karierę naukową.

Biurokracja

W późny sobotni wieczór 7 sierpnia, w samym środku wakacji, moja skrzynka pocztowa wypełniła się listami członków Komitetu Informatyki PAN, popierających protest rektora AGH, profesora Ryszarda Tadeusiewicza, przeciwko propozycji Rady Głównej Szkolnictwa Wyższego (RGSW), która w efekcie mogła doprowadzić do likwidacji studiów informatycznych na politechnikach. Problem wziął się stąd, że Rada Główna zajęła się porządkowaniem listy kierunków studiów. Mają zostać wyodrębnione grupy (pewnie 10), a w każdej grupie kilka lub kilkanaście kierunków (najlepiej po 10), powstanie charakterystyka kierunku, sylwetka absolwenta, definicja minimum programowego oraz zasady kształcenia w trybie zawodowym. W ten sposób zbuduje się coś na kształt ustawy zasadniczej, regulującej sposób zdobywania wiedzy na 100 lub może 104 kierunkach. Oczywiście, jak można przeczytać na stronie RGSW, „w sytuacji jasno określonej potrzeby wynikającej z rozwoju badań naukowych" możliwe będzie wyodrębnienie nowego kierunku studiów.

Centralizację decyzji dotyczących programów studiów uważam za absurd, który projekt prezydencki jeszcze wzmacnia. Przyszła ustawa, podobnie jak obecna, da uczo-

nym prawo kierowania uczelnią i zarządzania jej finansami. Wyłonionemu w demokratycznych wyborach rektorowi, który wcześniej był ornitologiem lub mediewistą, i na ogół nie ma pojęcia ani o finansach, ani o zarządzaniu, powierza się kierowanie przedsiębiorstwem zatrudniającym kilka tysięcy ludzi, którego budżet wynosi setki milionów i które świadczy usługi dla dziesiątek tysięcy studentów. Ta sama ustawa nie daje Radzie Wydziału, w której składzie jest kilkudziesięciu kompetentnych matematyków, w tym kilku wybitnych, prawa ułożenia programu studiów matematycznych.

Kolejny absurd to wiązanie kierunków studiów z dziedzinami nauki, tak jakby podstawowym zadaniem uniwersyteckiej dydaktyki było kształcenie kadr naukowych. Za absurdalne uważam samo pojęcie „sylwetki absolwenta". Nowoczesne studia powinny być elastyczne i to student powinien sam, poprzez podejmowane w trakcie studiów decyzje, kształtować swoją „sylwetkę".

Po ostatniej rekrutacji na studia wyższe liczba studentów w Polsce zbliżyła się do 2 milionów. Studiować będzie tysiące młodych matematyków, geografów, chemików, historyków literatury, prawników, etnografów (lista kierunków może się oczywiście zwiększyć „w sytuacji jasno określonej potrzeby wynikającej z rozwoju badań naukowych"). Przez pięć lat, za spore pieniądze, państwowe lub rodziców, zdobywać będą specjalistyczną wiedzę, napiszą prace magisterskie na ezoteryczny temat, po to, by potem znaleźć się na rynku pracy potrzebującym zaledwie kilku dziesiątek etnografów, tyluż historyków średniowiecza, może setek matematyków. Zaledwie kilka procent absolwentów specjalistycznych studiów wykorzysta istotną część swojej wysoko specjalistycznej wiedzy.

Młodzi ludzie zapisują się na studia kierunkowe, bo tylko takie studia są. Nie dlatego, że w przyszłości chcą zajmować się historią literatury lub matematyką, lecz dlatego, że chcą skończyć studia, a potem znaleźć pracę. I trafiają na rynek, gdzie – z punktu widzenia pracodawcy – ich dyplom dokumentuje tylko to, że prawdopodobnie są trochę bardziej inteligentni od przeciętnego Polaka, że zapewne mają trochę ogłady i ... niewiele więcej. W znacznej większości będą wykonywać pracę, która nie ma żadnego związku z ich wykształceniem.

W USA masowe studia wyglądają zupełnie inaczej. Studiuje się na uniwersytecie, a nie na przykład na etnologii. Po ukończeniu czteroletnich studiów w college'u uzyskuje się tytuł (BA lub BS) i specjalizację (major). Specjalizacja w zakresie informatyki oznacza na przykład, że w trakcie studiów około 40% punktów „kredytowych" zdobyło się za przedmioty informatyczne i wypełniło się pewne absolutnie minimalne minima programowe. Pozostałe punkty można zdobyć zaliczając zajęcia na temat komunikacji interpersonalnej (bardzo potrzebna prawie w każdym zawodzie), ekonomii (bo przecież znaczny procent informatyków będzie pracował dla banków), malarstwa (bo jak robić strony internetowe nie czując kolorów), czy tańca (bo lubimy tańczyć). Przeciętny absolwent takich studiów lepiej się sprawdzi na rynku pracy od kogoś, kto zdobył 90% punktów zaliczając programowanie kwantowe, neuronowe, genetyczne, obiektowe, imperatywne, funkcjonalne, deklaratywne, logiczne, współbieżne, wielowątkowe, gridowe (i jeszcze aspektowe, ostatnio bardzo modne).

Konieczne jest rozerwanie więzów zmuszających do studiowania ustalonych dyscyplin naukowych. Trzeba stworzyć możliwość studiowania w systemie zbliżonym do anglosaskiego, to znaczy nie w instytucie, lecz na uniwersytecie. Takie studia będą

tańsze i dadzą większości absolwentów lepsze gwarancje uzyskania zatrudnienia. Uważam, że należy stworzyć co najmniej możliwość powstania takich studiów równolegle do istniejących. Powinny to być studia masowe, w odróżnieniu od proponowanych elitarnych studiów międzywydziałowych, których celem ma być wykształcenie uczonych przygotowanych do prowadzenia badań interdyscyplinarnych. Po to, by takie studia zostały dobrze przyjęte przez potencjalnych studentów i pracodawców, należy je dobrze „sprzedać". Zaczynając na przykład od prowadzenia studiów wspólnie z uniwersytetami amerykańskimi i oferując dyplom podkreślający, że są to studia w systemie anglosaskim. Tak jak zrobiono ze studiami MBA (Master of Business Administration), które odniosły wielki sukces rynkowy.

Oba projekty przewidują studia dwustopniowe, których wprowadzenie ma być ważnym elementem procesu bolońskiego. Właśnie masowe studia dwustopniowe prowadzone są w systemie anglosaskim, gdzie ogromna liczba studentów zdobywa tytuł bakałarza (licencjata), a tylko niewielu zapisuje się na studia zawodowe (professional) lub magisterskie i doktoranckie (graduate). Jest przyjęte, że nie kontynuuje się studiów na tej samej uczelni, przejście na wyższy poziom oznacza więc zmianę miejsca studiowania i może też oznaczać wybór lub zmianę specjalizacji. System anglosaski pozostawia taką możliwość, gdyż jest ogromnie elastyczny.

Absolwent naszych studiów licencjackich jest niedokończonym socjologiem, trochę historykiem lub ćwierćmatematykiem, a wymagania stawiane przy przyjęciach na uzupełniające studia magisterskie (zaliczone 60% standardów) sprawiają, że polski licencjat może kontynuować studia tylko w wybranej uprzednio specjalności. Ponadto niska wartość rynkowa półspecjalistów powoduje, że wszyscy chcą kontynuować naukę na studiach magisterskich, do tego na tej samej uczelni (jeśli to tylko możliwe). W zasadzie dla studenta uczelni, oferującej studia magisterskie, dwustopniowość oznacza jedynie dodatkowy egzamin po trzecim roku studiów. A mówi się, że ma promować zmianę uczelni i ewentualnie specjalizacji, tak jak to jest w USA.

Do Bolonii czy do Berkeley

Szkolnictwo wyższe w Polsce, podobnie jak w innych krajach europejskich, boryka się z wieloma problemami. Z niektórymi poradziliśmy sobie lepiej niż inni. Możemy prowadzić selekcję kandydatów na studia, możemy zakładać, i zakładamy, prywatne uniwersytety.

Jednym z poważnych problemów, którego nie mają inne kraje europejskie (przynajmniej w „starej" Europie) jest brak kadry dydaktycznej w wielu rynkowych dziedzinach: w informatyce, neofilologiach, pedagogice, socjologii, oraz niski poziom badań naukowych w tych dziedzinach. Z informacji, które podał profesor Andrzej Kajetan Wróblewski w rozmowie z Andrzejem Hołdysem („Gazeta wyborcza", 28.07.04), wynika, że fizyka, matematyka i historia są w Polsce uprawiane na dobrym poziomie i zajmują w rankingach ISI, odpowiednio, 12, 13 i 15 miejsce na świecie. Sytuacja informatyki, która zajmuje 27 miejsce, jest zła, sytuacja pedagogiki zaś – tragiczna. Znalazła się w siódmej dziesiątce.

Z drugiej strony, na pedagogice w roku akademickim 2002/2003 studiowało ponad 146 tysięcy studentów, na informatyce ponad 62 tysiące, na historii 25 tysięcy, na matematyce ponad 18 tysięcy, a na fizyce niecałe 10 tysięcy. Sytuacja się nie

poprawia i dysproporcje pomiędzy potrzebami dydaktycznymi a liczbą nauczycieli rosną. W latach 2000–2002 z pedagogiki nadano 34 habilitacje, z informatyki – 32, natomiast z historii – 139, a z fizyki – 113. Nie robi się nic, żeby przyspieszyć tempo wzrostu kadry nauczającej na kierunkach rynkowych. Także nowa ustawa nie stwarza mechanizmów, które mogłyby pomóc w rozwiązaniu tego problemu.

Jedną z przyczyn, dla których autorzy projektów ustawy o szkolnictwie wyższym odbierają uczelniom prawo ustalania programów nauczania, jest próba dostosowania się do wymagań Unii Europejskiej, a może, jak niektórzy sądzą, zdobycie władzy za pomocą unijnego straszaka. Jednym z celów procesu bolońskiego jest doprowadzenie do wzajemnej uznawalności dyplomów, co będzie możliwe, jeśli zrówna się poziom nauczania na wszystkich uniwersytetach europejskich. Raport Komisji Europejskiej, zatytułowany *Rola uniwersytetów w Europie wiedzy*, wyraża niepokój spowodowany różnorodnością europejskich uniwersytetów, nie tylko pomiędzy poszczególnymi państwami, „ale także wewnątrz każdego z krajów". Jest to, zdaniem autorów raportu, fakt niepożądany.

W USA różnice pomiędzy uniwersytetami są ogromne. Powszechnie wiadomo, jak bardzo Uniwersytet Stanforda różni się od UC w Santa Cruz. O University of Mass-Lowell mało kto słyszał. Ja do dziś nie słyszałem. Uniwersytety, które wymieniam, to, odpowiednio, 1, 50 i 98 na liście rankingowej wydziałów informatyki. Wszystkie te uniwersytety należą do kategorii „research university", których w USA jest nieco ponad 120. Oprócz tego jest około 450 uniwersytetów nadających stopień doktora, a wszystkich uniwersytetów jest ponad 4000. Różnice między tymi z pierwszej setki, a tymi z ostatniej są zapewne takie, jak pomiędzy jakością życia w Szwajcarii i Korei Północnej. Różnice te dotyczą wszystkiego: rynkowej wartości dyplomu, jakości kadry i budżetu. Uniwersytet Stanforda jest bogaty (na same badania otrzymuje z zewnątrz blisko miliard dolarów rocznie), uniwersytety z końca pierwszej setki są już biedne. Nie stać ich na istotne dofinansowywanie badań. Uniwersytety z końca listy żyją skromnie z niskiego czesnego, płacąc niskie pensje kadrze, której jakość nie jest wysoka. System amerykański jest efektywny, bo pieniądze i talent kumulują się w kilku miejscach, co daje olbrzymią masę krytyczną o wielkim potencjale.

Autorzy projektu ustawy chcieliby, aby w Polsce dyplom magistra miał taką samą wartość niezależnie od tego, która uczelnia go wydała. W końcu nie jest to dyplom Uniwersytetu Jagiellońskiego czy Uniwersytetu Warszawskiego, lecz dyplom państwowy (art. 155). Państwowa Komisja Akredytacyjna ma stać na straży tego, żeby się ktoś ani w górę, ani w dół nie wychylił.

W USA nie ma państwowych tytułów zawodowych. Moje, wprawdzie dość ograniczone, kontakty z prywatnymi pracodawcami, zatrudniającymi informatyków w Polsce i w Europie, pozwalają mi mniemać, że to, czy kandydat na pracownika ma tytuł magistra, jest dla nich znacznie mniej ważne niż to, co on umie. A jeśli w ogóle oglądają dyplomy, to sprawdzają, kto je wystawił.

Próby zrównania poziomu uniwersytetów są skazane na niepowodzenie. Jeśli nie wprowadzi się rejonizacji i nie zlikwiduje konkursowego trybu przyjmowania kandydatów na atrakcyjne kierunki, co na szczęście raczej nam nie grozi, uzdolniona młodzież dalej będzie garnęła się do dobrych szkół, a dobre szkoły będą z tego korzystać i wybierać najlepszych kandydatów.

Dyplomy mają różną wartość nie tylko dlatego, że na przykład na Uniwersytecie Warszawskim kadra nauczająca jest lepsza niż w Prywatnej Wyższej Szkole Informatyki w Pcimiu Dolnym. Także dlatego, że na studia informatyczne na Uniwersytecie Warszawskim na 120 miejsc przyjmuje się około 40 laureatów olimpiad przedmiotowych, a pozostałe 80 osób wybiera się spośród dwóch tysięcy bardzo dobrych kandydatów. Podobnie jest na całym świecie. Absolwentów informatyki Rice University ceni się wyżej niż absolwentów University of Illinois w Urbanie, mimo że wydział informatyki w Urbanie jest chyba lepszy. Po prostu warunki finansowe i system rekrutacji w Rice powodują, że studiuje tam uzdolniona młodzież. We Francji absolwenci Grandes Écoles są lepsi od absolwentów uniwersytetów, mimo że wiele francuskich uniwersytetów ma doskonałą kadrę.

WIEDZA CZY NARZĘDZIA? EWOLUCJA POLSKICH PROGRAMÓW NAUCZANIA OD TEORII DO PRAKTYKI

GRZEGORZ RAWICZ-MAŃKOWSKI

SAS Institute Polska

Rozwój edukacji w Polsce doprowadził do wzrostu liczby instytucji niepaństwowych zajmujących się szkolnictwem wyższym w Polsce.

