IX Konferencja PLOUG
Kościelisko
Październik 2003

Jak sprawić, aby dane były czyste?

Tomasz Wyrozumski

Biuro Matematyki Stosowanej S.C.

Omawiamy pokrótce przyczyny, dla których jakość danych gromadzonych w systemach informatycznych jest na ogół gorsza niż oryginalnie zakładana przez ich projektantów. Staramy się przy tym wskazać te cechy systemów, które mają najbardziej negatywny wpływ na jakość wprowadzanej do nich informacji. Pokazujemy też, w jaki sposób zanieczyszczenie informacji wpływa zwrotnie na złe funkcjonowanie systemów, lecz z drugiej strony argumentujemy, że nadmierne wymagania w stosunku do danych obniżają efektywną użyteczność oprogramowania, które je gromadzi. Przedstawiamy wreszcie typowe sytuacje, w których konieczne jest przeprowadzenie operacji czyszczenia danych i próbujemy zobrazować, w jaki sposób użycie metod data mining może okazać się do tego przydatne.

Informacja o autorze:
Z wykształcenia fizyk, absolwent Uniwersytetu Jagiellońskiego. Doktorat w dziedzinie fizyki teoretycznej na Uniwersytecie Wiedeńskim. Od 1992 roku zajmuje się profesjonalnie informatyką, kierując Biurem Matematyki Stosowanej (Kraków). Uczestniczy przy tym aktywnie w komercyjnych projektach informatycznych, związanych między innymi z analizą i eksploracją danych („data mining”).

1. Początki

Trudno powiedzieć, kiedy po raz pierwszy zaczęto zbierać dane, ale przypuszczalnie było to tysiące lat wcześniej nim na świat przyszły komputery. Można przypuszczać, że właśnie wtedy pojawił się też problem czystości danych i nawet jeśli miał on inny wymiar niż dziś, to bez wątpienia w pewnych sytuacjach potrafił nieźle dawać się we znaki. Pomyślmy o gigantycznych przedsięwzięciach informatycznych Starożytności, jakimi były spisy powszechne, np. ten przeprowadzony przez cesarza Augusta, o którym między innymi wspomina Biblia *(Łk. II, 1)*. Wyobraźmy sobie dane napływające do Rzymu z całego ówczesnego świata, zbierane przez różne osoby, z różną starannością i zapewne w różnym formacie, bo nawet jeśli cesarski dekret w miarę dokładnie precyzował, co i jak należy zapisywać, to w końcu wykonawcami byli tylko ludzie, których natura nie zmieniła się specjalnie przez ostatnie dwa tysiąclecia. W efekcie więc przetwarzanie danych musiało wymagać dokonywania pewnej obróbki wstępnej, powszechnie nazywanej czyszczeniem.

Przejdźmy jednak do czasów współczesnych – epoki komputerów, które z jednej strony niesłychanie usprawniają działania człowieka, z drugiej jednak bezlitośnie eksponują jego błędy i zaniedbania, swoją mocą obliczeniową potęgując je do niewyobrażalnych wprost rozmiarów. Spróbujemy zastanowić się nad problemem czystości danych, podchodząc do tematu pragmatycznie. Z jednej strony chcielibyśmy zachować dość ogólny charakter rozważań, z drugiej jednak – nie wdawać się w dywagacje bardzo teoretyczne i przesadnie formalne. Naszym celem jest w końcu podzielenie się z projektantami systemów informatycznych pewnymi doświadczeniami, które być może w jakiś sposób pomogą im w rozwiązywaniu konkretnych, technicznych problemów. Będziemy starali się operować przykładami, a także formułować pewne fenomenologiczne tezy o charakterze jakościowym. Nazwiemy je obserwacjami i twierdzeniami, lecz nie przeprowadzimy żadnego formalnego dowodu. Byłoby to zresztą trudne, zważywszy nieszczególną precyzję naszej głównej definicji. Mamy nadzieję, że czytelnik wybaczy nam w tym miejscu nawiązanie do sposobu prezentacji typowego dla literatury matematycznej i potraktuje je raczej jako swoistą figurę stylistyczną, służącą zwiększeniu przejrzystości tekstu.

2. Dane brudne i czyste

Co to są dane zanieczyszczone? Dla naszych celów proponujemy przyjąć definicję mało precyzyjną, lecz prostą i intuicyjnie zrozumiałą.

DEFINICJA. Dane zgromadzone w systemie informatycznym nazywamy czystymi, jeśli są prawdziwe i wyglądają tak, jak wyobrażał to sobie projektant tego systemu.

Bazujemy zatem na zgodności danych zarówno z rzeczywistością, jak i z oryginalnym zamysłem projektanta. Gdy zawierają one fałsz lub wspomniany zamysł nie został spełniony, mówimy o zanieczyszczeniu danych. Aby uprościć nieco rozważania ograniczymy się przy tym wyłącznie do pewnej klasy systemów – tych, które funkcjonują w oparciu o bazy relacyjne. Jak wiadomo, w praktyce jest to przeważająca większość rozwiązań.