Rektor uczelni, której żaden absolwent nie jest bezrobotny (Rektor PJWSTK dr Jerzy Paweł Nowacki) podczas inauguracji roku akademickiego 2004/2005 zwrócił uwagę, iż przed 1989 rokiem, gdy Polska nie miała dostępu do najnowocześniejszych technologii, a opóźnienie technologiczne wynosiło 15–20 lat w stosunku do USA, Japonii i Europy Zachodniej, polscy naukowcy – informatycy świadomie skoncentrowali się na teorii informatyki, aby nie pozostać w tyle za resztą świata. Czy po 15 latach wolnego rynku, nauka potrafiła wypracować mechanizmy, które skutecznie pozwolą jej zmniejszać przewagę konkurencyjną uczelni zachodnich? Czy potrafiła zaadoptować zachodnie wzorce zarządzania poparte technologią komputerową, czy też nadal głównymi narzędziami są tablica i kreda, a komputer jest wykorzystywany jako maszyna do pisania i kalkulator?

Czy uczelnie wyższe przygotowane są na współpracę z dostawcami IT, którzy »¿ dysponują niosą ze sobą nie tylko najnowsze technologie, ale najnowocześniejsze wzorce zarządzania oraz efektywnego wykorzystania zasobów? Czy uczelnie wyższe wykorzystują możliwości stwarzane przez firmy zachodnie, czy próbują jedynie wprowadzić wschodni model – „daj, a ja wydam"? Na ile programy nauczania na kierunkach informatycznych odstają od poziomu nauczania na uczelniach w USA, Europie Zachodniej i Japonii? W jakim stopniu biurokracja i ukryte bezrobocie w państwowych instytucjach jakimi są uczelnie spowolniają rozwój polskiej nauki? Wydaje się, iż z problemem tym poradziły sobie uczelnie prywatne.

Jeśli w Unii Europejskiej co czwarte nowo tworzone miejsce pracy związane jest z informatyką, to kogo kształcą polskie uczelnie? Czyżby aż tak dobrze wyczuły koniunkturę, iż za 5–10 lat proporcje te rzeczywiście się odwrócą?

Na liście 400 najlepszych uczelni na świecie znalazły się tylko dwie uczelnie z Kraju nad Wisłą, a i to dopiero pod koniec czwartej setki.

Wszyscy dobrze zdają sobie sprawę z tego, iż nauka polska jest niedoinwestowana. Na ile jednak niedoinwestowanie wynika z braku środków, a na ile ze złego

sposobu ich wykorzystania i wadliwych systemów leżących u podstaw edukacji na poziomie wyższym w Polsce? Wszak nie sztuką jest otrzymać wsparcie finansowe i jakoś je wydać. Sztuką jest stworzenie wartości dodanej dla uczelni i dla partnera biznesowego. Tylko wtedy uczelnie traktowane będą jak partnerzy, a nie jak wiecznie głodne dzieci. Czy uczelnie Polskie dorosły do poziomu twardej gry rynkowej, poziomu, na którym od lat grają zachodnie koncerny, a także firmy, jakimi są uczelnie amerykańskie i do którego to poziomu dążą uczelnie europejskie? Podczas tegorocznej gali wręczania statuetek Najlepszy Partner w Biznesie, Rektor WSPiZ prof. Andrzej Koźmiński odbierając statuetkę w imieniu uczelni, zaznaczył, że jest to potwierdzenie, że jego uczelnia jest partnerem biznesowym dla innych instytucji i tak chce być postrzegana.

RACJONALNE ZARZĄDZANIE IRRACJONALNYMI ZACHOWANIAMI

RAFAŁ STEFAŃSKI

Zanim zaczniemy snuć mniej lub bardziej racjonalne wywody na temat racjonalnego zarządzania, a zarządzania irracjonalnymi zachowaniami w szczególności, wypada zacząć od określenia co przez racjonalność i racjonalne będzie rozumiane. Wedle encyklopedii PWN racjonalność to „cecha świadomej działalności ludzkiej polegająca na dobieraniu odpowiednich środków do osiągnięcia zamierzonych celów". Rozważając racjonalność zachowań w organizacji, od razu natrafimy na parę problemów związanych z cytowaną definicją:

- O czyje cele chodzi?
- Kto dobiera środki realizacji celów?
- Co to znaczy „świadomie"?

A wreszcie:

- Jakie konsekwencje dla racjonalności ma społeczny, grupowy charakter działań w organizacji?

Z punktu widzenia menedżerów (najwyższego szczebla) na powyższe pytania odpowiedzieć dość łatwo: chodzi o cele firmy (obiektywne, definiowane przez top management w konsultacji z niższymi szczeblami bądź bez tej konsultacji); środki muszą być dobierane w jednostkach organizacyjnych stosownie do zadań przydzielonych tym jednostkom; świadomie oznacza obiektywnie, rozsądnie, z uwzględnieniem konsekwencji, w oparciu o twarde dane itd.; praca zespołowa daje lepsze efekty niż indywidualizm (więc podnosi obiektywizm, zwiększa liczbę pomysłów, punktów widzenia – słowem zwiększa racjonalność decyzji). Jednocześnie menedżerowie często narzekają na nieracjonalność działań swoich podwładnych. Przyjrzyjmy się czy te narzekania są słuszne i z czego mogą wynikać?

Definiowanie celów, czyli „nie jestem nieracjonalny, to po prostu nie moja racjonalność"

Kiedy jestem indywidualnym przedsiębiorcą, sam ustalam swoje cele i zasady działania. Kiedy, razem z kolegami, współtworzę zespół lub spółkę, długo dyskutujemy co i w jaki sposób chcemy osiągnąć. Kiedy zatrudniam się w firmie, zazwyczaj myślę przede wszystkim o zarobkach, warunkach pracy, własnej karierze... – to są

cele, do których osiągania dobieram właściwe środki. A więc moje zachowania mogą być całkowicie racjonalne z mojej perspektywy i zarazem nieracjonalne (niepożądane) z punktu widzenia firmy. Kiedy dochodzę do wniosku, że mój dodatkowy wysiłek nie będzie dostrzeżony, nagrodzony czy doceniony, a z celami firmy się nie identyfikuję – czyż jest czymś nieracjonalnym, gdy nie przykładam się do dodatkowych obowiązków (np. wprowadzania danych do systemu CRM, albo dostarczania informacji zewnętrznym konsultantom). Jeżeli na dodatek rodzi się we mnie poczucie zagrożenia związane z ewentualnym sukcesem zmian organizacyjnych (np. scentralizowanie kontaktów z klientami w call-center), to opóźnienie zmian wręcz wydaje się leżeć w moim interesie. Z takiej perspektywy może nie ma co dziwić się postawom przywodzącym na myśl Luddystów, niszczących u progu rewolucji przemysłowej maszyny parowe jako zabierające pracę ludziom. W krótkiej perspektywie indywidualnych interesów (niemłodzi już fachowcy rzemieślnicy) mieli rację – rozpowszechnianie maszyn parowych nie było korzystne, tak jak nie są korzystne – ciągle z perspektywy ludzi zagrożonych zwolnieniem, marginalizacją – fuzje, przejęcia, reorganizacje dnia dzisiejszego. W dłuższej perspektywie zmiany techniczno-organizacyjne były korzystne dla społeczeństwa, szczególnie jeżeli rozpatrywać tę korzyść z punktu widzenia nowych wartości ukształtowanych w czasie i w wyniku rewolucji techniczno-organizacyjnej.

W przypadku rzeczywistego konfliktu interesów trudno mówić, by potrzeby jednej ze stron były nieracjonalne – ich realizacja może być niemożliwa, ale to inna sprawa. A więc zarządzania tak rozumianą nieracjonalnością należałoby szukać w sferze efektywnej komunikacji, zarządzania konfliktem, negocjacji, mediacji.

Jeżeli zaś nieracjonalność bierze się z subiektywnie spostrzeganej różnicy interesów, albo z różnic w wartościach? W tym ostatnim przypadku można raczej mówić o irracjonalności niż nieracjonalności – zarządzanie tym obszarem wymaga budowy i uzgadniania kultury organizacyjnej, a także rekrutowania i kształcenia ludzi dopasowanych do tej kultury.

Dobieranie środków czyli „zachowuję się nieracjonalnie, bo nie wiem jak działać racjonalnie"

Nieznajomość procedur, brak doświadczeń, brak umiejętności uniemożliwią racjonalne działanie nawet najbardziej zidentyfikowanemu z firmą pracownikowi. Po prostu nie będzie trafnie dobierał środków do realizacji celów choćby nie wiem jak się z nimi zgadzał. To banalna przyczyna nieefektywnych zachowań. Edukacja, edukacja i jeszcze raz edukacja to banalna recepta. Na tym mógłby się skończyć niniejszy akapit, ale warto kwestię edukacji – zwiększania trafności w dobieraniu środków ująć nieco szerzej.

Przyjrzyjmy się jakże rozpowszechnionej i cenionej koncepcji jaką jest Zarządzanie Przez Cele (Management by Objectives). Ciekawe jest, że u źródeł tej koncepcji był sprzeciw wobec „zurzędniczenia" i biurokratyzacji pracy menedżerów. Teoria mówi: zdefiniujmy (wspólnie) cele, przekażmy odpowiedzialność za ich realizację podwładnemu (wyzwanie, nadanie sensu) i dajmy mu dowolność w doborze środków (przedsiębiorczość). Brzmi to jak handlowe porozumienie wolnych, kompetentnych przedsiębiorców w celu realizacji wspólnego zamysłu. W praktyce często wygląda to nieco inaczej: cele są narzucane z góry, odpowiedzialność jest przekazywana w dół

(to jedyny element MBO wdrażany bardzo konsekwentnie) i niech sobie radzą. Takie zwulgaryzowane podejście do Zarządzania Przez Cele nie tylko tworzy poczucie rozbieżności interesów między szefem (firmą) i podwładnym, ale też zwalnia szefa (firmę) z odpowiedzialności za edukację pracowników.

Wspólne działania, czyli „to nie ja jestem nieracjonalny, to my jesteśmy nieracjonalni"

Załóżmy, że nasz zespół (firma) gromadzi ludzi zidentyfikowanych z nadrzędnymi celami i zarazem znakomicie wykwalifikowanych. Świadomi celów nadrzędnych, wspólnie definiują cele cząstkowe, które realizują z użyciem optymalnych środków wybranych dzięki ich wiedzy, umiejętnościom i doświadczeniu. Cóż, zebrać i zbudować taki zespół nie jest łatwo, ale być może jest to recepta na racjonalność. Badania nad zespołami wskazują, że zrozumienie celów, stałe ich potwierdzanie, skierowanie wysiłków w jedną stronę, nabór, docenianie i nagradzanie za wyniki, wiedzę i kompetencje są kluczowymi czynnikami skuteczności zespołów. Ale niewystarczającymi, gdyż czasem włączają się dwa mechanizmy, które takie znakomite zespoły rozregulowują.

Mechanizm pierwszy to realizacja przez członków grupy tak zwanych celów drugorzędnych, wynikający z ambicji osobistych i relacji między współpracownikami. Jeżeli nasz wspólny cel jest dla mnie najważniejszy, zaharowuję się żeby go osiągnąć, czy jest coś złego w tym że chcę by mnie szanowano, doceniono, że chciałbym żebyśmy osiągnęli sukces dzięki mnie. Promowanie własnych pomysłów, zawłaszczanie zadań najbardziej widocznych i prestiżowych, wykazywanie że się ma rację, zdobywanie atrakcyjnych, wyrazistych ról zespołowych (przywódcy, którego wszyscy słuchają, celnego krytyka, któremu wszystko wolno, wesołego kumpla, którego wszyscy lubią), budowa koalicji, kontrolowanie informacji i zasobów – to nasze codzienne arsenały w grze grupowej. Wszystkie te bitwy osłabiają racjonalność decyzji i działań zespołów. Z drugiej strony można powiedzieć, że jeżeli ludzie są świadomi co robią i wybierają realizację celów drugorzędnych (może wtedy już pierwszorzędnych) miast celów zespołu to postępują racjonalnie. Jeżeli są świadomi..., a najczęściej nie są. Budowanie zespołów, „czyszczenie relacji", pomoc pracownikom w znalezieniu własnej roli w zespole, treningi pracy zespołowej lub nawet treningi interpersonalne będą stosownymi środkami dla przeciwdziałania tym zjawiskom.

Drugi mechanizm jest bardziej paradoksalny. O ile pierwszy można było nazwać osłabieniem spójności grupowej, to drugi mechanizm wiąże się z nadspójnością. Najsławniejszym przykładem zjawiska zwanego przez psychologów społecznych Syndromem Grupowego Myślenia (Groupthink) było zaplanowanie przez sztab prezydenta Kennedy'ego – grupę wybitnych ekspertów o szerokich i komplementarnych kompetencjach – tzw. inwazji w Zatoce Świń. Źle dobrani i źle wyekwipowani ludzie, którzy mieli obalić Castro, wylądowali w źle wybranym miejscu, a plan awaryjny był niewykonalny. Przypomnijmy, że twórcy planu byli wybitnymi ekspertami a nie półgłówkami. Co więc zaszło? Irving Janis, który analizował mechanizmy decyzyjne prowadzące do fiaska w Zatoce Świń oraz wiele innych podobnych nieracjonalnych działań, znalazł w wielu grupach znaczące podobieństwa – charyzmatyczny lub co najmniej silny przywódca, poczucie elitarności grupy i wyższości nad otoczeniem, bardzo wysoki wzajemny szacunek prowadzący do rozproszenia odpowiedzialności i konformizmu (skoro

wybitny kolega tak mówi..., jeżeli wybitny kolega nie protestuje...). Te grupy po prostu się nakręcały, ludzie przejściowo tracili zdrowy rozsądek. Praca z nadspójnym zespołem jest większym wyzwaniem niż ze skonfliktowaną grupą. Analitycy polecają różne procedury zmniejszające izolację (i wywyższanie się grupy), spowalniające decyzje, wymuszające konfrontację między członkami zespołu, słowem procedury, które chronią grupę przed zbyt dobrym samopoczuciem.

Świadomość, czyli „jestem nieracjonalny, bo jestem człowiekiem"

Wreszcie ostatnia grupa czynników powodujących znaczące odchylenia od racjonalności – natura ludzka. Nie alienacja od celów, nie brak wiedzy, nie walka lub naciski grupy tylko emocje i umysł. Roli emocji nie trzeba omawiać. Każdy z nas pamięta sytuacje, w których był na tyle zdenerwowany, przestraszony, zły lub na odmianę szczęśliwy, że mu „na chwilę rozum odjęło".

Ciekawsze są pułapki umysłu ludzkiego. Eksperymenty psychologów wykazują niezbicie, że większość z nas (matematycy i statystycy też) w codziennym funkcjonowaniu gwałci rozmaite zasady racjonalności. Nasza ocena prawdopodobieństwa zdarzeń zależy od konsekwencji tych zdarzeń dla nas (generalnie prawdopodobieństwo zdarzeń dobrych przeceniamy). Zdarzenia mało prawdopodobne uznajemy za niemożliwe, a wysoce prawdopodobne za pewne. Przyczyny własnych sukcesów widzimy w sobie, przyczyny własnych porażek w otoczeniu (a w przypadku innych ludzi często odwrotnie). Mamy lekką ale wyraźną skłonność do ryzyka, gdy grożą nam straty oraz jesteśmy konserwatywni (mamy awersję do ryzyka), gdy na horyzoncie widać wyraźne zyski. Jeden z psychologów tropiących te ułomności ludzkiego umysłu – Daniel Kahneman, laureat nagrody Nobla – twierdzi, że przyczyna większości nieudanych fuzji (a większość jest nieudanych), nietrafionych inwestycji, nieoszacowanych projektów tkwi w decydentach a nie warunkach otoczenia. Menedżerowie grzeszą nadmiernym optymizmem, zawyżoną oceną (siebie, zespołu, firmy) oraz pogardą dla choćby prostej statystycznej analizy doświadczeń innych firm. Poza tą właśnie analizą statystyczną, „spojrzeniem z zewnątrz" umiejscawiającym własną firmę, zespół, projekt na tle otoczenia, wprowadzaniem roli adwokata diabła w dyskusjach, niewiele jest metod radzenia sobie z własnym umysłem. Na pewno punktem wyjścia do zarządzania tego typu nieracjonalnymi zachowaniami jest zrozumienie i pogodzenie się z własną – gatunkową – ułomnością.

Postscriptum – zwycięzców się nie sądzi, czyli pochwała irracjonalności

Ostatnia grupa przyczyn nieracjonalnych zachowań zasługuje na szczególny suplement. Warto się zastanowić dlaczego ludzie są tak często nieracjonalni. Czy nieracjonalność jest tylko błędem ewolucyjnym czy może pełni jakieś istotne funkcje w naszym życiu? W perspektywie indywidualnej nieracjonalność daje potrzebne do życia nadzieję, optymizm i energię. Ilu młodych sportowców znalazłoby siłę na morderczy trening po starannej analizie statystycznych szans zdobycia wyśnionego złotego medalu olimpijskiego, ilu zdecydowałoby się założyć rodzinę, gdyby nie wierzyli, że szanse jej szczęśliwej trwałości są wyższe niż te statystyczne kilkadziesiąt

procent. Psychologowie mówią o życiodajnej roli nieracjonalności, o „złudzeniach, które pozwalają żyć". Czyż nie podobnie jest z zespołami i firmami? One też nie mogą czerpać energii z mdłej racjonalności. Collins i Porras w swojej książce *Wizjonerskie organizacje* wskazują, że jedną z cech wielkich firm jest podejmowanie wyzwań, na myśl o których racjonalnym analitykom powinny wypaść wszystkie włosy. Bo czyż racjonalnym było wejście Boeinga na nowy rynek (samoloty pasażerskie) z nowatorskim produktem (pierwszy pasażerski samolot odrzutowy – 707), czyż racjonalne było przyjęcie zamówienia na samolot, który wystartuje na pasie o długości 1600 metrów (za krótki dla ówczesnych odrzutowców), pokona dłuższy dystans niż konkurenci i będzie miał sześć foteli w rzędzie (model 727)? Warto dodać, że fiasko każdego z owych projektów mogło pogrążyć firmę. I co? Boeing podjął ryzyko, wycisnął wodę z kamienia i wygrał. Ile firm „przekraczających" racjonalne decyzje padło – autorzy nie piszą.

MIERZĘ WIĘC JESTEM

NIE MOŻNA ZARZĄDZAĆ CZYMŚ, CZEGO NIE DA SIĘ ZMIERZYĆ

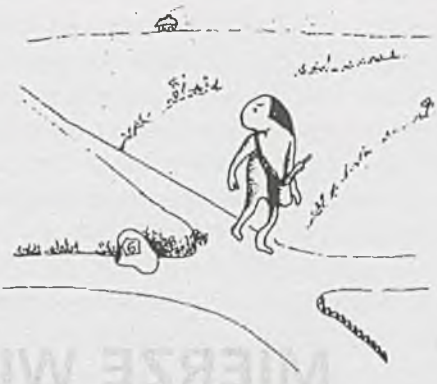
Zamiast wprowadzenia

Bromba nie posiada specjalnie niepokojącego wyglądu, nie wyróżnia się też niczym, co by przykuwało uwagę na pierwszy rzut oka. Jest nieco większym od wiewiórki stworzonkiem o różowym futerku, a na odrobinę za bardzo wystającym brzuszku nosi dużą torbę wypełnioną różnymi niezbędnymi przedmiotami, jak to: lornetką, wagą, kalendarzykiem, siedmioma termometrami, zegarem, metrem krawieckim oraz dużym złoconym dyplomem ozdobionym takim napisem:

Niniejszym zaświadcza się, że Bromba jest upoważniona do mierzenia i ważenia oraz
podawania wyników, ponieważ ukończyła kurs i okazała się bardzo precyzyjna.

podpisano:
Centralny Urząd do Mierzenia i Ważenia

Wszystkie informacje o Brombie za:
M. Wojtyszko: *Bromba i inni (po latach także)*, Agencja Edytorska Ezop, Warszawa, 2003.



Już tyle lat wędruję i
przebyłam wiele dróżek,
ale brakuje miary mi:
jak zmierzyć me podróże?
Czy kalendarzem, kiedy znów
kolejna kartka spadła,
czy też ilością miłych snów
od Pciucha i Fikandra?
Czy może sumą wszystkich sum
podanych przez maszynki
lub (jak doradzał pewien Fum)
policzyć odpoczynki?
Może uśmiechy wszystkich paszcz
dodawać w jeden szereg,
a może właśnie każdy płaszcz
powinien mieć numerek?
Wspólnych wyników moich tras
(chociaż nie chciałam wierzyć)
nie ma. I nie ma nawet szans,
by wszystko naraz zmierzyć.
Trudno obliczyć życia szlak,
lecz błądzą ci, co chcą, by
zwątpić w skądinąd znany fakt
nieomylności Bromby!

* * *

Trudno jest wyobrazić sobie cywilizację (a tym bardziej naszą branżę) bez miar. Czasami, o czym mówią załączone teksty, trudno jest także mierzyć. Znajmy więc miarę.

Zapraszam do lektury.

Michał Goliński

O INFRASTRUKTURZE TECHNICZNEJ SPOŁECZEŃSTWA INFORMACYJNEGO

czyli na czym stoimy (jedną nogą)

MAREK ŚREDNIAWA

Instytut Telekomunikacji
Politechnika Warszawska
mareks@tele.pw.edu.pl

1. Wstęp

Dzięki szerokiemu rozpowszechnieniu technik informacyjnych – IT (*Information Technology*) rozumianych jako metody elektronicznego przetwarzania, gromadzenia, przesyłania i dystrybucji informacji możemy obecnie snuć różne dywagacje wokół tzw. *społeczeństwa informacyjnego* (SI)[1]. W związku z tym prowadzone są badania i opracowywane metody identyfikacji zjawisk i wymiernej oceny efektywności działań związanych z wykorzystaniem IT i sieci w gospodarce i życiu codziennym. Problemami wysokiego poziomu, takimi jak: definicja wskaźników charakteryzujących stopień rozwoju SI i ich wykorzystanie do analiz sytuacji w Polsce i innych krajach oraz metodami pomiaru efektywności realizacji projektów informatycznych zajmują się pozostali współautorzy bloku tematycznego „Mierzę więc jestem". W niniejszej części uwaga została skupiona na sposobach i miarach oceny wydajności infrastruktury technicznej SI, która stanowi bazę materialną do budowy systemów, aplikacji i usług służących do realizacji różnych celów. Jakość i stopień rozwoju infrastruktury technicznej jest jednym z zasadniczych składników wpływających na wartość różnych wskaźników gotowości do funkcjonowania w sieci, np. NRI (*Network Readiness Index*), służących do pomiaru zaawansowania rozwoju SI.

Zauważamy, że żyjemy w coraz bardziej cyfrowym świecie, w którym wszelkie rodzaje informacji – głos, dane, programy, teksty, muzyka, filmy, obrazy mogą być

1. Autor nie jest osobiście wielkim zwolennikiem terminu „społeczeństwo informacyjne" uważając, że koncepcja jest nieco naciągana i dęta, podobnie zresztą jak i pokrewna idea „gospodarki opartej na wiedzy" (jakby istniała alternatywna „gospodarka oparta na niewiedzy"). Nie ulega wątpliwości, że komputery i sieci są znaczącymi wynalazkami istotnie wpływającymi na styl życia i gospodarkę, ale wcześniej podobnie znaczącą zmianę spowodowały takie nowości jak np.: samochód, samolot, radio i telewizja, wówczas jednak chyba nikt nie mówił o „społeczeństwie motoryzacyjnym" czy „audiowizualnym". Podsumowując wydaje się, że właściwszym i ogólniejszym terminem oddającym istotę sprawy jest, po prostu, „społeczeństwo post-przemysłowe".

wyrażone w postaci sekwencji bitów. Taka forma reprezentacji informacji może być poddawana przetwarzaniu, przechowywaniu i przesyłaniu. Rodzi to naturalne pytania o miary wyrażające sprawność operowania informacją, które pozwalają porównywać:

- wydajność przetwarzania informacji,
- zdolność do tymczasowego przechowywania informacji i jej trwałego gromadzenia,
- szybkość przesyłania informacji.

Oczywiście ważne są również koszty przetwarzania, przechowywania i przesyłania informacji. Mogą one w zasadniczy sposób wpływać na koncepcję projektowanych systemów informatycznych i teleinformatycznych. W dalszym ciągu uwaga zostanie skupiona na miarach dotyczących wydajności przetwarzania.

2. Pomiary w informatyce

Na początku sprawa wydawała się dość prosta – komputer, który był w stanie wykonać więcej operacji na jednostkę czasu był uważany za wydajniejszy. Troszkę tak jak w przypadku samochodów gdzie za szybszy uważano samochód, który rozwijał wyższą szybkość maksymalną. Stopniowo w miarę rozwoju informatyki, podobnie jak w przypadku przemysłu motoryzacyjnego, mieliśmy do czynienia z coraz większą specjalizacją systemów komputerowych związaną z różnymi obszarami ich wykorzystania, takimi jak np.:

- obliczenia numeryczne,
- modelowanie i symulacja złożonych procesów (chemicznych, fizycznych),
- rozpoznawanie i przetwarzanie obrazów,
- gry komputerowe,
- prognozowanie pogody,
- obsługa transakcji związanych z korzystaniem z baz danych,
- sterowanie procesami w czasie rzeczywistym,
- serwery WWW i wiele innych.

Podobnie jak w przypadku samochodów stało się jasne, że jeden parametr nie wystarczy i że kryteria definiujące miary wydajności należy dobierać adekwatnie do zastosowania. W związku z tym do oceny wydajności systemów komputerowych stosuje się wiele miar, które z czasem i ewolucją architektury komputerów również się zmieniają. Zmiany dotyczą zarówno metodologii pomiaru jak również kryteriów i konstrukcji samych miar.

Historycznie pierwszą miarą była liczba operacji wykonywanych na sekundę przez pojedynczy procesor lub grupę procesorów stanowiących główny element przetwarzający systemu komputerowego. Klasycznymi miarami są:

MIPS (*Million Instructions per Second*) – liczba milionów instrukcji wykonywana na sekundę.

MFLOPS (*Million Floating Point Operations per Second*) – liczba milionów operacji zmiennoprzecinkowych wykonywana na sekundę. W celu ujednolicenia miary przyjęto, że odnosi się ona do operacji mnożenia lub dodawania liczb w reprezentacji zmiennoprzecinkowej 64-bitowej.

Miary te stosowane są do dzisiaj przy ustalania rankingu *Top 500* najwydajniejszych superkomputerów na świecie, przy czym z czasem – prawo Moore'a ciągle działa – przedrostki zmieniły się na *giga* i *tera* i obecnie używamy już GFLOPS (miliardy operacji zmiennoprzecinkowych na sekundę) i TFLOPS (biliony operacji zmiennoprzecinkowych na sekundę), a z czasem będziemy zapewne używać przedrostka *peta* i PFLOPS (tryliony operacji zmiennoprzecinkowych na sekundę).

Wskaźniki MIPS i FLOPS tylko w części spełniają wymienione kryteria[2], więc oprócz nich wprowadzono i stosuje się w praktyce szereg innych miar wydajności, których definiowaniem i modyfikacją zajmują się wyspecjalizowane organizacje, stowarzyszenia lub branżowe grupy przemysłowe.

Miarę FLOPS stosuje się do badania wydajności superkomputerów na podstawie realizacji testu LINPACK (http://www.netlib.org/benchmark/hpl/), który polega na rozwiązaniu układu równań liniowych opisanego macierzą gęstą, za pomocą identycznego algorytmu o złożoności obliczeniowej $2/3n^3 + O(n^2)$. Wybrano taki właśnie test, ponieważ jest on szeroko stosowany i jego wyniki dostępne są dla większości superkomputerów oferowanych na rynku. W wynikach podaje się dwa parametry $R_{\max}$ (GFLOPS) – maksymalną osiągniętą wydajność – oraz R_{peak} (GFLOPS) – teoretyczną maksymalną wydajność. Co pół roku publikowana jest lista *Top 500* (http://www.top500.org/) najwydajniejszych komputerów na świecie. Tabela 1 prezentuje aktualną pierwszą dwudziestkę listy *Top 500.*

Ostatnie doniesienia prasowe dają podstawy do przypuszczeń, że przy najbliższej aktualizacji listy nastąpią zmiany w czołówce. Superkomputer firmy *IBM – BlueGene/L* – pokonał wynikiem 36,01 TFLOPS w testach wydajnościowych system *NEC Earth Simulator*, który był dotąd uważany za najpotężniejszy serwer na świecie (informacja z października 2004). Ale dotychczasowy lider *NEC* miesiąc później przedstawił superkomputer SX-8 o wydajności 65 TFLOPS. Japończycy nie zachowają jednak zbyt długo pierwszeństwa, ponieważ firma IBM zapowiada na początek 2005 roku uruchomienie systemu w *Lawrence Livermore National Laboratory* o wydajności ok. 360 TFLOPS – wyścig zatem trwa.