Poprawnie zbudowana baza relacyjna nie powinna dopuścić do nadmiernego zanieczyszczenia danych, a to dlatego, że już sama jej struktura narzuca pewne ograniczenia. Teoretycznie możemy sobie wyobrazić zarówno bazę idealnie czystą, jak i całkowicie zanieczyszczoną. Dla celów dalszych rozważań wprowadźmy odpowiednią miarę czystości bazy relacyjnej.

$$Czystość\ danych = \frac{Ilość\ pól\ czystych}{Ilość\ wszystkich\ pól}$$

Zliczając pola będziemy dla uproszczenia brać pod uwagę wszystkie tabele i wszystkie ich kolumny, również te wypełniane przez różnego rodzaju automatyczne mechanizmy (np. sekwencje).

Oznacza to, że osiągnięcie czystości zerowej – a więc bazy całkowicie brudnej – byłoby możliwe tylko przy zupełnie fatalnym zaprojektowaniu systemu. Z kolei baza całkowicie czysta (wartość 1) musiałaby być wynikiem sterylnej pracy operatorów, bądź urządzeń zewnętrznych i – jak będziemy dalej argumentować – w praktyce stanowi nieosiągalny ideał.

3. Dlaczego dane ulegają zanieczyszczeniu?

Ogólnie rzecz biorąc można wyróżnić dwie przyczyny zanieczyszczeń danych: pierwsza to błędne działania maszyny (na ogół oprogramowania), wynikłe z drobnych niedopatrzeń twórców. Jeżeli np. omyłkowo linia do wprowadzania danych osobowych, opisana etykietą „Imię", została podłączona do pola „Nazwisko" w bazie danych, to poprawne działania operatora będą w oczywisty sposób generować złe dane. To źródło błędów jest w pewnym sensie trywialne. Remedium stanowi rzecz jasna wprowadzenie odpowiedniej poprawki do programu, a o błędach w kodzie, ich usuwaniu i testowaniu oprogramowania napisano już dostatecznie wiele (np. [1]), aby poświęcać im tu więcej uwagi. Ta kategoria przyczyn zanieczyszczeń danych po prostu nie będzie nas w ogóle interesować.

Paradoksalnie, dużo ciekawszym zjawiskiem jest psucie się danych z winy operatora lub zewnętrznych urządzeń zasilających, czytników, rejestratorów, innych programów itp. Technika jest jak wiadomo zawodna, lecz generalnie dużo łatwiej jest radzić sobie z jej kaprysami, niż z błędami i szczególnymi przyzwyczajeniami człowieka. Dlatego też znów ograniczymy pole naszych analiz i skoncentrujemy się na postępowaniu ludzi, najsłabszych elementów w systemach informatycznych.

Rozważmy pewien przykład: oto zmęczony, czy roztargniony człowiek myli się wprowadzając numer telefonu. Dane stają się przez to nieprawdziwe. Jeżeli np. pomyłka dotyczyła jednej cyfry, wskutek czego do bazy wprowadzono numer charakteryzujący innego abonenta, to trudno tu mówić o jakiejś niezgodności z zamysłem projektanta. Jeśli jednak wpisano o jedną cyfrę za mało lub za dużo, to uzyskany w ten sposób ciąg znaków w ogóle nie będzie numerem telefonu, a więc dodatkowo mamy też ową niezgodność. Można argumentować, że sam system powinien odpowiednio zareagować. W takim przypadku skoro wiadomo, że numery określonych stref mają stałą strukturę, to można stworzyć w bazie odpowiednie tabele, które będą ją opisywać, służąc następnie dokładniejszej weryfikacji wprowadzanych danych. Wydaje się proste. W szczególności numery telefonu nie mogą zawierać liter ani znaków specjalnych, poza rozdzielającymi: spacją lub myślnikiem, a zatem jeśli operator pomyli cyfrę „3" z literą „e" (ten sam rejon klawiatury), system powinien się przed taką pomyłką obronić. To jest nawet jeszcze prostsze. Oryginalny zamysł projektanta, polegający na tym, że kolumna „numer telefonu" zawiera w istocie numery telefonów, nie zaś śmieci, powinien więc przybrać fizyczną postać różnych zabezpieczeń: masek na formatkach, tabel opisujących związek między strukturą numeru a regionem kraju, funkcji sprawdzających integralność wprowadzanych danych, wymagań unikalności itp. Dochodzimy zatem do pewnego podsumowującego stwierdzenia.

OBSERWACJA. Przyczynami dla których dane ulegają zanieczyszczeniu są zarówno błędy operatora, bądź urządzeń zewnętrznych, jak i niedoskonałości projektu systemu informatycznego.

Ile jednak musi natrudzić się twórca systemu, aby jego użytkownik mógł spokojnie wprowadzić kilka cyfr! Nie trzeba chyba nikogo przekonywać, jak wzrosną koszty i czas realizacji takiego rozwiązania w porównaniu z pierwotną koncepcją jednego pola tekstowego o długości np. 25 znaków (tak na wszelki wypadek). Puryści zaoponują jednak w tym miejscu: porządny system musi kosztować, a klient chętnie zapłaci za czystość danych (na pewno?). Narzuty na czas przetwarzania zrekompensuje się mocniejszymi serwerami i wszystko będzie dobrze. Co się jednak stanie, jeśli nasz operator zechce przy którymś z klientów wprowadzić dodatkowo numer wewnętrzny? Oczywiście i to należy przewidzieć, zawczasu przygotowując odpowiednie pole w bazie danych i na formatce. A jeśli numerów wewnętrznych miałoby być z jakichś powodów więcej? Powinni-