Miara FLOPS nadaje się przede wszystkim do oceny systemów komputerowych przeznaczonych do realizacji złożonych obliczeń numerycznych na dużą skalę, wymaganych zwykle w zastosowaniach wojskowych, inżynierii materiałowej, prognozowaniu pogody czy modelowaniu złożonych zjawisk fizycznych. Nie jest ona adekwatna do wyrażania np. wydajności systemów obsługujących transakcje czy graficznych stacji roboczych wspomagających projektowanie. Z tego powodu w praktyce posługujemy się również innymi miarami, które pozwalają w syntetyczny sposób ocenić wydajność z uwzględnieniem specyfiki zastosowań.

Wiarygodność wyników porównania komputerów o różnych architekturach wymaga odpowiedniego sformułowania miar wydajności i sposobów ich pomiaru. W związku z tym przy ich definicji brano pod uwagę następujące kryteria:

- reprezentatywność – dobór testów do wybranej klasy rzeczywistych zastosowań,

2. Z faktu, że dla dwóch komputerów A i B, $\text{MIPS}(A) > \text{MIPS}(B)$ nie wynika, że czas wykonania wszystkich programów $T(A) < T(B)$. Oprócz szybkości samego procesora w grę wchodzi bowiem wiele dodatkowych czynników, takich jak stopień zrównoleglenia operacji, organizacja pamięci, wydajność operacji we/wy, itd.

Tabela1. „Top 20" z listy „Top 500" najwydajniejszych komputerów

	INSTYTUCJA; KRAJ I ROK PRODUKCJI	NAZWA KOMPUTERA; LICZBA PROCESORÓW; PRODUCENT	R_{max} R_{peak} [TFLOPS]
1	Earth Simulator Center; Japonia – 2002	Earth-Simulator; 5120; NEC	35,860 40,960
2	Lawrence Livermore National Laboratory; USA – 2004	Thunder Intel Itanium2 Tiger4 1.4GHz, Quadrics; 4096; California Digital Corporation	19,940 22,938
3	Los Alamos National Laboratory; USA – 2002	ASCI Q – AlphaServer SC45, 1.25GHz; 8192; HP	13,880 20,480
4	IBM – Rochester; USA – 2004	BlueGene/L DD1 Prototype (0.5GHz PowerPC 440 w/Custom); 8192; IBM/LLNL	11,680 16,384
5	NCSA; USA – 2003	Tungsten PowerEdge 1750, P4 Xeon 3.06GHz, Myrinet; 2500; Dell	9,819 15,300
6	ECMWF; Wielka Brytania – 2004	eServer pSeries 690 (1.9GHz Power4+); 2112; IBM	8,955 16,051
7	Institute of Physical and Chemical Research (RIKEN); Japonia – 2004	RIKEN Super Combined Cluster; 2048; Fujitsu	8,728 12,534
8	IBM – Thomas Watson Research Center; USA – 2004	BlueGene/L DD2 Prototype (0.7GHz PowerPC 440); 4096; IBM/LLNL	8,655 11,469
9	Pacific Northwest National Laboratory; USA – 2003	Mpp2 Integrity rx2600 Itanium2 1.5GHz, Quadrics; 1936; HP	8,633 11,616
10	Shanghai Supercomputer Center; Chiny – 2004	Dawning 4000A, Opteron 2.2GHz, Myrinet; 2560; Dawning	8,061 11,264
11	Los Alamos National Laboratory; USA – 2003	Lightning Opteron 2GHz, Myrinet; 2816; Linux Networx	8,051 11,264
12	Lawrence Livermore National Laboratory; USA – 2002	MCR Linux Cluster Xeon 2.4GHz, Quadrics; 2304; Linux Networx/Quadrics	7,634 11,060
13	Lawrence Livermore National Laboratory; USA – 2000	ASCI White, SP Power3 375MHz; 8192; IBM	7,304 12,288
14	NERSC/LBNL; USA – 2002	Seaborg SP Power3 375MHz 16 way; 6656; IBM	7,304 9,984
15	NCSA; USA – 2004	TeraGrid, Itanium2 1.3; 1.5GHz, Myrinet; 1776; IBM	7,215 10,259
16	Lawrence Livermore National Laboratory; USA – 2003	xSeries Cluster Xeon 2.4GHz, Quadrics; 1920; IBM/Quadrics	6,586 9,216
17	Lawrence Livermore National Laboratory; USA – 2004	Lilac xSeries Xeon 3.06GHz, Quadrics; 1540; IBM	6,232 9,425
18	HPCx; Wielka Brytania – 2004	eServer pSeries 690 (1.7GHz Power4+); 1600; IBM	6,188 10,880
19	Grid Technology Research Center, AIST; Japonia – 2004	AIST Super Cluster P-32, Opteron 2.0GHz, Myrinet; 2200; IBM	6,155 8,800
20	Oak Ridge National Laboratory; USA – 2004	Cray X1; 504; Cray Inc.	5,895 6,451

Źródło: http://www.top500.org/

- przenośność – oprogramowanie wykorzystywane do testów wydajności powinno być napisane w językach programowania wysokiego poziomu,
- łatwość modyfikacji – uwzględnienie zmian i postępu w zakresie algorytmów oraz metod technologii,
- powtarzalność wyników pomiarów,
- obiektywność – uniezależnienie od presji producentów,
- liniowość – zapewnienie zgodności z intuicją,
- prostota – zapewnienie łatwości realizacji pomiaru.

Problem niepełnej reprezentatywności miar wydajności jest również widoczny z perspektywy użytkowników komputerów osobistych. Rozważając zakup nowego komputera jako jeden z istotnych parametrów bierze się pod uwagę częstotliwość taktowania zegara procesora. Jednak różnice architektoniczne między różnymi modelami procesorów, np. firmy *Intel* – *Pentium*, *Celeron*, *Centrino*, a także między procesorami Firm *Intel* i *AMD* sprawiają, że kryterium częstotliwości zegara jest niewystarczające i nie w pełni określa rzeczywistą wydajność systemów.

W związku z potrzebą stosowania miar wydajności dobranych do zastosowań podjęto prace nad ich zdefiniowaniem. Jedną ze znanych instytucji zajmujących się tą problematyką jest SPEC – *Standard Performance Evaluation Corporation* (http://www.spec.org/). Organizacja opracowała cały zestaw miar podzielonych na kategorie reprezentujące typowe zastosowania bądź aspekty działania komputerów:

- Wydajność procesorów.
- Wydajność usług sieciowych (web services). Pomiar skalowalności platform aplikacyjnych dla przedsiębiorstw (np. J2EE i .NET).
- Wydajność aplikacji graficznych.
- Wydajność systemów do obliczeń równoległych i rozproszonych.
- Wydajność serwerów aplikacji Java.
- Wydajność serwerów pocztowych.
- Wydajność sieciowych systemów plikowych.
- Wydajność serwerów WWW.

Dla każdej z wymienionych kategorii prowadzone są rankingi, a same miary są modyfikowane bądź dodawane nowe, gdy pojawią się nowe aplikacje.

Inną ważną organizacją zajmującą się pomiarami wydajności komputerów do zastosowań komercyjnych jest *TPC – Transaction Processing Performance Council* (http://www.tpc.org). Jej członkami są czołowi producenci sprzętu i oprogramowania (m.in. *AMD*, *BEA*, *Bull*, *Dell*, *Fujitsu*, *HP*, *Hitachi*, *IBM*, *Intel*, *Microsoft*, *NEC*, *Novell*, *Oracle*, *Sun*, *Sybase*, *Teradata*, *Unisys*). TPC opracowało następujący zestaw wskaźników charakteryzujących wydajność systemów transakcyjnych:

- TPC-C – OLTP (*On Line Transaction Processing*): przeznaczony do pomiaru wydajności systemów przetwarzających transakcje w czasie rzeczywistym, np. systemów baz danych. Miarą wydajności jest liczba transakcji zrealizowanych w ciągu minuty *tpmC*.
- TPC-H – dla systemów wspomagających podejmowanie decyzji z zapytaniami ad hoc. Miarą wydajności jest obsługiwana w ciągu godziny liczba zapytań zbiorczych – *QphH@Size* (*Composite Query-per-Hour Performance Metric*) do bazy danych o określonym rozmiarze (*Size*).

- TPC-R (*Decision Support for Business Reporting*) – dla systemów wspomagających podejmowanie decyzji w sprawozdawczości. Miarą wydajności jest obsługiwana w ciągu godziny liczba zapytań zbiorczych – *QphR@Size* (*Composite Query-per-Hour Performance Metric*) do bazy danych o określonym rozmiarze (*Size*).
- TPC-W – dla systemów wspomagających handel elektroniczny w Internecie. Miarą wydajności jest *WIPS@scale* – (*Web Interactions Per Second*) liczba obsługiwanych interakcji na sekundę przy założeniu określonej liczby towarów (*Scale*).

Do przeprowadzania pomiarów wydajności każdego z wymienionych wskaźników używa się znormalizowanego obciążenia symulującego typowe warunki eksploatacji systemu. Uwzględnia ono skalę aplikacji i typowe profile ich wykorzystywania. Dla wszystkich miar dodatkowo rozpatruje się także wskaźniki pochodne określające koszt jednostkowy transakcji.

3. Podsumowanie

Przytoczone wcześniej porównanie pomiarów wydajności komputerów do oceny samochodów okazuje się być dość trafną analogią. W zależności od wymagań, preferencji, posiadanych środków możemy wybrać rozwiązanie najbliższe naszym potrzebom, ale musimy wziąć pod uwagę wiele wskaźników. Odwołanie się do jednego tylko kryterium wydajności, np. FLOPS czy szybkości maksymalnej, nie daje nam jednoznacznej odpowiedzi, który produkt jest „lepszy" i za pomocą którego szybciej osiągniemy cel. Musimy zatem używać wielu miar, dobierać je adekwatnie do kontekstu i przypisywać im subiektywne nawet czasem wagi. Z doświadczenia wiemy, że wybierzemy inne konfiguracje komputerów do pracy biurowej, a inne do graficznego projektowania inżynierskiego czy gier multimedialnych. Przemysł motoryzacyjny również oferuje nam pełne spektrum produktów od bolidów F1, poprzez samochody sportowe i terenowe, po zwykłe osobowe i ciężarowe. Ciekawostką integrującą oba nurty i puentującą całość jest pojawienie się na rynku nowej rodziny wydajnych komputerów przenośnych, *Acer Ferrari 3000*, ze znakiem wierzgającego czarnego konia na żółtym polu i charakterystycznym jaskrawo czerwonym kolorem obudowy.

MIARY PROCESU I PRODUKTU PROGRAMOWEGO

ANDRZEJ KOBYLIŃSKI

Szkoła Główna Handlowa w Warszawie
Katedra Informatyki Gospodarczej
andrzej.kobylinski@sgh.waw.pl

If you don't know where you are, a map won't help.
Watts Humphrey

1. Wstęp

Kluczowym elementem każdego procesu inżynierskiego są pomiary. W przeciwieństwie do innych dyscyplin inżynierskich, inżynieria oprogramowania nie opiera się na ścisłych prawach teorii pomiarów. Ale ścisłe pomiary, do których przyzwyczaiła nas fizyka, takie jak na przykład pomiary odległości, napięcia, masy, czy temperatury, nie są powszechne w przypadku produkcji oprogramowania.

Z literatury znane są setki różnych miar oprogramowania (patrz np. Zuse 1991 [1], Simmons i in. 1998 [2], Fenton & Pfleeger 1997 [3], Kan 2002 [4]). Tymczasem praktycznie jedynym pytaniem dotyczącym pomiarów oprogramowania, jakie nasuwa się przeciętnemu użytkownikowi oprogramowania jest: ile ono kosztuje? W przypadku twórców oprogramowania można w tym zakresie liczyć na nieco bardziej wnikliwe zapytanie, rzeczywiście odnoszące się do pomiarów oprogramowania: jaka jest jego długość (wielkość)?

Tradycyjnie, mówiąc o pomiarach oprogramowania, rozróżnia się dwa podstawowe aspekty: pomiar procesu wytwórczego (z dalszym podziałem na pomiar procesu i pomiar przedsięwzięcia) i pomiar produktu informatycznego:

- procesu programowego, czyli ogółu czynności realizowanych w organizacji wytwórczej, związanych z wytwarzaniem oprogramowania (miary procesu),
- konkretnego przedsięwzięcia informatycznego, w celu lepszej estymacji czasu i kosztu, kontroli jakości realizowanych zadań, oceny produktywności i, ogólnie, poddanie przedsięwzięcia kontroli (miary przedsięwzięcia),
- produktu programowego – zarówno oceny jego jakości, jak również innych, niejakościowych cech produktu (miary produktu).

2. Miary procesu programowego

Pomiary procesu programowego mają dla organizacji wytwarzającej oprogramowanie znaczenie strategiczne. Ich zastosowanie pozwala wejrzeć w proces programowy realizowany w organizacji. Pozwala to kierownictwu, ale również pracownikom technicznym, ocenić, które procesy w firmie realizowane są poprawnie, a co do których można mieć zastrzeżenia. Miary procesu powinny być zbierane i uśrednione ze wszystkich realizowanych przedsięwzięć w długim okresie czasu.

Do oceny procesu programowego w ostatnim 20-leciu powstało kilka metod, które zyskały szerszą popularność: Capability Maturity Model (CMM) (w kilku wariantach), ISO/IEC 15504, Bootstrap, Trillium, kryteria nagród jakości itp. Oprócz wspomnianych już przyczyn powstania, miały one pomóc organizacjom poszukującym dostawcy oprogramowania, w wyborze najlepszego kontrahenta. Firmy zajmujące się wytwórstwem oprogramowania starają się uzyskać maksymalnie wysoką ocenę i w ten sposób przekonać potencjalnego odbiorcę o tym, że są w stanie dostarczyć zamówione oprogramowanie na czas, w ramach budżetu, o dobrej jakości. Ocena taka może być dokonywana na poziomie całej organizacji (Software-CMM, CMM Integration – Staged Representation, Trillium) lub poszczególnych procesów szczegółowych realizowanych w firmach i obszarów działania (CMMI – Continuos Representation, ISO/IEC 15504 (SPICE), Bootstrap, kryteria nagród jakości).

Problemem otwartym i wymagającym dalszych badań pozostaje kwestia, czy poprawny, na wysokim poziomie kompetencji realizowany proces wytwórczy skutkuje wytworzeniem lepszego produktu. Obecnie wydaje się udowodnione, że dojrzałe organizacje, osiągające wysoki poziom doskonałości, są w stanie z wyższym prawdopodobieństwem dostarczyć zamówione oprogramowanie na czas i w ramach budżetu. Kwestią nie do końca rozstrzygniętą pozostaje jakość dostarczonego oprogramowania – brakuje przekonujących danych empirycznych, a w szczególności studiów porównawczych, które dowiodłyby, że organizacje znajdujące się na wyższym poziomie dojrzałości są w stanie dostarczyć lepsze jakościowo oprogramowanie, niż firmy znajdujące się na niższym poziomie kompetencji. Szczególnie w wypadku małych firm wytwarzających oprogramowanie, zorganizowanych na zasadach przyjacielskich; skupiających ludzi przepojonych wspólną ideą; o dużym potencjale intelektualnym, umiejętnościach i doświadczeniu, wysoka ocena dokonana przy pomocy któregoś z wymienionych modeli, wcale nie jest niezbędna. Mimo chaosu i braku określonych procedur pracy firmy takie są w stanie dostarczyć oprogramowanie o wysokiej jakości.

3. Miary przedsięwzięcia informatycznego

Pomiary procesu programowego analizowane nie na poziomie całej organizacji, ale pojedynczych realizowanych przedsięwzięć pozwalają kierownictwu projektu: ocenić stan realizacji przedsięwzięcia, śledzić potencjalne zagrożenia, oceniać, które obszary mogą stać się krytyczne, przydzielać pracownikom zadania, ocenić zdolność zespołu wytwórczego do wytworzenia produktu programowego o odpowiedniej jakości.

Ale jednocześnie nie można zapomnieć o aspekcie ekonomicznym realizowanego przedsięwzięcia – konieczna jest systematyczna kontrola kosztów i czasu wykonywanego projektu.

Dlatego też wydaje się niezbędne, by podczas realizacji przedsięwzięcia kierownictwo mierzyło i kontrolowało:

1. wielkość powstającego produktu (mierzoną punktami funkcyjnymi, liczbą linii kodu lub innymi metodami),
2. dotychczasowy koszt realizacji,
3. dotychczasowy czas trwania przedsięwzięcia,
4. pracochłonność (mierzoną za pomocą osobomiesięcy),
5. jakość.

Te 5 miar podstawowych powinno być stosowane we wszystkich fazach cyklu życia oprogramowania, ale oprócz nich można również wykorzystywać inne, specyficzne dla tego właśnie etapu, miary (patrz np. Kobyliński 1998 [5], Bluemke 1999 [6]).

4. Miary produktu programowego

Jak już wspomniano, w minionych latach powstały setki miar produktu programowego. Niestety, trudno powiedzieć, by któraś z nich zyskała powszechną aprobatę. O ile punkty funkcyjne zyskały sporą popularność, głównie ze względu na możliwość ich obliczenia na stosunkowo wczesnym etapie cyklu życia oprogramowania i niezależność od języka, to wartości pozostałych miar produktu daje się wyznaczyć obiektywnie i algorytmicznie dopiero po powstaniu kodu. A co więcej, tak obliczone miary, nie dość, że są dostępne stosunkowo późno w cyklu życia, to dodatkowo trudno przy ich pomocy ocenić to, co jest najistotniejsze dla ostatecznego odbiorcy, a mianowicie jakość użytkową (ang. *quality in use*) oprogramowania.

W obszerny sposób problem ten ujmuje niepopularna, niestety, w naszym kraju norma międzynarodowa ISO/IEC 9126, dotycząca jakości produktu programowego. Jej pierwsza wersja powstała w 1991 r., ale na język polski nie została przetłumaczona. Wyliczonych w niej było 6 podstawowych cech jakości (w części normatywnej), a uszczegółowiające charakterystyki jakościowe zawarte zostały w dodatku informacyjnym. W 2001 r. norma ta została zaktualizowana: część informacyjna została włączona do części normatywnej. Obecnie norma ta występuje pod numerem ISO/IEC 9126-1:2001. Pochodzący z niej rys. 1 obrazowo wyjaśnia zależności między jakością procesu wytwórczego, jakością produktu programowego i jakością użytkową.

Rysunek 1. Jakość w cyklu życia (Źródło: ISO/IEC 9126-1:2001)

W normie stwierdza się, że dobry jakościowo produkt jest efektem realizacji dobrych jakościowo procesów wytwórczych, a jakość produktu ma wpływ na jakość

użytkową produktu (Sikorski 2000 [7]). Dlatego ocena i poprawa procesu jest środkiem do poprawy jakości produktu, a ocena i poprawa jakości produktu jest środkiem do poprawy jakości użytkowej.

Jakość produktu programowego może być mierzona przy pomocy miar wewnętrznych (zwykle przy pomocy statycznych miar kodu źródłowego programu) lub miar zewnętrznych (zwykle jako zachowanie się programu w czasie jego wykonywania).

W ostatnim czasie opublikowano również 2, 3 i 4 część tejże normy, w których opisano miary wewnętrzne, zewnętrzne i jakości użytkowej produktu programowego.

5. Czy nie za dużo tych miar?

Wertując literaturę z obszaru pomiarów oprogramowania, przeglądając normę 9126, napotykamy na olbrzymie bogactwo sugerowanych miar. Jest ich tak wiele, że organizacja zainteresowana zastosowaniem pomiarów, może być przytłoczona obszernością koniecznego do zgromadzenia materiału. W takim przypadku, gdy organizacji nie stać lub z innego powodu nie jest zainteresowana gromadzeniem tak dużego materiału do analizy, można zasugerować zastosowanie paradygmatu zaproponowanego przez Victora Basiliego (Basili 1984 [8]), o nazwie GQM (ang. *Goal – Question – Metric* / Cel – Pytanie – Miara).

Polega ono na tym, że na wstępie organizacja musi sformułować główny cel (cele) do osiągnięcia (ang. *Goal*). W kolejnym kroku należy postawić pytania, na które odpowiedzi pozwolą określić stopień osiągnięcia celu (ang. *Question*). Odpowiedzi na te pytania powinny mieć charakter mierzalny. Dlatego z każdym pytaniem należy związać jedną lub kilka miar, dla których obliczenie ich wartości pozwoli odpowiedzieć na pytania (ang. *Metric*). Zastosowanie takiej procedury pozwala na ograniczenie się do kilku / kilkunastu miar.

6. Czy modele oceny procesów programowych wymagają dokonywania pomiarów?

I na koniec warto jeszcze wspomnieć o tym, jak poszczególne modele oceny procesów programowych odnoszą się do kwestii pomiarów.

W modelu CMM dopiero od organizacji znajdującej się na 4 poziomie (zarządzanym), wymaga się realizacji skutecznego programu pomiarowego. Na tym poziomie wszystkie przebiegające w organizacji procesy powinny być na bieżąco mierzone i przebiegać w wytyczonych granicach. Organizacja musi ustanowić zbiór mierzalnych celów jakościowych, i to zarówno dla procesów, jak i produktów. Wszystkie przedsięwzięcia realizowane w organizacji, wszystkie procesy, przebiegają pod kontrolą procesu pomiarowego. Mierzona jest produktywność pracowników, jak i różne aspekty jakościowe produktów. W ogólnoorganizacyjnej bazie gromadzone są wszystkie dane potrzebne do dalszych analiz.

W modelu Trillium jeden z 28 przewodników (odpowiedników procesów głównych) w całości zadedykowany jest pomiarom. Zakłada się, że pomiarom podlegać powinny: produkt, przedsięwzięcie, procesy, popełnione błędy, zadowolenie klientów. Poszczególne wymagane przez model praktyki rozłożone są na poszczególnych poziomach, tak więc już organizacja na drugim poziomie jest zmuszona do realizacji 11 najprostszych praktyk pomiarowych (podczas gdy np. w CMM pomiary wymagane są

dopiero od 4 poziomu). Na wyższych poziomach dojrzałości organizacji wymaga się dokonywania systematycznej analizy danych pomiarowych, włącznie ze statystyczną kontrolą procesu wytwórczego.

W modelu ISO/IEC 15504 proces pomiarowy wyróżniony został jako jeden z 24 procesów głównych.

Norma ISO 9001 zawiera postanowienia dotyczące monitorowania i pomiarów zarówno procesów zarządzania jakością, jak i wyrobów. Wymagania stawiane przez normę przedstawione są wyjątkowo lakonicznie i w sposób bardzo ogólny. Niemniej trudno sobie wyobrazić, by stosowanie metod statystycznych, jakie są wymagane przez standard, było możliwe bez dokonywania pomiarów. Wspomniana lakoniczność odpowiedniego zapisu wymusza zatem wprowadzenie przez organizację programu pomiarowego w zakresie uznanym przez nią za właściwy, nie narzucając konkretnych procedur. Zadaniem pomiarów jest wykazywanie zgodności wyrobu ze specyfikacją, odpowiedniości wykorzystywanego systemu zarządzania jakością oraz skuteczności realizowanego programu doskonalenia, a w szczególności monitorowania zadowolenia klienta. Nic się natomiast w normie nie mówi o pomiarach procesów wytwórczych oraz zarządzania (z wyjątkiem zarządzania jakością).

Prawie wszystkie obszary wyróżniane w kryteriach nagród jakości oceniane są za pomocą miar ilościowych. Wprawdzie miary te nie są odgórnie definiowane, ale oceniana organizacja musi wykazać, że do oceny zadowolenia klientów, satysfakcji osób zatrudnionych, oddziaływania na otoczenie, osiąganego wzrostu itp. stosowany jest odpowiedni zestaw miar. Zwykle około 5% maksymalnej liczby punktów można uzyskać, gdy jest się w stanie udowodnić, że wartości tych miar (dane) są na bieżąco zbierane, przechowywane, integrowane, a następnie analizowane.

Literatura

[1] H. Zuse: *Software Complexity: Measures and Methods*, De Gruyter, Berlin, 1991.

[2] D.B. Simmons, N.C. Ellis, H. Fujihara, W. Kuo: *Software Measurement. A Visualization Toolkit for Project Control and Process Improvement*, Prentice-Hall, 1998.

[3] N.E. Fenton, S.L. Pfleeger: *Software Metrics. A Rigorous and Practical Approach* (2. wyd.), PWS Publishing Company, Boston, 1997.

[4] S.H. Kan: *Metrics and Models in Software Quality Engineering* (2. wyd.), Addison-Wesley, 2002.

[5] A. Kobyliński: *Miary oprogramowania*, Informatyka, nr 11, s. 26–29, 1998.

[6] I. Bluemke: *Miary oprogramowania obiektowego, pozwalające oszacować złożoność oprogramowania*, Informatyka, nr 2, s. 18–21, 1999.

[7] M. Sikorski: *Zarządzanie jakością użytkową w przedsięwzięciach informatycznych*, Monografie 17, Wydawnictwo Politechniki Gdańskiej, Gdańsk, 2000.

[8] V.R. Basili, D.M. Weiss: *A methodology for collecting valid software engineering data*, IEEE Transactions on Software Engineering, t. SE-10, s. 728–738, listopad 1984.

[9] *ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model*
ISO/IEC 9126-2:2003 Software engineering – Product quality – Part 2: External metrics
ISO/IEC 9126-3:2003 Software engineering – Product quality – Part 3: Internal metrics
ISO/IEC 9126-4:2004 Software engineering – Product quality – Part 4: Quality in use metrics.

POMIAR ROZMIARU FUNKCJONALNEGO SYSTEMU INFORMATYCZNEGO – SPOSÓB NA „CHAOS PERMANENS"?

BEATA CZARNACKA-CHROBOT

Szkoła Główna Handlowa
Katedra Informatyki Gospodarczej
e-mail: bczarn@sgh.waw.pl

Not everything that Counts can be Counted,
and not everything that can be Counted Counts.
Albert Einstein

Uwagi wstępne

Z badań Standish Group wynika, że permanentny chaos występujący w projektowaniu systemów informatycznych (SI) – obecnie w USA jedynie 34% tego typu przedsięwzięć kończy się sukcesem, a odsetek ten jest największy w historii badań – to przede wszystkim rezultat nieumiejętnego zarządzania działaniami projektowymi [1]. Jak z kolei słusznie zauważył Tom DeMarco: *nie można prawidłowo zarządzać czymś, czego się nie mierzy* [2]. Skuteczne zarządzanie projektem SI nie jest zatem możliwe bez estymacji i pomiaru rozmiaru SI. Na prawidłowy pomiar i szacowanie wielkości SI pozwalają zaś jedynie odpowiednie metody.

Wobec powyższego celem niniejszego wystąpienia jest ogólne przybliżenie użyteczności metod wykorzystywanych do estymacji i mierzenia tzw. rozmiaru funkcjonalnego SI, a w szczególności mocnych i słabych stron oraz różnic między:

- metodą punktów funkcyjnych IFPUG – najpopularniejszą z technik wymiarowania funkcjonalnego tzw. pierwszej generacji; oraz
- metodą COSMIC-FFP – najnowszą techniką wymiarowania funkcjonalnego i jak dotąd jedyną drugiej generacji.

Obie wyżej wymienione metody należą do tzw. metod ekstrapolacyjnych wymiarowania SI.

1. Dlaczego warto wymiarować SI?

Potrzeba stosowania odpowiedniego modelu wymiarowania SI wynika z następujących faktów:

- potrzeby właściwego planowania projektu SI, tj. oszacowania jego podstawowych parametrów: wielkości, pracochłonności, czasu realizacji, kosztów i produktywności;

- projekt SI to zwykle poważne przedsięwzięcie inwestycyjne: wg danych Standish Group średni roczny koszt projektu SI w USA wynosi ok. 0,7–1,4 mln USD [1];
- realizacja projektu SI angażuje wiele zasobów, a najdroższe są zasoby ludzkie: koszty osobowe to największa składowa budżetu – nawet do 80%, koszty sprzętu to ok. 16–17%, reszta zaś to koszty stałe (utrzymanie i rozwój środowiska pracy);
- o kosztach projektu SI decyduje zatem jego pracochłonność – jej prawidłowe oszacowanie umożliwia więc podjęcie właściwej decyzji inwestycyjnej;
- z tego powodu nakłady pracy to parametr kluczowy przede wszystkim dla odbiorcy SI – jako zasadniczy składnik kosztów powinien bowiem decydować o cenie projektu SI;
- prawidłowa ocena nakładów pracy umożliwia spełnienie zadań związanych z zarządzaniem projektem SI, tj. dostarczenie systemu zgodnego z wymaganiami użytkownika na czas i mieszczącego się w budżecie;
- warunkiem koniecznym prawidłowego określenia pracochłonności jest prawidłowe wyznaczenie wielkości SI;
- potrzeby właściwego kontrolowania projektu SI: pomiar tworzonego SI umożliwia uchwycenie sytuacji niezgodnych z planem;
- potrzeby rozwoju metod wymiarowania SI: pomiar gotowego produktu pozwala na wyprowadzenie zależności wykorzystywanych w przyszłych estymacjach.