śmy mieć listę pól na numery wewnętrzne. Nawet nie warto już dodawać, że i samych numerów zewnętrznych może być więcej, co oznacza konieczność wprowadzenia odpowiedniej struktury. Oczywiście nie można zapomnieć i o prefiksach kierunkowych, najlepiej wypełnianych automatycznie po podaniu nazwy kraju i miejscowości, gdzie zainstalowany jest aparat. I tak dalej, i tak dalej... Banał jakim jest pole na numer telefonu urasta do rangi nie lada problemu, przybierając nieco bardziej materialną postać roboczogodzin i już całkiem materialną – złotych polskich, euro, dolarów, franków szwajcarskich lub innych podobnych jednostek. Niemniej jednak rozwiązanie wydaje się być widoczne na horyzoncie. Wydaje się, bo naszym zdaniem jest to tylko fatamorgana…

Operator zawsze będzie omylny. Jakie są jednak szanse, że projektant przewidzi wszystko? Każdy kto tworzył najmniejszy choćby system informatyczny, odpowie od razu: żadne. Jest natomiast też druga strona medalu. Komplikując oprogramowanie utrudniamy de facto życie naszemu operatorowi. Wystarczy wyobrazić sobie, że przyjdzie mu wprowadzić do bazy telefon osoby, która mieszka w strefie numeracyjnej nie zdefiniowanej jeszcze w systemie. Wtedy trzeba będzie na chwilę przerwać pracę i zająć się uzupełnieniem tych bardziej podstawowych danych, co – jeśli ma być zrobione dobrze – może wymagać odpowiednich kompetencji, uprawnień, nierzadko interwencji administratora. Nasuwa się nam więc drugi wniosek.

OBSERWACJA. System zbyt rygorystycznie pilnujący czystości danych jest mało przyjazny dla użytkownika.

Jeśli z kolei w trakcie korzystania z systemu okaże się, że jest on zbyt restrykcyjnie skonstruowany, to wcale nie jest oczywiste, że jego użytkownik natychmiast zleci producentowi odpowiednią przebudowę. Może się zdarzyć, że sam spróbuje znaleźć jakieś lepsze lub gorsze rozwiązanie – na przykład bardzo nietypowe numery telefonu zacznie wpisywać do pola „Uwagi", dopuszczającego dowolny tekst o długości do 255 znaków. Gorzej, jeśli wejdzie mu to w nawyk, o co zresztą nietrudno, gdyż takie pole, przyjmujące wszystko, co kto tylko wymyśli, będzie się zachowywać bardzo „przyjaźnie" na tle pozostałych, rygorystycznie filtrujących informację. W efekcie u człowieka może zadziałać elementarny psychologiczny mechanizm – polubi pole „Uwagi"! I tu znów ogólniejsza konstatacja.

OBSERWACJA. Użytkownik najchętniej wprowadza dane tam, gdzie nikt nie pilnuje ich czystości.

Ewentualnością alternatywną, np. w przypadku braku „Uwag", byłoby utrzymywanie czystości danych za cenę rezygnacji z pewnych działań w sferze realnej: „System nie przyjmuje pańskiego numeru telefonu, więc nie będziemy do pana dzwonić, proszę dzwonić do nas." Tego rodzaju praktyki dotyczą bardziej administracji publicznej niż prywatnej przedsiębiorczości, ale bynajmniej nie należą do wyjątków. Rozwiązaniem częściej spotykanym w firmach jest używanie różnego rodzaju baz uzupełniających (np. tzw. „zeszytów"), w efekcie czego dane „wyciekają" poza oryginalnie przeznaczone dla nich pola lub nawet poza cały system, gdzie już oczywiście znacznie łatwiej o zanieczyszczenia.

Niekiedy użytkownicy modyfikują we własnym zakresie systemy informatyczne – nie tyle w sensie przepisywania kodu, co raczej zmieniania przeznaczenia niektórych rozwiązań. Autorowi znany jest przypadek, gdy w pewnej bazie danych firma gromadziła informacje zarówno o swoich klientach, jak i dostawcach. Ponieważ jednak system nie pozwalał na ich skuteczne rozróżnianie, wymyślono, że dostawców charakteryzować się będzie pewnym szczególnym kodem pocztowym: 00-000. Nie wystawiano im faktur i w zasadzie nie wysyłano do nich listów, więc nikomu to specjalnie nie przeszkadzało, lecz dane stały się karykaturą tego, co oryginalnie założył projektant systemu!

Po tych wstępnych uwagach chcielibyśmy postawić pewną ogólniejszą, nieco prowokacyjną tezę. Chodzi mianowicie o to, że psucie się danych jest w naszej opinii nieuniknione i stanowi jesz-

cze jeden przejaw wszechobecnej drugiej zasady termodynamiki. Tym, którzy zapomnieli ją ze szkoły dedykujemy nasze ulubione, znakomicie przemawiające do wyobraźni sformułowanie:

W zamkniętym pokoju, w którym przebywają ludzie i nikt nie sprząta, bałagan nie maleje.