W rzeczywistości większość instytucji ma jedynie przybliżone pojęcie o rozmiarze swoich aplikacji, toteż przewidywanie pracochłonności, czasu i kosztów projektu SI jest zadaniem trudnym. To raczej reguła niż wyjątek. Problem pomiaru i szacowania rozmiaru SI stanowi zatem jedną z przeszkód prawidłowego rozwoju projektowania SI jako dziedziny wiedzy i życia. Stąd między innymi wynikają wieloletnie dążenia do stworzenia skutecznych i uniwersalnych metod. Punkt wyjścia to określenie listy ogólnych wymagań, które powinna spełniać odpowiednia metoda pomiaru i szacowania wielkości SI.

2. Jaką metodę wymiarowania SI stosować?

Odpowiednia, czyli użyteczna metoda wymiarowania SI to taka, która:

- pozwala nie tylko na kontrolę, ale również na planowanie projektu SI;
- jest możliwa do zastosowania w roli wiarygodnego estymatora względnie wcześnie w cyklu projektowym – daje przybliżoną orientację co do koniecznych wydatków;
- jest niezależna od stosowanej technologii – możliwość porównywania różnych projektów, od różnych wykonawców lub ich modułów różniących się technologią;
- mierzy pracochłonność wszystkich stadiów projektu;
- mierzy wielkość SI w jednostkach istotnych przede wszystkim dla użytkowników SI;
- sprzyja zaangażowaniu użytkownika w projekt SI, a wg badań Standish Group jest to od lat najważniejszy czynnik sukcesu [1];

- jest zgodna z definicją produktywności: wydajniejszy język programowania zmniejsza nie tylko pracochłonność i koszty całkowite, ale też i jednostkowe oraz zwiększa produktywność działań projektowych [3];
- jest uniwersalna, tj. skuteczna dla wszystkich kategorii SI.

3. Na czym polega podstawowy problem wymiarowania SI?

Jak dotąd nie istnieje metoda wymiarowania projektu SI, która spełniałaby wszystkie wyżej wymienione postulaty. Najbliższa ideału, chociaż niedoskonała, jest ekstrapolacyjna metoda pomiaru i estymacji rozmiaru funkcjonalnego COSMIC-FFP, którą w lutym 2003 zaakceptowano w całości jako standard ISO [4]. Bazuje ona przede wszystkim na metodzie punktów funkcyjnych IFPUG i jej wariantach (np. Mark II, NESMA), również uznanych przez ISO, ale tylko częściowo [5].

Metody ekstrapolacyjne (parametryczne) to techniki wymiarowania projektu SI oparte na empirycznych modelach parametrycznych, które buduje się na podstawie danych pochodzących z wielu przedsięwzięć zrealizowanych w przeszłości – uogólniają one zatem doświadczenia co do zależności pracochłonności (a więc i kosztów) oraz czasu konstrukcji projektu od wielkości SI.

Źródło: T. Koszlajda, *Estymacja projektów informatycznych*, Instytut Informatyki Politechniki Poznańskiej, s. 8.

4. Jak w metodach ekstrapolacyjnych mierzy się rozmiar SI?

Rozmiar SI można wyrazić za pomocą dwóch rodzajów jednostek:

- jednostek programowych: np. liczba linii kodu źródłowego, liczba poleceń;
- jednostek umownych, które dzieli się na:
  - miary złożoności funkcjonalnej SI (ocena funkcjonalności SI): np.
    - *punkty funkcyjne* (ang. *Function Points*),
    - *pełne punkty funkcyjne* (ang. *Full Function Points*),
    - punkty charakterystyczne (ang. *Feature Points*),
    - punkty przypadków użycia (ang. *Use Case Points* – dla metod obiektowych),
    - punkty internetowe (ang. *Internet Points* – dla projektów stron webowych);

- miary złożoności konstrukcyjnej SI (ocena architektury SI):
  - punkty konstrukcyjne, obiektowe (ang. *Object Points*).

Jednostka pomiaru wielkości SI to jedno z kryteriów podziału metod ekstrapolacyjnych. Z tego punktu widzenia wyróżnia się zatem dwa rodzaje takich metod:

- metody oparte na jednostkach programowych (JP):
  - np. SLIM, COCOMO, częściowo COCOMO II;
- metody oparte na jednostkach umownych (JU):
  - metody estymacji i pomiaru tzw. rozmiaru funkcjonalnego (RF) SI:
    - np. metoda punktów funkcyjnych IFPUG i jej warianty, COSMIC-FFP;
  - metody estymacji i pomiaru tzw. rozmiaru konstrukcyjnego SI:
    - np. metoda punktów obiektowych.

5. Na czym polega przewaga metod pomiaru rozmiaru funkcjonalnego?

Przewaga modeli estymacji i pomiaru rozmiaru funkcjonalnego SI:

- nad techniką oceny złożoności konstrukcyjnej wynika m.in. z:
  - większego znaczenia dla użytkownika funkcjonalności SI niźli złożoności jego architektury – z punktu widzenia użytkownika bowiem priorytetowy cel SI to faktyczne wspomaganie realizowanych przez użytkownika funkcji, czyli to funkcjonalność SI ma wpływać na jego konstrukcję, a nie odwrotnie;
- nad metodami opartymi na jednostkach programowych wynika z:
  - zestawienia listy wymagań dla odpowiedniej metody wymiarowania SI z cechami porównywanych rodzajów metod:

Pożądane charakterystyki metod	JP	JU (RF)
(1) nie tylko kontrola, również planowanie projektu SI	nie	tak
(2) możliwość wykorzystania w roli wiarygodnego estymatora na etapie formułowania wymagań	nie	tak
(3) niezależna od stosowanej technologii	nie	tak
(4) pomiar pracochłonność wszystkich stadiów projektu	nie	tak
(5) pomiar wielkość SI w jednostkach istotnych przede wszystkim dla użytkowników SI	nie	tak
(6) sprzyja zaangażowaniu użytkownika w projekt SI	nie	tak
(7) zgodna z ekonomiczną definicją produktywności	nie	tak
(8) uniwersalna, tj. możliwość pomiaru rozmiaru wszystkich kategorii SI	**tak**	**nie**

Wniosek: Wśród modeli ekstrapolacyjnych większą ogólną użytecznością charakteryzują się metody oparte na jednostkach umownych, w grupie takich metod zaś przewagę mają metody estymacji i pomiaru rozmiaru funkcjonalnego SI.

6. Co to jest rozmiar funkcjonalny wg ISO?

Zasadniczą przyczynę małej użyteczności metod bazujących na jednostkach programowych stanowi właśnie wykorzystywana przez nie jednostka. Prosta, „naturalna", precyzyjna i uniwersalna, ale nie uwzględniająca w żaden sposób punktu widzenia użytkownika, w tym funkcjonalności, jaką SI powinien mu dostarczać. Dla niego bowiem funkcjonalność SI, czyli możliwość realizacji przez SI wymaganych funkcji determinująca jego konstrukcję, to parametr decydujący o wartości systemu, a jej miary powinny być niezależne od technicznych parametrów projektowania i implementacji.

Rozmiar funkcjonalny (ang. *Functional Size*) zatem to *rozmiar oprogramowania otrzymany przez ilościowe określenie wymagań funkcjonalnych użytkownika* (ISO/IEC 14143-1: 1998, klauzula 3.6) [6]. Wymagania funkcjonalne użytkownika (ang. *Functional User Requirements* – FUR) zaś *reprezentują praktyki i procedury użytkownika, które oprogramowanie musi realizować w celu zaspokojenia potrzeb użytkownika. FUR wykluczają wymagania jakościowe oraz jakiekolwiek wymagania techniczne* (ISO/IEC 14143-1: 1998, klauzula 3.8) [6].

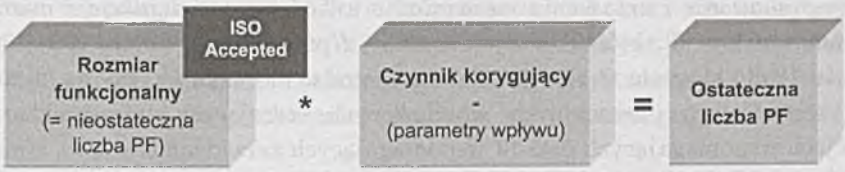
Do pomiaru i estymacji rozmiaru funkcjonalnego SI wykorzystuje się obecnie:

- metody pierwszej generacji: metodę punktów funkcyjnych IFPUG i jej warianty;
- metodę drugiej generacji: COSMIC-FFP.

7. Jakie są słabe strony metody punktów funkcyjnych IFPUG?

Punkty funkcyjne (PF) to *liczba bezwymiarowa, którą znajdujemy jako efektywną relatywną miarę wartości funkcji dostarczanych naszemu klientowi* [7]. W bardzo dużym przybliżeniu, na podstawie doświadczeń można założyć, że 1PF to średnio 8 h pracy w technologii 3GL lub 1,5 h w technologii 4GL.

Rozwojem i ustalaniem standardów metody PF, która w tym roku obchodzi ćwierćwiecze swojego istnienia, od połowy lat osiemdziesiątych zajmuje się organizacja IFPUG (ang. *International Function Point Users Group*). Wprowadzane przez nią udoskonalenia, polegające głównie na obiektywizacji pomiaru w wyniku ciągłych uściśleń różnych niejednoznacznych definicji, doprowadziły w listopadzie 2003 do zaakceptowania metody PF w wersji 4.1 (z 1999 r.) przez ISO (ISO/IEC 20926: 2003), ale tylko w części dotyczącej pomiaru rozmiaru funkcjonalnego (ang. *Unadjusted Function Points*). W kwietniu 2004 r. IFPUG ogłosił wersję 4.2 [8]. Jak dotąd jednak nadal nie ma ona pełnej akceptacji ISO – kontrowersje budzi obliczanie czynnika korygującego na bazie predefiniowanych charakterystyk SI, określających złożoność wymagań technicznych i jakościowych.



Metoda PF spełnia wszystkie – za wyjątkiem jednego – ww. wymagania, a do przyczyn jedynie częściowego jej zaakceptowania przez ISO należy:

- Brak uniwersalności, co uznaje się za jej zasadniczą wadę:

- ze względu na niewystarczające odzwierciedlenie wysokiej złożoności przetwarzania sprawdza się tylko dla SI dostarczających zaprogramowane funkcje, przy których użytkownik wprowadza/otrzymuje dane, sam funkcji nie tworząc;
- może być zatem z powodzeniem stosowana dla SI „sterowanych danymi", czyli wspomagających zarządzanie (ok. 60–80% zasobów SI typowej organizacji komercyjnej);
- praktyka pokazuje, że skuteczność metody PF wobec takich SI jest duża: odchylenie wielkości rzeczywistych od estymowanych (po fazie projektowania, gdy znana jest szczegółowa specyfikacja) oscyluje w granicach ± 10%;
- nie sprawdza się zaś dla: systemów czasu rzeczywistego, programów naukowych, narzędzi, dzięki którym użytkownik tworzy własne funkcje, etc.

- Niepełna lista parametrów wpływu i subiektywność w ocenie ich złożoności – do pojawienia się wersji 4.1 również pomiar rozmiaru funkcjonalnego uważano za subiektywny.

Ww. wady spowodowały pojawienie się wielu wariantów metody PF (obecnie co najmniej 20), których zadaniem miało być głównie rozszerzenie obszaru zastosowania rozmiaru funkcjonalnego na wszystkie (większość) kategorie SI. Najbardziej uznane to:

- metoda PF Mark II – bardzo popularna w Wielkiej Brytanii, również po części zaakceptowana przez ISO, oferuje wyższy poziom szczegółowości pomiaru;
- metoda NESMA (ang. *Netherlands Software Metrics Association*) – holenderska i uproszczona wersja metody IFPUG, też posiada częściową aprobatę ISO;
- metoda tzw. pełnych punktów funkcyjnych w wersji COSMIC-FFP.

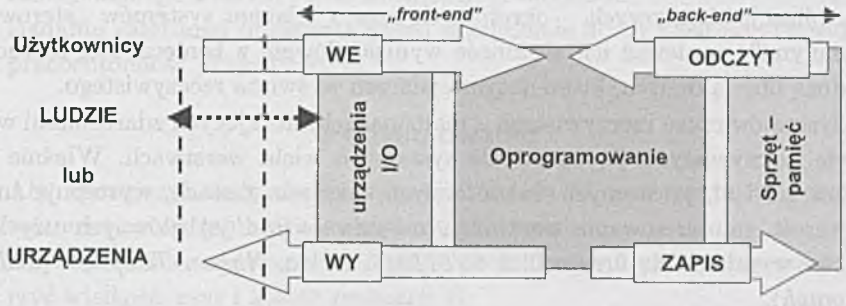
8. Co innego i nowego oferuje metoda drugiej generacji COSMIC-FFP?

Metoda zwykłych PF to efektywna miara rozmiaru funkcjonalnego SI, których użytkownikiem, czyli jedynym odbiorcą funkcjonalności, jest *człowiek*. Inne kategorie SI różnią się od nich m. in. faktem, iż *odbiorcą ich funkcjonalności nie jest człowiek bądź nie jest tylko człowiek*. Dlatego w celu rozszerzenia możliwości pomiaru rozmiaru funkcjonalnego na pozostałe rodzaje oprogramowania w 1997 r. na University of Quebec w Montrealu (UQAM) opracowano koncepcję metody FFP.

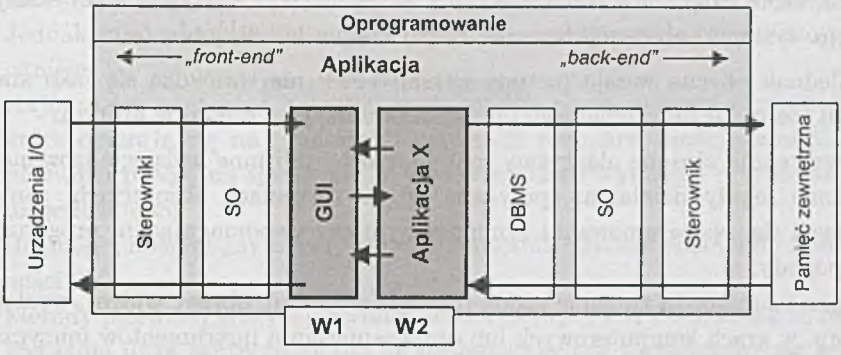
Punktem wyjścia stało się rozszerzenie definicji użytkownika wykorzystywanej przez IFPUG: *Użytkownikiem mogą być nie tylko osoby (specyfikujące FUR), ale też oprogramowanie i urządzenia mechaniczne wchodzące w interakcje z mierzonym oprogramowaniem* [9], której sens jest zgodny z późniejszą definicją ISO (ISO/IEC 14143-1: 1998, klauzula 3.17) [6]. Zatem metoda FFP skupia się na systemach czasu rzeczywistego, operacyjnych, wbudowanych, sterujących, infrastrukturalnych aplikacjach wspomagających oraz SI wspomagających zarządzanie.

Od końca lat dziewięćdziesiątych rozwojem metody FFP zajmuje się organizacja COSMIC (ang. *Common Software Measurement International Consortium*), która za cel postawiła sobie rozwinięcie metody pomiaru rozmiaru funkcjonalnego kolejnej generacji, tj. użytecznej dla wszystkich rodzajów SI oraz zgodnej z wymaganiami

ISO. Po licznych testach, przeprowadzanych zgodnie z wytycznymi ISO w bankowości, lotnictwie, przemyśle samochodowym i w telekomunikacji, oraz po 4 latach starań, metoda COCMIC-FFP zostaje zaakceptowana w lutym 2003 w wersji 2.2 przez ISO (ISO/IEC 19761: 2003) [4]. Jednostką miary jest tutaj Cfsu (ang. *Cosmic functional size unit*), definiowany jako pojedynczy przepływ danych (wejście, wyjście, odczyt, zapis):



Przepływ danych z punktu widzenia funkcjonalnego (COSMIC-FFP); źródło: [4], s. 19.



Przykład alokacji FUR dla oprogramowania wielowarstwowego; źródło: [10], s. 16, [4], s. 20.

Dzięki rozszerzeniu definicji użytkownika metoda COSMIC-FFP daje możliwość pomiaru funkcjonalności każdej tzw. warstwy oprogramowania, którą traktuje się jako *użytkownika* części pozostałej:

- Obecnie nie ma uznanej, jednoznacznej definicji warstwy oprogramowania: intuicyjnie rzecz ujmując jest to *część funkcjonalna oprogramowania operująca na innym poziomie wyodrębnienia funkcji.*
- Powyżej pokazano przypadek, w którym w celu dostarczenia funkcjonalności przez aplikację X (warstwa W2 „klient") konieczne jest wywołanie przez nią GUI (warstwa W1 podporządkowana), czyli wykorzystanie jego funkcjonalności – niewidocznej dla użytkownika-człowieka, który dostrzega jedynie funkcjonalność aplikacji X.
- Dla użytkownika-człowieka zatem funkcjonalność np. GUI, warstw systemów czasu rzeczywistego, oprogramowania sterującego, wbudowanego etc. nie jest bezpośrednio zauważalna, a więc będzie w analizie pomijana.
- Typowe aplikacje biznesowe są z reguły jednowarstwowe.

- Aby jednak dostarczyć funkcjonalność różnym użytkownikom różnych kategorii SI, często należy wykorzystać funkcjonalność innych rodzajów oprogramowania – nie tylko aplikacyjnego. Trzeba ją zatem również mierzyć.

9. Czego metoda COSMIC-FFP w obecnej wersji nie oferuje?

Jak już pokazuje praktyka metoda COSMIC-FFP sprawdza się dla:

- Aplikacji biznesowych – określanych często mianem systemów „sterowanych danymi", ponieważ ich złożoność wynika głównie z konieczności zarządzania dużą ilością danych, które dotyczą zdarzeń ze świata rzeczywistego.
- Systemów czasu rzeczywistego – realizujących/sterujących zdarzeniami w świecie rzeczywistym. Są to zwykle systemy o wielu warstwach. Właśnie w tej kategorii SI, po licznych, zakończonych sukcesem testach, występuje znaczny wzrost zainteresowania omawianą metodą (wśród jej głównych użytkowników wymienia się firmy: *BEA SYSTEMS*, *Nokia*, *Nippon Telephone and Telegraph*).
- Systemów hybrydowych, tj. łączących cechy obu wyżej wymienionych rodzajów SI, które dzięki omawianej metodzie mogą być wymiarowane w tej samej skali, np. systemy rezerwacji w czasie rzeczywistym hoteli, lotów, etc.

Jednak obecna wersja metody COSMIC-FFP nie sprawdza się jako standard pomiaru rozmiaru funkcjonalnego oprogramowania/jego części, w którym:

- występują złożone algorytmy matematyczne lub inne wyspecjalizowane i złożone reguły działania, spotykane np. w systemach eksperckich, samouczących się, oprogramowaniu symulacyjnym czy wspomagającym prognozowanie pogody;
- przetwarzane są zmienne ciągłe (dźwięki audio lub obrazy wideo), występujące np. w grach komputerowych lub oprogramowaniu instrumentów muzycznych.

Jednakże, mimo braku pełnej uniwersalności, nawet taka możliwość skutecznego zastosowania metody jest uznawana za bardzo szeroką, przełomową i unikalną. Ponadto wymienia się następujące słabe strony metody COSMIC-FFP:

- nie są możliwe szybkie, uproszczone obliczenia;
- konieczne są solidnie i mocno uszczegółowione specyfikacje;
- brak spójności w sposobie identyfikacji warstw oprogramowania (def. warstwy);
- mało źródeł poznawczych (przykładów, reguł w zróżnicowanych środowiskach, etc.).

10. W jakim kierunku zmierzają prace nad rozwojem metody COSMIC-FFP?

Bieżącej wersji (2.2) metody COSMIC-FFP nie uznaje się zatem za ostateczną i bezbłędną. Prace zmierzające do jej rozwoju skupiają się w szczególności na:

- Stworzeniu precyzyjnej definicji warstwy oprogramowania i kryteriów ich wyróżniania;
- Testowaniu spójności między różnymi miarami rozmiaru funkcjonalnego SI;

- Stworzeniu zasad i reguł konwersji rezultatów metody PF (w różnych wersjach) na rezultat wyrażony w Cfsu;
- Konstrukcji modeli na podstawie danych o rozmiarach SI wyrażonych w Cfsu;
- Stworzeniu zasad pozwalających na trafniejszą ocenę rozmiaru funkcjonalnego SI w początkowych stadiach jego cyklu życia;
- Studiach nad korelacją występującą pomiędzy oceną eksperta co do rozmiaru funkcjonalnego a rzeczywiście otrzymanym wynikiem;
- Badaniu zależności umożliwiających przeliczenie liczby Cfsu na przewidywaną pracochłonność projektu SI.

Podsumowanie

- Aby zmniejszyć skalę chaosu w projektowaniu SI, konieczny jest wzrost skuteczności zarządzania działaniami projektowymi;
- Aby skutecznie zarządzać działaniami projektowymi, należy estymować i mierzyć wielkość, czas i koszty realizacji SI;
- Aby szacować i wymiarować te parametry właściwie, konieczne jest stosowanie metod spełniających podstawowe wymagania co do ich ogólnej użyteczności;
- Jednakże jak dotąd metoda spełniająca wszystkie zasadnicze wymagania nie istnieje;
- Najbardziej użyteczne z istniejących modeli to te metody ekstrapolacyjne, które opierają się na pomiarze i estymacji rozmiaru funkcjonalnego SI: ich przewaga polega na spełnieniu niemal wszystkich wymagań – brak im jednak uniwersalności;
- Rozmiar funkcjonalny mierzy tylko wymagania użytkownika co do funkcjonalności SI;
- Metody pierwszej generacji wymiarowania SI (np. PF IFPUG, Mark II, NESMA) wykazują dużą skuteczność dla SI wspomagających zarządzanie, ale tylko dla nich;
- Metoda drugiej generacji COSMIC-FFP oferuje nowe spojrzenie na użytkownika SI, przez co umożliwia uchwycenie funkcjonalności każdego ze składników oprogramowania, nawet jeśli dla użytkownika-człowieka jest ona niewidoczna;
- Mimo przełomowego zakresu użyteczności, metoda COSMIC-FFP nie jest idealna – brakuje jej również m. in. pełnej uniwersalności.
- Niewątpliwie jednak ta metoda nowej generacji znacznie przybliża cel trwających od ćwierćwiecza prac nad poszukiwaniem uniwersalnej metody wymiarowania SI.

Metody wymiarowania SI oparte na rozmiarze funkcjonalnym są na świecie coraz chętniej wykorzystywane. Zauważono bowiem, iż warto przewidywać i liczyć wydawane na przedsięwzięcia informatyczne pieniądze, tj. traktować je jak każdą inną inwestycję – szczególnie, że faza entuzjazmu jest już za nimi, a obecnie nadeszła pora na realizm. Ten zmusza zaś firmy informatyczne do coraz większej konkurencji cenowej – aby produkcja oprogramowania była opłacalna, trzeba coraz dokładniej szacować i mierzyć jej koszty. *Branża informatyczna musi wziąć kolejną, gorzką lekcję żelaznych reguł ekonomii, które obowiązują wszystkich bez wyjątku uczestników tej gry* [11].

W Polsce jednak ich wykorzystanie pozostawia wiele do życzenia. Może nie potrafimy uczyć się na własnych błędach? Może nie stać nas na ponoszenie dodatkowych kosztów ich implementacji? A może nie doceniamy ich wagi, w rezultacie czego po prostu ich nie znamy?

A zatem:

- *Wiemy, dlaczego projekty upadają, wiemy jak zapobiec tym upadkom – więc dlaczego one ciągle upadają?* (Martin Cobb)
- Bo: *(...) Wszystkie kłopoty wynikają ze stosowania niewłaściwych metod (albo z pracowania w ogóle bez metod) (...). Innymi słowy, przyczyną marszy ku klęsce jest nasza głupota albo nieudolność.* (Edward Yourdon)
- Jednak idzie ku lepszemu: *People have become much more savvy in project management. When we first started the research, project management was a sort of black art. People have spent time trying to get it right (...).* (Fragment komentarza Jima Johnsona (prezesa Standish Group) do wyników Raportu z 2003 r.)

Literatura

[1] *The CHAOS Chronicles III, Standish Group*, West Yarmouth, Massachusetts, January 2003.

[2] T. DeMarco: *Controlling Software Projects*, New York, Yourdon Press, 1982.

[3] B. Czarnacka-Chrobot: *Porównanie metod pomiaru i szacowania projektów informatycznych – jednostki programowe a jednostki umowne*, w: *Efektywność zastosowań systemów informatycznych*, tom I, materiały konferencyjne XIII Szkoły Górskiej PTI pod red. J.K. Grabary i J.S. Nowaka, WNT Warszawa – Szczyrk 2001, s. 27–55.

[4] *COSMIC-FFP Measurement Manual*, version 2.2, Common Software Measurement International Consortium, January 2003.

[5] *IFPUG*, http://www.ifpug.org (30.11.2003).

[6] *ISO/IEC 14143-1: 1998 – Information Technology – Software Measurement – Functional size measurement – Part 1: Definition of concepts*, ISO 1998.

[7] A. Albrecht: *Measuring Application Development Productivity*, Proceedings IBM Application Development Symposium, Monterey, CA., Oct 14–17, 1979.

[8] *IFPUG Function Point Counting Practices Manual, Release 4.2*, IFPUG Princeton Junction, NJ, January 2004.

[9] D. St-Pierre, M. Maya, A. Abran, J.M. Desharnais: *Adapting Function Point to Real-Time Software*, IFPUG Fall Conference, 1997.

[10] *COSMIC-FFP Measurement Manual*, version 2.1, Common Software Measurement International Consortium, May 2001.

[11] A. Gontarz: *Ile kosztuje program?*, Computerworld, 17.03.03, s. 4–8.

ZMIERZYĆ NIEMIERZALNE
czyli pytania o społeczeństwo informacyjne

MICHAŁ GOLIŃSKI

Szkoła Główna Handlowa
Katedra Informatyki Gospodarczej
mgol@sgh.waw.pl

Man gave sizes to all the animals
In the beginning, in the beginning.
Man gave sizes to all the animals
In the beginning, long time ago.
(lekko zmieniony Bob Dylan)

1. Wstęp

Wzrost znaczenia informacji i technik informacyjnych (IT) we współczesnym świecie zaowocował wzrostem popularności licznych terminów próbujących hasłowo opisać charakter dokonujących się przemian. Społeczeństwo informacyjne, gospodarka oparta na wiedzy i wiele, wiele innych tego typu określeń zrobiło karierę w mediach, polityce i nauce.

W naturalny sposób pojawiła się potrzeba ilościowej oceny zjawisk wywoływanych przez i związanych z zastosowaniami IT. Okazało się to zadaniem trudnym. Komputery i sieci – jako dzieła inżynierskie były łatwo mierzalne – skutki ich zastosowań – jako, często zaskakujące, zjawiska społeczne i gospodarcze nie dawały się już tak łatwo ocenić.

2. Społeczeństwo informacyjne – rzeczywistość czy miraż?

Jednym z podstawowych terminów używanych dla opisu dokonujących się przemian jest pojęcie społeczeństwa informacyjnego (SI). Pomimo, lub może w wyniku, znacznej jego popularności używanie go w sposób odpowiedzialny staje się coraz trudniejsze. Zagadnienia SI należą do grupy zjawisk masowych i do badania ich niezbędne stają się narzędzia statystyczne. Taka analiza nie jest jednak zadaniem łatwym. Dlatego ważne jest stwierdzenie, jakie są możliwości statystycznego monitorowania tej problematyki, jakimi narzędziami opisu dysponujemy oraz jakie jest ich znaczenie, przydatność i wady.

Obecnie istnieje wiele, znacznie się różniących, badań opartych na różnych definicjach i mierzące różne elementy SI za pomocą różnych metod i narzędzi pomiaru (por. [1], [2], [3])[1]. Badania te dają się najogólniej podzielić na trzy kategorie: e-infrastruktura, e-gospodarka i e-społeczeństwo. Tabela 1 ukazuje miejsce Polski w wybranych badaniach. Tabela 2 ukazuje wartości autorskiego wskaźnika postrzeganej gotowości do funkcjonowania w sieci – NRPI.