Podobnie jest z informacją. Ilościową miarą bałaganu, tak w fizyce, jak i w informatyce jest wielkość zwana entropią, posiadająca zresztą ścisłą definicję w każdej z tych dziedzin (zobacz np. [2]) Pojęcie entropii informacyjnej mogłoby nota bene posłużyć sformalizowaniu naszych rozważań, gdyby oczywiście zaszła taka potrzeba. W chwili obecnej chcielibyśmy natomiast tylko podkreślić, że narastanie chaosu w użytkowanych systemach informatycznych jest czymś naturalnym, podobnie jak „samoczynne" psucie się wszelkiego, ciężką pracą zaprowadzonego ładu. Z termodynamiki wiadomo, że kostka lodu wrzucona do szklanki z gorącą wodą roztopi się, a entropia układu wzrośnie. Aby przeciwdziałać takiemu naturalnemu procesowi należy użyć specjalnego urządzenia zwanego lodówką, które w środku chłodzi, a na zewnątrz grzeje. Wymaga ono jednak dostarczania energii elektrycznej, czyli wykonywania pewnej pracy. Nic za darmo. Podobnie jest z danymi: psują się same, zaś aby je naprawić, trzeba nie lada wysiłku. Kończymy więc ten fragment rozważań nieco pesymistycznym podsumowaniem.

TWIERDZENIE. W bazach danych, których nikt nie czyści, entropia informacyjna nie maleje.

Innymi słowy, w każdym nietrywialnym systemie gromadzącym informację wcześniej, czy później pojawią się zanieczyszczone dane.

4. Konsekwencje bałaganu

Skoro doszliśmy już do wniosku, że w danych zawsze pojawiać się będą śmieci i w zasadzie nie ma na to rady, to może warto przez moment zastanowić się, czy w ogóle jest o co walczyć. Na pozór pytanie wydaje się retoryczne, bo w końcu nie po to wprowadza się systemy informatyczne, aby sankcjonować informacyjny bałagan. Z drugiej strony, jeśli taki „naturalnie" wkradający się w dane szum nie byłby zbyt duży, to przy założeniu stabilności układu nie powinien prowadzić do zbyt poważnych problemów i można by z nim jakoś żyć. Tak się zresztą dzieje w rzeczywistości. Użytkownicy godzą się z tym, że w ich danych jest trochę śmieci i chcąc nie chcąc przechodzą nad tym do porządku dziennego. Niekiedy łudzą się, że dane są czyste, a niekiedy świadomie udają, że tak jest (choć nie do końca wiadomo po co to robią). Dopiero okazyjne czyszczenia danych, dokonywane najczęściej przy okazji migracji z jednego systemu do drugiego, uświadamiają im, że skala problemu jest dużo większa niż sądzili.

Odrębny przypadek stanowią użytkownicy celowo dopuszczający nieporządek w danych. Jak się okazuje, nie zawsze dobrze jest dużo wiedzieć. Bywa na przykład, że przepisy nie obligują firmy do gromadzenia pewnych informacji, jednakże jeśli już je gromadzi, mogą mieć w nie wgląd określone organa kontrolne. W sytuacji, gdy prawo jest złe i nieprzejrzyste, a urzędnicza samowola stanowi normę, przedsiębiorstwo woli, aby nie dało się w żaden sposób uzyskać niektórych informacji o jego funkcjonowaniu, nawet za cenę niższej jakości procedur wewnętrznych i gorszego zarządzania. To zrozumiałe, choć bardzo smutne zjawisko, i to nie tylko z punktu widzenia informatyka. Nawiasem mówiąc, łatwiej przekonać firmę do zaniechania takich praktyk argumentując to wymogami norm ISO, które planuje ona w przyszłości wdrożyć, aniżeli perspektywą rzeczywistej poprawy organizacji pracy. To zresztą jeszcze smutniejsze.

Wróćmy jednak do pytania o konsekwencje bałaganu w danych, bałaganu narastającego naturalnie w trakcie użytkowania systemu. Jak się okazuje, nie utrudnia on specjalnie życia operatorom w ich codziennej działalności. Dzieje się tak dlatego, że nawet jeśli zajmują się oni odczytywaniem danych, nie zaś ich zapisywaniem, to odczyt taki dotyczy zazwyczaj poszczególnych rekordów, a na tym poziomie dane zanieczyszczone są zgodnie z pewną wprowadzoną przez człowieka logiką, dla niego całkowicie czytelną, choć trudną do przetworzenia przez maszynę. Problem powstaje wtedy, gdy próbujemy przeprowadzić automatycznie różne operacje na większych zesta-

wach danych. Może tu chodzić zarówno o wyszukiwanie wzorców, jak i filtrowanie informacji, tworzenie raportów, ekstraktów itp., krótko mówiąc, jeśli zamiast „przeglądać" bazę, tak jak się przegląda zeszyt, wydajemy – mniej lub bardziej bezpośrednio – polecenie typu „SELECT... WHERE...". Wyobraźmy sobie na przykład, że z jakichś powodów chcemy znaleźć wszystkie numery telefonów kończące się cyfrą 3. Co otrzymamy, jeżeli w bazie znajdą się wpisy takie jak np.

„234-67-20 w. 103", „769-45-83, -84", „692-15-03 – sekretariat"?

Widać więc mechanizm zjawiska. Człowiek narzuca sobie pewne rygory, ale ich nie przestrzega, bo tak mu wygodniej, albo też dlatego, że rygory okazały się nieżyciowe. Jak długo wprowadzana informacja przetwarzana jest tak naprawdę tylko przez niego, bądź przez innego człowieka, wszystko jest w porządku, bo ludzie myślą bardzo elastycznie i potrafią dostrzec pewien kontekst przetwarzania. Problem powstaje wtedy, gdy za dane zabiera się maszyna – ta robi tylko i wyłącznie to, co jej kazano. Zasadniczo można by powiedzieć, że znów winien jest człowiek, który chcąc np. wyszukać pewien wzorzec w danych, zadaje zbyt proste kryteria (cyfra „3" na końcu łańcucha znaków). Zwróćmy jednak uwagę, że kryteria te dopasowuje właśnie do projektu systemu informatycznego, nie zaś do beztroskich zachowań jego użytkowników. Być może, gdyby miał świadomość owych zachowań, działałby inaczej, lecz przecież dziwne praktyki operatorów na ogół nie znajdują odbicia w żadnych opisanych procedurach, raczej odbywają się ad hoc, na zasadzie „przecież i tak wiadomo o co chodzi". Domyślny czytelnik zgadł już zapewne, jaką tezę chcemy postawić w tym miejscu.

OBSERWACJA. Im więcej funkcji raportujących w systemie, tym większe znaczenie ma czystość zgromadzonych w nim danych.

Z tego właśnie względu omawiany przez nas problem staje się niezwykle istotny we wszystkich projektach związanych z hurtowniami danych, eksploracją (data mining), systemami MIS itp. Tam bałagan ma bezpośredni wpływ na jakość informacji na wyjściu, a z uwagi na dużą automatyzację przetwarzania należy niekiedy liczyć się nawet ze skrajną niestabilnością wyników, kiedy to niewielki błąd w danych może prowadzić do całkiem absurdalnych rezultatów.

W rzeczywistości czyszczenie informacji stanowi niemal zawsze element procesu zasilania hurtowni, czy też eksploracji danych, element niezbyt efektowny, lecz bardzo ważny. Autor niedawno miał okazję brać udział w pewnym projekcie data mining, który zakończył się uzyskaniem kilku ciekawych wyników, jednakże klient stwierdził, iż osiągnięto jeszcze jeden, dodatkowy rezultat: uświadomiono mu, jak bardzo zanieczyszczone są gromadzone przezeń dane!

5. Złoty środek

Co można doradzić projektantom systemów informatycznych, z góry skazanych na porażkę w nierównej walce o czystość danych? Jeśli za wszelką cenę będą starali się wygrać, ryzykują ogromny koszt przedsięwzięcia, w wyniku którego powstanie rozwiązanie nadzwyczaj skomplikowane, trudne do wdrożenia i administrowania, a także ogólnie nieprzyjazne użytkownikom. Jeśli z kolei potraktują problem czystości danych zbyt luźno, to mają wszelkie szanse zbudować nie tyle system informatyczny, co raczej bardzo wyspecjalizowany kosz na śmieci.

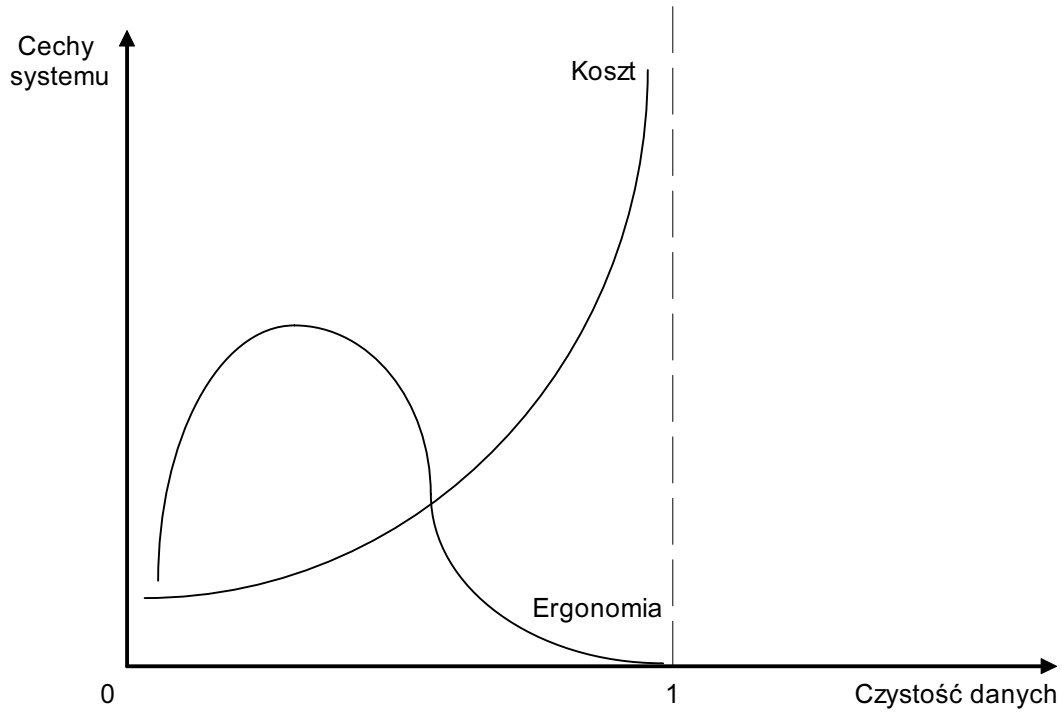
Gdzie zatem leży złoty środek? Szukając go warto pamiętać, że w istocie powinno chodzić nam o to, by dane były czyste, lecz niekoniecznie sterylne. Projektując system należy oczywiście mieć na uwadze kwestię czystości danych, lecz dobrze jest postrzegać ją nie jako cel sam w sobie, lecz raczej środek prowadzący do innego, którym jest możliwość automatycznego ekstrahowania z bazy niezbędnych informacji. I ten właśnie cel powinno się mieć zawsze przed oczami.

Warto pamiętać, że normalizacja baz (zobacz np. [3]), wpisana w istotę modelu relacyjnego, bardzo korzystnie wpływa na czystość danych, gdyż operatorzy w wielu przypadkach nie mogą wprowadzać wartości dowolnych, lecz muszą wybierać je z określonych słowników. Jeśli przykła-

dowo w pewnej bazie adresowej słownikować będziemy nazwy miejscowości, to najprawdopodobniej nie dojdzie do sytuacji, w której jedno miasto nazywać się będzie na wiele sposobów (np. „Kraków" i „Krakow") wskutek literówek operatora. Oczywiście, sprawne działanie systemu będzie wymagało zasilenia słownika wszystkich miast, z którymi będzie mógł się spotkać operator podczas swojej pracy, a to już spore przedsięwzięcie. Wyobraźmy sobie, że zapomnimy o jakimś przysiółku i pechowo pojawi się osoba, która właśnie tam mieszka. Rozwiązaniem byłoby albo umieszczenie na formatce przycisku z napisem „Dodaj", albo też polecenie operatorowi, by w takiej sytuacji przerwał pracę i zwrócił się do administratora (lub kogoś uprawnionego) o uzupełnienie słownika. Pierwszy pomysł wydaje się lepszy, jednak może prowadzić do bardzo złych skutków. Operator przeoczywszy pewną nazwę na liście może z rozpędu wprowadzić ją do słownika i to w dodatku w nieco zmienionym brzmieniu (np. „Nowy Targ", „N. Targ", „Nw. Targ"). Drugie rozwiązanie jest poprawniejsze, ale wiąże się z dodatkową uciążliwością, o czym zresztą wspominano już wcześniej.

Pójdźmy jednak dalej i zastanówmy się, czy warto słownikować ulice. Z punktu widzenia czystości danych, zapewne tak – autor widział już bazę, w której ulica Staszica wystąpiła w dziewięciu odmianach włączając w to również literówki („Staszica", „S. Staszica", „St. Staszica", „Stanisława Staszica", „Ks. Staszica", Staszca", itd.). Słownikując ulice natrafimy jednak na te same problemy, co przy okazji miejscowości i jeszcze bardziej skomplikujemy tak system, jak i pracę operatorów. Należy zadać sobie zatem pytanie, jaki głębszy cel będzie miało utrzymanie czystości danych w tym zakresie. Jeśli zechcemy kiedyś uzyskać spis wszystkich osób mieszkających w określonym mieście przy określonej ulicy, to na pewno słownikowanie ulic jest głęboko uzasadnione, jeżeli jednak nie planujemy zadawania takich pytań, to może lepiej byłoby z niego zrezygnować. I tu jak na dłoni widać dylemat projektanta, który z jednej strony powinien przewidzieć jak najwięcej przyszłych potrzeb użytkownika, a z drugiej musi pamiętać o takich przyziemnych sprawach, jak czas realizacji projektu, jego budżet, wydajność systemu i wreszcie – ergonomia pracy operatorów.

Na załączonym rysunku przedstawiliśmy zależność pomiędzy dwiema istotnymi cechami systemu informatycznego, tzn. jego kosztem oraz ergonomią obsługi a oczekiwaną czystością danych. Przez koszt rozumiemy przy tym ilość godzin pracy potrzebnej do wytworzenia i uruchomienia oprogramowania, zaś przez ergonomię – odwrotność czasu, jaki operator musi średnio przeznaczyć na wprowadzenie jednego rekordu danych. Oczywiście wykres ma charakter symboliczny – nie jest produktem żadnej teorii, ani też efektem pomiarów. Jego zadaniem jest wyłącznie graficzne zilustrowanie niektórych przemyśleń przedstawionych w niniejszym artykule. Jeśli zgodzić się z nimi, koszt systemu rośnie monotonicznie wraz ze wzrostem wymagań odnośnie jakości danych i asymptotycznie zmierza do nieskończoności, gdy dane osiągają ideał. Inaczej zachowuje się naszym zdaniem ergonomia obsługi. Rozwiązania, które w ogóle nie pilnują czystości informacji nie są na ogół specjalnie wygodne, dlatego początkowo ergonomia rośnie, ale po osiągnięciu dość rozmytego maksimum zaczyna spadać, aby w granicy doskonałych danych osiągnąć zero. Istotnie, system idealnie pilnujący czystości danych staje się nieznośny w obsłudze, gdyż aby wprowadzić doń najprostszą informację, trzeba najpierw zasilić różne struktury pomocnicze, których czystości strzegą inne struktury pomocnicze itd. W efekcie czas potrzebny do wykonania wszystkich tych operacji rośnie do nieskończoności.



Zależność cech systemu od oczekiwanej czystości danych

Niestety nie ma uniwersalnej odpowiedzi na pytanie, gdzie leży ów złoty środek wyznaczający idealny kompromis pomiędzy czystością danych a kosztem i ergonomią pracy. Do każdego problemu należy podejść indywidualnie. Projektantom możemy radzić tylko, aby tworząc koncepcje systemów informatycznych mieli na uwadze wszystkie aspekty zagadnienia i ani nie budowali śmietników ku wygodzie użytkownika, ani też w trosce o doskonałość danych nie konstruowali monstrów, z którymi nie da się żyć.

6. Jak czyścić dane?

A przede wszystkim: kiedy czyścić dane? Ktoś dowcipny powiedziałby: kiedy są brudne. Można by też pomyśleć o regularnych porządkach: raz na tydzień, na miesiąc, przed Świętami (odpowiednikiem byłby tu jakiś nadzwyczajny audyt). Problem polega jednak na tym, że czyszczenie danych jest niezwykle trudną i kosztowną procedurą. Praktyka pokazuje, że przeprowadza się ją tylko podczas przenoszenia danych pomiędzy systemami, co może mieć miejsce albo incydentalnie, zazwyczaj przy okazji wymiany oprogramowania, albo regularnie, gdy jeden system (np. hurtownia danych) jest zasilany informacjami z innego, wspierającego działalność operacyjną przedsiębiorstwa.

W pierwszym przypadku, gdy mamy do czynienia z migracją, czyszczenie przeprowadza się w zasadzie „ręcznie". Nie chodzi przy tym o przeglądanie rekordu po rekordzie, ale raczej o to, że trudno wymyślić jakieś uniwersalne, automatyczne procedury, nie wiedząc, czego się można po danych spodziewać. Prowadzący odpowiednie prace konsultanci jednocześnie więc czyszczą dane i uczą się ich, rozpoznając strukturę, typowe błędy, niespójności itp.

Drugi przypadek różni się od pierwszego zasadniczo tylko tym, że najpierw czyszczenie przeprowadza się ręcznie i w trakcie tych prac tworzy odpowiednie narzędzia i procedury, które później służyć będą do czyszczenia automatycznego, np. raz dziennie, podczas zasilania hurtowni.

Jak widać, zawsze trzeba dobrze poznać dane i zrozumieć mechanizmy powstawania błędów. Częstą techniką stosowaną podczas czyszczenia danych jest porównywanie różnych ich źródeł, jeśli oczywiście mamy takie do dyspozycji. Jeżeli przedsiębiorstwo wdraża tzw. system zintegrowany w miejsce kilku niezależnych rozwiązań, to bardzo często zdarza się, że te same informacje przechowywane są w różnych bazach, do których zostały niezależnie wprowadzone, nierzadko przez różnych operatorów. Można wtedy zbudować kwerendy porównujące odpowiednie tabele i wyraportować zidentyfikowane rozbieżności, które następnie zostaną szczegółowo sprawdzone. Niestety zdarza się, że tylko jedna baza ma postać elektroniczną, podczas gdy druga stanowi zbiór dokumentów papierowych. Wtedy opisana metoda weryfikacji danych sprowadzałaby się niemal do ponownego ich wpisania do systemu i najprawdopodobniej zostałaby odrzucona jako zbyt pracochłonna.

Możliwe jest jednak zawsze badanie wewnętrznej merytorycznej spójności danych. Przykładowo, jeśli baza zawiera adresy, to można analizować zgodność kodów pocztowych z nazwami miejscowości, których dotyczą, najlepiej posiłkując się zewnętrzną książką kodową (w postaci elektronicznej, rzecz jasna). Numery PESEL możemy z kolei porównywać z datami urodzenia itd.

Dość skuteczną techniką czyszczenia danych jest dokładne przeglądnięcie ich losowo wybranej próbki, w celu zorientowania się, jakiego rodzaju nieprawidłowości mogą w nich wystąpić, a następnie zbudowanie procedur szukających właśnie takich nieprawidłowości (np. liter w numerach telefonów). Przypomina to trochę moduły wykrywania nadużyć (*Fraud Detection*), w które wyposaża się systemy bankowe, bądź telekomunikacyjne, a których zadaniem jest wychwytywanie nietypowych zachowań klientów, opisanych określonymi wzorcami.

Warto zauważyć, że poszukiwanie zanieczyszczonej informacji stanowi w istocie zagadnienie z dziedziny eksploracji danych (data mining), choć z pozoru można by sądzić, że chodzi w nim o coś innego. Eksplorację przedstawia się zazwyczaj jako działania mające na celu odkrycie zawartej w bazach danych wiedzy, mającej postać pewnych związków, prawidłowości lub korelacji, których poznanie może mieć dla przedsiębiorstwa określoną, wymierną wartość [4]. Wydaje się jednak, że jej istotą jest nie tyle poszukiwanie wiedzy, co raczej znajdowanie odpowiedzi na nieprecyzyjnie postawione pytania [5], w rodzaju: „Co jest złego w mojej bazie?”.

Przyjrzyjmy się następującemu przykładowi. W pewnym systemie informatycznym na pełny adres zamieszkania przewidziano jedno długie pole tekstowe. Obecnie przenosimy dane do systemu, gdzie nazwy miejscowości są słownikowane. Załóżmy, że udało się nam wyekstrahować te nazwy i zbudować odpowiedni słownik, w którym oczywiście nie ma już żadnych powtórzeń. Mogą w nim jednak występować literówki („Ludźmierz”, „Ludzmierz”), wskutek których osoby mieszkające w tej samej miejscowości postrzegane są tak, jakby każda z nich mieszkała gdzie indziej. Należy zatem wyczyścić bazę usuwając wspomniane błędy. Nie jest to jednak zadanie proste, jeśli oczywiście nie dysponujemy odpowiednim elektronicznym wykazem nazw. Rozwiązaniem mogłoby być natomiast użycie metody zwanej analizą skupień [4, 5]. Zauważając, że wyrazy są w istocie ciągami znaków, można w matematycznie poprawny sposób zdefiniować odległość pomiędzy nimi. Nie wdając się w szczegóły powiemy tylko, że odległość ta będzie mała dla wyrazów, w których nie zgadza się tylko jedna litera, większa, gdy nie zgadzają się dwie itd. Wyposażeni w taką metrykę możemy, korzystając z określonych algorytmów (np. *K-uśrednień*), poszukać skupień w naszych nazwach. Blisko siebie znajdą się więc przykładowe „Ludźmierz” i „Ludzmierz”, ale też „Koty” i „Kozy” – tu akurat nie ma żadnego błędu. Dalej należałoby więc na przykład analizować kody pocztowe w obrębie skupień i dokładnie przyjrzeć się tylko tym nazwom, dla których są one identyczne.

Jak widać, czyszczenie danych jest poniekąd zajęciem fascynującym, godnym zdolnego detektywa, w zależności od sytuacji stawiającego różne hipotezy i wymyślającego coraz to nowe techniki śledztwa. Bywa jednak, że wymaga ono też pracy żmudnej i wtedy przypomina bardziej zajęcie Kopciuszka niż Sherlocka Holmesa. Niestety...

7. Cudowny lek przyszłości?

W naszych rozważaniach z jednej strony staraliśmy się przekonać czytelnika, jak ważną cechą danych jest ich czystość, z drugiej jednak zaprezentowaliśmy nieco anarchistyczny pogląd, że idealna czystość jest całkowicie nieosiągalna, zaś informacja psuje się niejako samoistnie, na mocy fundamentalnych praw rządzących tym światem. Odpowiadając na pytanie postawione w tytule, spróbowaliśmy udzielić czytelnikowi kilku praktycznych porad, lecz jednocześnie argumentowaliśmy, że zbytnia troska o jakość danych nie wychodzi na zdrowie systemom informatycznym, podnosząc ich koszt i czyniąc je nadmiernie trudnymi w obsłudze. Nieco na marginesie zauważyliśmy też, że człowiekowi jest znacznie łatwiej niż maszynie uporać się z zanieczyszczoną informacją. I do tego właśnie spostrzeżenia chcielibyśmy na moment powrócić. W istocie odporność ludzkiego mózgu na szum informacyjny wydaje się całkiem spora, a to dzięki bardzo specyficznemu sposobowi równoległego przetwarzania informacji. Może zatem zamiast sterylizować dane należałoby raczej pomyśleć o maszynach, które wykażą się większą tolerancją? Bardzo nieudolną – jak na razie – imitacją mózgu są tzw. sztuczne sieci neuronowe, więc może to właśnie one stanowić będą jednostki centralne komputerów przyszłości? Zainteresowanych tematem odsyłamy do obszernej literatury (np. [6, 7, 8]), z której niestety dość jednoznacznie wynika, że przyszłość, o której mowa, wydaje się raczej odległa. Z drugiej jednak strony postęp nauki i techniki jest tak nieprzewidywalny, że wszelkie konkretne prognozy w tym zakresie na ogół narażają tylko ich autorów na śmieszność.

Bibliografia

1. Jaszkiewicz, A.: Inżynieria oprogramowania, Wydawnictwo Helion, 1997, ISBN 83-7197-007-2.
2. Encyklopedia fizyki, Państwowe Wydawnictwo Naukowe, 1972.
3. Beynon-Davies, P.: Systemy baz danych, Wydawnictwa Naukowo-Techniczne 1998, ISBN 83-204-2257-4.
4. Berry M.J.A., Linoff G.: Data Mining Techniques for Marketing, Sales, and Customer Support, Wiley Computer Publishing 1997, ISBN 0-471-17980-9.
5. Wyrozumski T.: Eksploracja danych – dlaczego nie w przemyśle?, VIII Konferencja PLOUG, Zakopane 2002, ss. 123-133.
6. Tadeusiewicz R.: Sieci neuronowe, Akademicka Oficyna Wydawnicza RM 1993, ISBN 83-85769-03-X.
7. Hertz, J., Krogh, A., Palmer, R.G.: Wstęp do teorii obliczeń neuronowych, Wydawnictwa Naukowo-Techniczne 1993, ISBN 83-204-1680-9.
8. Wyrozumski T.: Prognozowanie neuronowe w oparciu o dane ekonomiczne z baz Oracle, VI Konferencja PLOUG, Zakopane 2000, ss. 297-304.