Brak jest powszechnie akceptowanej definicji SI i brak jest ustaleń dotyczących znaczenia i granic tego pojęcia. Powoduje to sceptycyzm niektórych autorów powątpiewających w sens pojęcia SI: „... centralne znaczenie informacji jest tak stare jak społeczeństwo przemysłowe. Przyjęcie rozróżnienia pomiędzy społeczeństwem przemysłowym a społeczeństwem informacyjnym jest bezsensowne i wprowadzające w błąd." (por. [10]).

Historia pojęcia SI liczy blisko czterdzieści lat. Już w latach 60. wielu badaczy usiłowało analizować i kwantyfikować dokonujące się przemiany (por. [11], [12]). Równie stare są także próby zdefiniowania pojęcia SI. Brak jest jednak jednej, powszechnie akceptowanej definicji. W piśmiennictwie funkcjonuje wiele różnorodnych podejść, praktycznie każdy z autorów posługuje się własną definicją i własnym sposobem pojmowania tego terminu.

Wobec swoistej dewaluacji pojęcia SI wielu badaczy podchodzi do niego dość nieufnie a czasami nawet unika go, starając się opisać dokonujące się przemiany za pomocą innych terminów. Przykładem może być tu Castells, jedna z najważniejszych postaci światowej dyskusji o roli IT we współczesnym świecie. Nie traktuje on swej trylogii Information Age jako opisu SI. Co więcej stara się unikać tego określenia z powodu negatywnych, jego zdaniem, konotacji, istniejących na skutek wcześniejszego używania tego terminu przez innych autorów. Twierdzi raczej, iż jego analiza dotyczy informacyjnego trybu rozwoju współczesnego społeczeństwa kapitalistycznego (por. [13], s. 20–21).

W książce Theories of Information Society, Frank Webster podaje typologię definicji SI, wyróżniając pięć typów: technologiczne, ekonomiczne, zawodowe, przestrzenne i kulturowe. Analizując, powstałe do 1995 roku, definicje dochodzi on do wniosku, iż: „...chociaż wydają się one na pierwszy rzut oka solidne, są, tak naprawdę, mętne i nieprecyzyjne, niezdolne do ustalenia czy 'społeczeństwo informacyjne' powstało czy też powstanie kiedyś w przyszłości." (za: [10]).

3. Problematyka pomiaru SI

Problematyka statystycznego opisu SI, ekonomiki informacji i sektora informacyjnego jest złożona i mająca tak długą historię jak wymienione pojęcia. Już pierwsze analizy tego obszaru zagadnień koncentrowały się na poszukiwaniu metod opisu ilościowego. Trzeba tu wspomnieć o badaniach japońskich czy pracach Machlupa i Porata (zob. [12] czy [14]).

Znaczenie analiz statystycznych wzrosło w wyniku upolitycznienia problematyki SI. Państwa i organizacje międzynarodowe – przynajmniej deklaratywnie –

1. Jednym z najciekawszych, zdaniem autora, opracowań jest badanie przeprowadzone pod kierownictwem G. Kirkmana (por. [4]). Przedstawiona analiza charakteryzuje się umiejętnym połączeniem miar „miękkich" i „twardych" i dużym stopniem szczegółowości.

Tabela1. Porównanie wyników wybranych badań

Badanie: \ Kraj:	cz	de	ee	es	fi	hu	pl	us
	miejsce / pośród badanych krajów							
NRPI – Net Readiness Perception Index (por. [5])	28/49	27/49	20/49	36/49	1/49	24/49	49/49	3/49
ISI – Information Society Index – International Data Corporation (por. [6])	28/55	15/55	–	24/55	8/55	29/55	31/55	4/55
EIU – The Economist Intelligence Unit e-readiness rankings – EIU, IBM (por. [7])	27/60	8/60	–	22/60	10/60	29/60	31/60	1/60
GITR – Global Information Technology Report – World Economic Forum (por. [8])	28/75	17/75	23/75	26/75	3/75	30/75	35/75	1/75
IIDLI – Information Infrastructure Development Level Index – M. Goliński (por. [9])	26/29	7/29	–	23/29	11/29	25/29	28/29	6/29

Tabela2. Ranking i wartości NRPI

	KRAJ	ccTLD	NRPI		KRAJ	ccTLD	NRPI
1	Finlandia	fi	8,10	–	wartość średnia	–	6,28
2	Singapur	sg	7,96	26	Wielka Brytania	uk	6,24
3	USA	us	7,94	27	Niemcy	de	6,10
4	Islandia	is	7,53	28	Czechy	cz	6,03
5	Kanada	ca	7,51	29	Filipiny	ph	6,00
6	Holandia	nl	7,47	30	RPA	za	5,76
7	Dania	dk	7,41	31	Turcja	tr	5,72
8	Australia	au	7,33	32	Tajlandia	th	5,69
9	Szwecja	se	7,32	33	Francja	fr	5,68
10	Austria	at	7,13	34	Grecja	gr	5,64
11	Szwajcaria	ch	7,11	35	Włochy	it	5,64
12	Izrael	il	7,07	36	Hiszpania	es	5,63
13	Hong Kong	hk	7,05	37	Kolumbia	co	5,51
14	Malezja	my	7,05	38	Słowacja	sk	5,48
15	Luksemburg	lu	7,02	39	Indie	in	5,47
16	Nowa Zelandia	nz	6,92	40	Słowenia	si	5,46
17	Irlandia	ie	6,85	41	Japonia	jp	5,39
18	Chile	cl	6,77	42	Portugalia	pt	5,23
19	Tajwan	tw	6,77	43	Rosja	ru	5,21
20	Estonia	ee	6,69	44	Chiny	cn	5,12
21	Korea	kr	6,59	45	Meksyk	mx	5,01
22	Norwegia	no	6,58	46	Wenezuela	ve	4,98
23	Belgia	be	6,42	47	Argentyna	ar	4,88
24	Węgry	hu	6,39	48	Indonezja	id	4,37
25	Brazylia	br	6,37	49	Polska	pl	4,11

traktują rozwój SI jako jeden z najważniejszych czynników wzrostu. Pojawiła się zatem potrzeba narzędzi pozwalających na dokonywanie porównań przestrzennych i historycznych.

Jedną z fundamentalnych trudności opisu ilościowego stanowi brak powszechnie uzgodnionej definicji SI. Brak ten rodzi pytanie jak mierzyć i porównywać coś, czego nie można jednoznacznie zdefiniować.

Pomimo licznych prób prace nad stworzeniem zbioru wskaźników opisujących SI nie przyniosły jak dotąd zadowalających rezultatów. Przeszkodą jest brak uzgodnień jakie charakterystyki SI analizować i jakich wskaźników do tego celu używać. Istnieją trudności z pozyskaniem potrzebnych danych statystycznych a w wypadku porównań międzynarodowych częściowa ich nieporównywalność. Wszystko to powoduje, że praca nad zbiorem odpowiednich narzędzi znajduje się, ciągle jeszcze, na etapie początkowym.

Statystyka jest zorientowana na dobra fizyczne – ich produkcję i dystrybucję – charakterystyczne dla społeczeństwa przemysłowego. Systemy klasyfikacji koncentrują się na produktach materialnych a nie na posiadającej charakter niematerialny informacji. Powoduje to nieprzystosowanie statystyki do zmieniającej się rzeczywistości – nie możemy zliczać „kawałków" informacji tak jak liczyliśmy wyprodukowane statki, samochody, zboże czy trzodę chlewną.

Użyteczność informacji jest zależna od kontekstu, od tego kto i jak ją wykorzystuje. Informacja może być używana przez wielu odbiorców w tym samym czasie. Także proces utraty wartości jest w wypadku informacji inny niż w wypadku dóbr fizycznych. Wszystko to powoduje, że statystyka staje wobec nowych wyzwań. Potrzebne są nowe koncepcje, definicje, mierniki, i metody pozyskiwania danych źródłowych, które pozwoliłyby na monitorowanie SI i gospodarki informacyjnej.

Szybki rozwój IT i nowych zastosowań powoduje potrzebę monitorowania coraz to innych aspektów problematyki SI. Pojawia się potrzeba narzędzi pozwalających badać i mierzyć nowo powstające usługi i produkty informacyjne i ich otoczenie. Zastosowania IT są źródłem wielu nowych zjawisk i fenomenów społecznych, politycznych i gospodarczych także wymagających nowych metod pomiaru.

Skala i głębokość przemian wywoływanych przez IT powoduje wzrost znaczenia analiz socjologicznych. Przedmiotem zainteresowania statystyki SI powinna się stać także sfera polityki. IT już zmieniają także tę sferę procesów społecznych.

Pojawia się problem opłacalności badań statystycznych i kosztów pozyskania danych źródłowych. Zasygnalizowane powyżej postulaty mają oczywiście charakter życzeniowy. Spełnienie części z nich może w praktyce okazać się niewykonalne lub bardzo kosztowne. Koszt pozyskania informacji nie powinien być większy niż jej wartość, zasada ta musi obowiązywać także w tym wypadku.

Osobnym problemem jest aktywność badawcza firm prywatnych. Brak oficjalnych mierników i statystyk w połączeniu z popytem na informacje dotyczące SI spowodował reakcję rynku. Wiele firm konsultingowych rozpoczęło monitorowanie tej tematyki i publikację licznych raportów, wypełniając lukę stworzoną przez oficjalne służby statystyczne. Istotnym problemem jest różna jakość tych opracowań, nie zawsze znane narzędzia, których użyto, różne metody badawcze i wprost przysłowiowa

już rozbieżność wyników. Jest jednak faktem, iż niektóre z tych opracowań zdobyły sobie silną pozycję.

Innym problemem jest częsty brak wymiernych wyników zastosowań IT. Znaczna część korzyści biznesowych nie da się opisać w kategoriach pieniężnych czy ilościowych. W grę wchodzą tu profity o charakterze jakościowym takie jak; możliwość zaoferowania odbiorcy większego wyboru, polepszenie jakości obsługi, większa łatwość dostępu do informacji, oszczędność czasu czy wygoda klienta. Istniejące miary statystyczne, bazujące najczęściej na ilościowych metodach oceny wyniku działalności gospodarczej, zawodzą w takich przypadkach.

Wypracowanie statystycznych narzędzi opisu SI jest procesem długofalowym, wymagającym rozwiązania wielu wspomnianych problemów. Dostosowanie systemu statystyki do nowych zjawisk zajmuje dużo czasu potrzebnego do zaprojektowania i przeprowadzenia badań. Powoduje to nieuchronną dezaktualizację otrzymywanych wyników. Pewne opóźnienie wyników jest oczywiste i nieuchronne, można powiedzieć że charakterystyczne dla statystyki. Jednak w obszarze charakteryzującym się tak znaczną dynamiką opóźnienie to staje się szczególnie dokuczliwe. Ranking mocy produkcyjnych w przemyśle stoczniowym czy hutnictwie prezentujący dane sprzed dwóch lat zachowuje swą aktualność. W sferze IT te same dwa lata mogą oznaczać zupełnie nową sytuację technologiczną i rynkową.

Jest swoistą ironią, iż pomimo coraz większej ilości informacji, coraz większej jej dostępności i rosnącego znaczenia, o społeczeństwie, które powstaje w wyniku oddziaływania właśnie informacji, wiemy stosunkowo niewiele. Trudno jest nam opisać zjawiska zachodzące w otaczającej nas rzeczywistości a jeszcze trudniej przewidzieć co nas czeka. W gruncie rzeczy o społeczeństwie informacyjnym wiemy bardzo niewiele. Tak naprawdę nie wiemy nawet czego nie wiemy. Jak pisze Lem: „Wiek XXI będzie inny niż jego liczne teraz przewidywania, wysadzane klejnotami dziwacznych pomysłów" (por. [15], s. 218).

Odpowiedzialne i przemyślane stosowanie narzędzi statystycznych może przyczynić się do zmniejszenia luki informacyjnej dotyczącej społeczeństwa informacyjnego.

Literatura

[1] N. Choucri et al., *Global E-Readiness – for What?*, MIT, Cambridge, 2003, http://ebusiness.mit.edu/research/papers/177_Choucri_GLOBAL_eREADINESS.pdf

[2] N.N.: *Comparison of E-Readiness Assessment Models*, 2001, http://www.bridges.org/ereadiness/report.html

[3] N.N.: *E-readiness Assessment: Who is Doing What and Where*, 2002, http://www.bridges.org/ereadiness/where.html

[4] G. Kirkman: *The Global Information Technology Report 2001–2002: Readiness for the Networked World*, http://cyber.law.harvard.edu/itg/libpubs/gitr.html

[5] M. Goliński: *Ocena zdolności Polski do funkcjonowania w sieci*, w: J. Goliński, D. Jelonek, A. Nowicki: *Informatyka ekonomiczna. Przegląd naukowo-dydaktyczny*, Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu, Wrocław, 2004.

[6] http://www.worldpaper.com/indexes/ISI/012002/TheWorldPaperISI2002.pdf

[7] http://www.ebusinessforum.com/index.asp?layout=rich_story&doc_id=5768

[8] http://www.weforum.org/pdf/ger/gitrr/gitrr2002_execsumm.pdf

[9] M. Goliński: *Infrastruktura informacyjna Polski*, w: *Wybrane zagadnienia współczesnej informatyki gospodarczej*, SGH, Warszawa, 2004.

[10] P. Mullan: *Information society: frequently un-asked questions*, http://www.spiked-online.com/Printable/0000000053AA.htm

[11] J. Lubacz (red.): *W drodze do Społeczeństwa Informacyjnego*, OZSI-IPWC, Warszawa, 1999.
[12] M. Goliński: *Poziom rozwoju infrastruktury informacyjnej społeczeństwa. Próba pomiaru*, PLJ, Warszawa, 1997.
[13] M. Castells: *The Rise of the Network Society. The Information Age*, Blackwell, Oxford, 1996.
[14] M. Hensel: *Die Informationsgesellschaft. Neuere Ansätze zur Analyse eines Schlagwortes*, Verlag Reinhard Fischer, München, 1990.
[15] S. Lem: *Bomba megabitowa*, Wydawnictwo Literackie, Kraków, 1999.

Sponsorzy:

Polskie Towarzystwo Informatyczne

ul. Św. Jana 10, 40-012 Katowice
tel. (32) 253 96 78, 253 87 25, fax: (32) 253 96 78
NIP: 522-000-20-38
Konto: PKO BP S.A. I Oddział Katowice 64 1020 2313 0000 3402 0024 0267

Patronat medialny: