



wczoraj, dziś i jutro

wydziału elektroniki
i technik informacyjnych
politechniki warszawskiej

1951-2001

Zbiór esejów pod redakcją Romana Z. Morawskiego
wydany z okazji Jubileuszu Pięćdziesięciolecia
Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej



Politechnika Warszawska

**Plac Politechniki 1
00-661 Warszawa**

**Wydział Elektroniki
i Technik Informatycznych**

**Nowowiejska 15/19
00-665 Warszawa**

Zbiór esjów pod redakcją Romana Z. Morawskiego
wydany z okazji Jubileuszu pięćdziesięciolecia
Wydziału Elektroniki i Technik Informatycznych Politechniki Warszawskiej

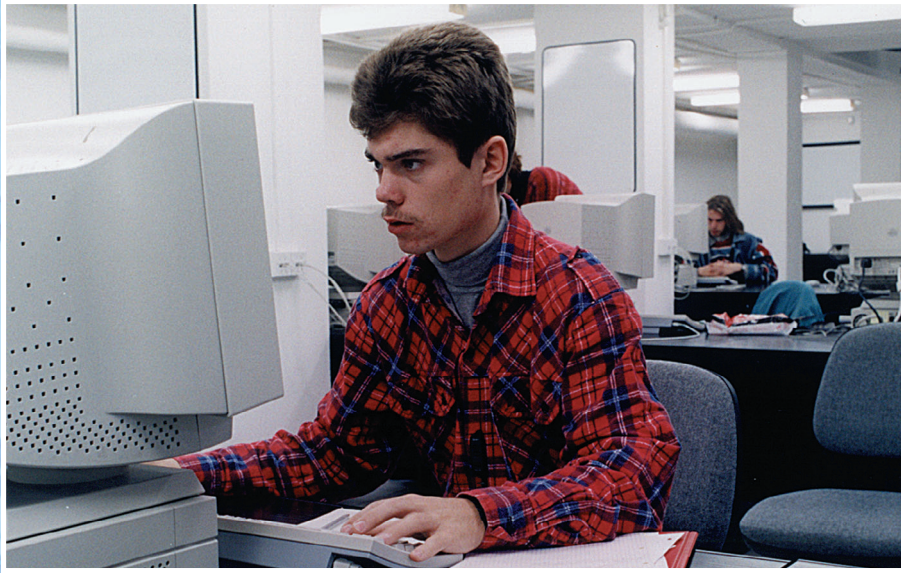
wczoraj, dziś i jutro

wydziału elektroniki i technik informatycznych politechniki warszawskiej

1951-2001

OD REDAKTORA ZBIORU (Roman Z. Morawski)	3
WYDZIAŁ W PERSPEKTYWIE HISTORYCZNEJ (Roman Z. Morawski)	7
SYSTEM KSZTAŁCENIA (Andrzej Kraśniewski)	9
TEORETYCZNE PODSTAWY DISCYPLIN UPRAWIANYCH NA WYDZIALE (Jerzy Osiowski)	29
ELEKTRONIKA (Andrzej Jakubowski)	29
TELEKOMUNIKACJA (Józef Lubacz)	29
INFORMATYKA (Jerzy Mieścicki)	29
AUTOMATYKA, INŻYNIERIA BIOMEDYCZNA, METROLOGIA I SYSTEMY POMIAROWE (Władysław Findeisen, Grażyna Gortat, Krzysztof Malinowski, Zdzisław Pawłowski) .	29
REFLEKSJA JUBILEUSZOWA, CZYLI PYTANIE O PRZYSZŁOŚĆ (Roman Z. Morawski) .	63





informatyka

Bezpośrednim pretekstem do napisania tego eseju jest oczywiście 50. rocznica akademickiej instytucji: Wydziału Elektroniki i Technik Informatycznych. Taka „okrągła” okazja zawsze skłania do podsumowań, wspomnień i prób wyciągnięcia nauk na przyszłość. Sytuacja informatyki różni się jednak od innych dyscyplin uprawianych na Wydziale. Dla elektroniki, telekomunikacji czy radiotechniki data utworzenia pewnej akademickiej instytucji jest raczej jedynie umowną cezurą, ważną głównie dla obchodzącej ten jubileusz społeczności. Tymczasem w przypadku informatyki półwiecze to zawiera jednocześnie prawie całe jej dzieje, nie tylko jako dziedziny badań i nauczania na naszym Wydziale, ale w ogóle na świecie. Co więcej, fascynujący rozwój tej dziedziny — od pierwszych „maszyn matematycznych” w pracowniach uczonych do oplatającej cały świat Sieci — był jednym z najważniejszych zjawisk cywilizacyjnych minionego XX wieku. Nie jest żadną przesadą stwierdzenie, że osiągnięcia w dziedzinie informatyki i jej zastosowań przekształciły nie tylko praktycznie wszystkie dziedziny nauki i techniki, lecz także zmieniły ludzkie obyczaje i sposoby społecznego funkcjonowania, a nawet polityczny układ świata.

Na rozwój tej dziedziny stale oddziaływała historia i polityka, a zwłaszcza technologiczna i polityczna rywalizacja między Wschodem a Zachodem. To oddziaływanie miało zresztą charakter sprzężenia zwrotnego: informatyka była jednocześnie i podmiotem, i ważnym narzędziem rywalizacji. Nawet w skali Wydziału Elektroniki związki te są wyraźnie widoczne: wpływają na losy ludzi, tematykę badań, tendencje rozwojowe. Badania w dziedzinie informatyki na naszym Wydziale (a także dzieje samego Instytutu Informatyki) nie toczyły się przecież w akademickiej „wieży z kości słoniowej”, izolowanej od wydarzeń świata. Będziemy się to starali pokazać w tym eseju.

Na potrzeby naszych rozważań całe pięćdziesięciolecie podzielimy na kilka następujących po sobie, umownych okresów. Ułatwi nam to omawianie losów Instytutu Informatyki, który (zwłaszcza przez pierwsze trzydzieści lat) był tą jednostką Wydziału, w której przede wszystkim uprawiano informatykę. W samym tekście nie będziemy się jednak zbyt ściśle trzymać chronologii, czasami wybiegając w przód, kiedy indziej pozwalając sobie na dygresję, za główny cel mając tematyczną ciągłość opowiadania. Na koniec postaramy się odpowiedzieć na pytanie: dlaczego warto nadal zajmować się badaniami z zakresu informatyki? Cały ten burzliwy pięćdziesięcioletni okres — to przecież dopiero sam początek rozwoju dziedziny, którą uprawiamy.

OD NIEPAMIĘTNYCH CZASÓW DO KOŃCA LAT PIĘCDZIESIĄTYCH XX WIEKU, CZYLI SKĄD TO SIĘ WSZYSTKO WZIĘŁO

NARODZINY WSPÓŁCZESNEJ INFORMATYKI

Można powiedzieć, że narodziny i młodzieńczy, romantyczny okres współczesnej informatyki (choć sama nazwa pojawiła się znacznie później) przypadają na pierwsze dziesięć lat po zakończeniu II wojny światowej. Oczywiście, już znacznie wcześniej pojawiały się pomysły naukowe i rozwiązania techniczne, mające na celu automatyzację obliczeń, formułowanie podstaw teoretycznych procesów obliczeniowych itp., ale nie spowodowały one przełomu

Jerzy Mieścicki

dr inż.
Instytut Informatyki
Politechniki Warszawskiej
Gmach Elektroniki, p. 350
ul. Nowowiejska 15/19
00-665 Warszawa
tel. (48 22) 660-7650

i nie wyzwoliły lawiny inwencji, zainteresowania i żywiołowego rozwoju. Mechaniczny sumator (do dodawania i odejmowania) był znany od dawna: to dzieło francuskiego myśliciela, fizyka i wynalazcy Blaise Pascala (1623–1662). Rozbudowa jego funkcji o mnożenie i dzielenie była potem tylko sprawą udoskonalenia techniki wytwarzania elementów mechanicznych, kół zębatych itp. W latach trzydziestych XIX wieku Charles Babbage (1792–1871) podjął w Anglii próbę (niestety nieudaną) skonstruowania mechanicznej maszyny zdolnej do wykonywania nie pojedynczych operacji, lecz całych *sekwencji* (predefiniowanych przez człowieka, a więc *programowanych*) czterech działań arytmetycznych. Angielski matematyk George Boole (1815–1864) zaksjomatyzował znaną od czasów Arystotelesa logikę dwuwartościową, zapewne nie przypuszczając, że tak powstała *algebra Boole’a* będzie za sto lat jednym z podstawowych narzędzi pracy tysięcy inżynierów na całym świecie. W końcu lat osiemdziesiątych XIX wieku zespół amerykańskich inżynierów pod kierownictwem Hermana Holleritha skonstruował (na potrzeby spisu ludności w USA w 1890 r.) elektromechaniczną maszynę do wykonywania (wstępnie „ustawianych” przez człowieka) sekwencji działań arytmetycznych na masowych danych, wczytywanych z kart dziurkowanych. Takie *maszyny analityczno-liczące* były potem przez dobre 70 lat wykorzystywane na całym świecie w urzędach statystycznych i większych przedsiębiorstwach, np. przy masowym wystawianiu rachunków za telefon, elektryczność, rozliczaniu przewozów kolejowych. Równolegle matematycy opracowywali *metody numeryczne*, pozwalające na możliwie sprawne obliczanie różnorodnych funkcji, rozwiązywanie układów równań itp. Jest oczywiste, że towarzyszyła temu naukowa refleksja nad samym pojęciem funkcji, obliczenia, funkcji obliczalnej, algorytmu. Początek XX wieku — to również wynalazek lampy elektronowej i burzliwy rozwój elektroniki, zwłaszcza w dziedzinie telekomunikacji, radia, a potem telewizji. Na marginesie tych podstawowych zastosowań pojawiają się *układy impulsowe*, w tym w szczególności dziwny układ Eccless-Jordana, wprawdzie zupełnie nieprzydatny we wzmacniaczu czy radiodiodniku, ale interesujący i wymagający wyjaśnienia, dlaczego ma tylko dwa stany stabilne.

Obecnie, z perspektywy lat, można odnieść wrażenie, że tuż przed II wojną światową wszystkie te rozproszone idee i rozwiązania techniczne zaczęły nagle zbiegać się ku sobie i razem gwałtownie dojrzewać, a wybuch wojny i potrzeby działań wojennych jeszcze przyspieszyły ten proces. W roku 1936 brytyjski matematyk Alan Turing formułuje genialny w swej przejrzystości model *wszelkich* obliczeń (a więc nie tylko numerycznych), które jest w stanie efektywnie wykonać zarówno maszyna, jak i człowiek. Wkrótce, w latach II wojny światowej, Turing będzie naukowym kierownikiem zespołu brytyjskich kryptologów z Bletchley Park, odczytujących szyfry nieprzyjaciela. Na podstawie — między innymi — jego pomysłów powstaną w Wielkiej Brytanii i w USA supertajne elektroniczne maszyny do złożonych operacji na symbolach, zdolne do wypróbowania milionów kombinacji w poszukiwaniu klucza szyfru.¹ W roku 1938 w USA Claude Shannon spostrzega, że algebra Boole’a stanowi gotowe, znakomite narzędzie do analizy i syntezy złożonych układów przekaźnikowych, stosowanych w telekomunikacji, przemyśle i transporcie. Niezależnie i równolegle w Niemczech inżynier Konrad Zuse buduje (w latach 1940–41) pierwszą binarną maszynę liczącą, opartą właśnie na zastosowaniu przekaźników. Maszyna działa, ale jest za mała i zbyt wolna; Zuse zamierza zastąpić elektromechaniczne przekaźniki elektronicznymi, lampowymi przerzutnikami Eccless-Jordana. Niestety — dla niego, a chyba na szczęście dla nas — władze niemieckie nie udzielają wsparcia tak dziwnaczemu pomysłowi i projekt Zusego zostaje zaniechany. Tymczasem w Stanach Zjednoczonych do automatyzacji obliczeń na potrzeby wysiłku wojennego przywiązuje się znacznie większą wagę. W 1943 roku rusza tajny projekt ENIAC, którego celem jest zbudowanie wielkiej lampowej maszyny do obliczeń — przede wszystkim — numerycznych (zaawansowane projekty wojskowe, w tym także atomowy projekt Manhattan). ENIAC jest z powodzeniem wykorzystywany w końcowym okresie wojny, a jego odtajnienie i pokazanie opinii publicznej w październiku 1946 roku budzi wielką sensację.

¹ Dziś wiemy, że polscy matematycy: Marcin Rejewski, Henryk Zygałski i Jerzy Różycki rozpracowali zasady konstrukcji niemieckiej maszyny szyfrującej ENIGMA już w 1932 r. Zespół Turinga najprawdopodobniej korzystał z ich osiągnięcia.

Istotnie, było to ogromne osiągnięcie techniczne i pierwsze urządzenie nazwane przez dziennikarzy „mózgiem elektronowym”. Jednak to nie ENIAC jest bezpośrednim przodkiem współczesnych komputerów. Wprawdzie ENIAC był znacznie większy i doskonalszy od ówczesnych maszyn analityczno-liczących, a układy lampowe (w przeciwieństwie do elektromechanicznych) zapewniały mu większą szybkość działania, lecz zasada *zewnętrznego programowania* pozostała: przygotowanie maszyny do wykonania pewnego obliczenia wymagało ręcznego ustawienia („na sztywno”) setek przełączników i pokręteł.

Rewolucyjną modyfikację zaproponowali w 1946 roku von Neumann, Burks i Goldstone z Institute of Advanced Study w Princeton, USA. Niech program — powiedzieli — będzie zapisywany w *pamięci maszyny*, podobnie, jak dane. Niech maszyna cyklicznie pobiera z pamięci — na zmianę — kolejną instrukcję i dane potrzebne do jej wykonania. Oczywiście, trzeba będzie wynaleźć układy pozwalające na pamiętanie większej ilości informacji, ale z tym współczesna elektronika powinna sobie poradzić. Układ sterowania będzie bardziej skomplikowany, ale to się też powinno dać przezwyciężyć dzięki prostocie systemu dwójkowego. Za to, program może być wcześniej przygotowany, na przykład na kartach dziurkowanych, i — gdy trzeba — błyskawicznie wprowadzony do maszyny, dokładnie tak, jak dane. Co więcej, program może modyfikować swój własny przebieg, gdyż do komórek pamięci zawierających instrukcje i adresy ma podobny dostęp, jak do komórek zawierających dane. Daje to niewiarygodne możliwości: przede wszystkim decydowanie na bieżąco o kolejności wykonywania instrukcji (rozkazy skoku), indeksowany dostęp do danych, tworzenie złożonych rozgałęzień i pętli itd. Ponadto, przy tworzeniu programów będzie można wykorzystać fragmenty innych programów, napisanych kiedyś do innych celów. Jeśli tak, to może warto takie *podprogramy* pisać od razu w sposób bardziej uniwersalny i rozmyślnie tworzyć *biblioteki* podprogramów i funkcji?

Ten genialny pomysł: dwójkowa maszyna licząca z pamiętanym programem (*stored-program computer*), zasada cyklu rozkazowego, użycie licznika instrukcji, zasady wyliczania adresu efektywnego itd. — jest przypisywany Johnowi von Neumannowi (1903–1957), jednemu z najwybitniejszych matematyków XX wieku. Warto podkreślić, że mimo niezwykle go postępu technologicznego i wielu późniejszych ulepszeń, praktycznie wszystkie współczesne komputery działają dokładnie tak, jak zaproponował von Neumann (i, dodajmy, robią dokładnie i tylko to, co abstrakcyjna maszyna Turinga). Z późniejszych wynalazków właściwie jedynie komputery sterowane przepływem danych (*data flow computers*) i sieci neuronowe wyłamują się z zasad architektury von Neumanna. Pierwsze z nich odegrały większej praktycznej roli, drugie — obiecujące i wciąż w początkowym okresie rozwoju — są obecnie z zasady symulowane przy użyciu komputerów o klasycznej, „von-Neumannowskiej” architekturze.

Tak ruszyła lawina. Koncepcję von Neumanna szybko ucieleśniono w postaci komputera Mark I (Uniwersytet Harvarda), podobne projekty są wkrótce realizowane w wielu ośrodkach badawczych i uniwersytetach, gdzie przyciągają najbardziej twórczych inżynierów i matematyków. Wybitni naukowcy są świadomi, że otworzyły się nowe możliwości, przyprowadzające doprawdy o zawrót głowy. Alan Turing, przekonany, że jego model obliczeń opisuje dokładnie wszelką intelektualną działalność człowieka, wprowadza pojęcie *sztucznej inteligencji* i przewiduje, że w ciągu kilku (no, może kilkunastu) lat maszyny będą samodzielnie rozumować, a człowiek nie będzie w stanie odgadnąć, czy rozmawia z komputerem, czy z innym człowiekiem. Sam Turing poświęca się badaniom nad nową dziedziną, którą nazwał *biologią matematyczną*. Norbert Wiener zauważa (w 1948 r.) głębokie podobieństwa procesów komunikowania się i sterowania w maszynach, organizmach żywych i w całych społeczeństwach, proponując, by badania nad nimi zintegrować w jedną dziedzinę nauki: *cybernetykę*. W tym samym 1948 r. wspomniany już wcześniej Claude Shannon formułuje (wspólnie z Warrenem Weaverem) podstawy współczesnej matematycznej *teorii informacji*. Prace McCullocha, Pittsa, Kleenego stwarzają podstawy matematycznej *teorii automatów*. Niedługo potem, Noam Chomsky publikuje matematyczny model gramatyki kombinatorycznej. Szybko okazuje się, że jego podejście, uzupełnione pojęciami z dziedziny teorii automatów, tworzy zręby *lingwistyki matematycznej*, umożliwiającej opis języków zarówno sztucznych, jak naturalnych, a w szczególności tworzenie sztucznych języków programowania. Tak więc, na przełomie czterdziestych i pięćdziesiątych lat XX wieku już było wiadomo,

że ludzkość dostała do ręki niezwykle sprawne i praktyczne narzędzie do obliczeń. Co więcej, badaczom wydaje się, że dzieli ich jedynie mały krok od inteligentnych robotów, cybernetycznych organizmów (*cyborgów*) i od pełnego zrozumienia mechanizmów działania ludzkiego mózgu.

ŻELAZNA KURTYNA

Te zdumiewające informacje o osiągnięciach docierają sporadycznie do opinii publicznej w sensacyjnych dziennikarskich doniesieniach o mózгах elektronowych i na kartach literatury fantastyczno-naukowej. Mimo wszystko są tylko ciekawostką i nie mają one jeszcze większego wpływu na życie codzienne poszczególnych ludzi ani na problemy całych społeczeństw. Są ważniejsze sprawy: świat zostaje rozcięty żelazną kurtyną, a dwa wrogie obozy polityczne grożą sobie wzajemnie szybko rosnącymi arsenałami broni masowej zagłady. Jednocześnie, mimo atomowego zagrożenia, społeczeństwa Zachodu korzystają z dobrodziejstw powojennego ożywienia gospodarczego i cieszą się wynalazkami, które odmieniają życie codzienne. Tworzywa sztuczne, nylonowe pończochy, penicylina, telewizja, rozwój motoryzacji, możliwość podróżowania. Po drugiej, naszej stronie żelaznej kurtyny — świat jest znacznie mniej barwny. Odbudowa zniszczeń wojennych, elektryfikacja, likwidacja analfabetyzmu, Nowa Huta i przemysł ciężki, i narastający absurd gospodarki planowej.

Oba główne światowe mocarstwa: USA i ZSRR, odgradzają się od siebie murem wrogości i podejrzliwości. U nas, i w innych krajach tzw. obozu socjalistycznego, wszelkie zachodnie nowinki są politycznie podejrzane. Kolorowe skarpetki i przydługie włosy oznaczają niedopuszczalny „amerykański styl życia”. Również żadnych „mózgów elektronowych” nie ma i być nie powinno. Z niewytłumaczalnych ideologicznych powodów, aż do „odwilży” po śmierci generalissimusa Stalina w 1953 r., cybernetyka jest uznawana za „burżuazyjną łże-naukę”. Z pewnością radzieccy naukowcy i inżynierowie (podobnie, jak ich polscy koledzy) po cichu interesują się zachodnimi osiągnięciami, kryjąc się za zasłoną matematyki czy elektroniki. Dopiero w połowie lat pięćdziesiątych zaczną na dobre odrabiać straty: powstaną instytuty cybernetyki i „elektronicznej techniki obliczeniowej”, będzie się wydawać w pospiesznych, masowych tłumaczeniach prace zachodnich autorów, powstaną własne projekty „maszyn matematycznych”.

Tymczasem na Zachodzie w pierwszej połowie lat pięćdziesiątych komputery wkraczają do gospodarki. Powstają FORTRAN i COBOL, do dnia dzisiejszego bodaj najbardziej rozpowszechnione w praktyce języki programowania. Wkrótce — w połowie lat pięćdziesiątych — lampy elektronowe zastępuje technika tranzystorowa, oznaczająca miniaturyzację i znaczne zmniejszenie poboru energii, ale przede wszystkim — zwiększenie niezawodności, a więc możliwość realizacji coraz większych i bardziej złożonych urządzeń.

BARIERA NIEZAWODNOŚCIOWA

Mało kto (z nie-elektroników) zdaje sobie sprawę z faktu, że podstawowym ograniczeniem rozwoju nowej techniki komputerowej nie był ani niedostatek ludzkiej wyobraźni, ani bariera kosztów, lecz *bariera niezawodności*. Spowodowane to było ogromną złożonością układu, nieporównanie większą niż w przypadku radioodbiornika, telewizora czy stacji radarowej. Początkowo urządzenia były niesłychanie zawodne. Główną przyczyną nieustannych awarii były nie tylko delikatne lampy próżniowe, lecz przede wszystkim setki tysięcy punktów połączeniowych: miejsca lutowania, a zwłaszcza mechaniczne styki — lamp z podstawkami, pakietów z gniazdami itd. Nagrzewanie się maszyny podczas pracy, drgania mechaniczne, przeciągi i trzaskanie drzwiami powodowały, że któryś z dziesiątków tysięcy styków rozłączał się, że ktośś z lutowań okazywało się niedokładne i cały system przestawał poprawnie działać. Międzyawaryjny czas maszyny bywał rzędu kilkunastu czy kilkudziesięciu minut, a nieprzerwane wielogodzinne obliczenia były ewenementem. W ten sposób bariera niezawodnościowa ograniczała skutecznie rozwój dziedziny, eliminując z góry lub każąc odłożyć na półkę pomysły wielu rozwiązań, wymagających zbyt złożonego sprzętu, zbyt wielkiej pamięci operacyjnej lub dłuższego czasu obliczeń, wielokrotnie przekraczającego czas międzyawaryjny maszyny.

ZACZĄTKI PÓŹNIEJSZEGO INSTYTUTU INFORMATYKI NA NASZYM WYDZIALE

Wróćmy do Politechniki Warszawskiej i nowo utworzonego Wydziału Łączności. W 1951 roku Kierownikiem Katedry Radiofonii na tym Wydziale zostaje Antoni Kiliński, czterdziestodwuletni wówczas magister inżynier radiotechnik, przed wojną asystent Mieczysława Pożaryskiego na Wydziale Elektrycznym PW i konstruktor w Państwowym Instytucie Telekomunikacyjnym. Po Powstaniu Warszawskim, pobycie w oflagu i powrocie do Polski — jako potrzebny nowej władzy inżynier — zostaje powołany do Ludowego Wojska Polskiego, pracuje w wojskowych instytucjach centralnych, potem trafia do powstającej właśnie Wojskowej Akademii Technicznej w Warszawie i wreszcie — do Politechniki Warszawskiej. Karierę wojskową kończy kilka lat później, z własnej woli, już w stopniu pułkownika, jako ceniony fachowiec i sprawdzony organizator, a do tego działacz ustosunkowany w kręgach władzy i wojska. Zapewne nie pozostało to bez wpływu na tematykę prac Katedry i dynamikę jej rozwoju. W 1953 r. kierowana przez Kilińskiego jednostka staje się Katedrą i Zakładem Konstrukcji Telekomunikacyjnych i Radiofonii. Początkowo grupuje ona niewielki zespół inżynierów elektroników i mechaników, ma własny personel techniczny oraz warsztat elektryczny i mechaniczny, ponieważ — zgodnie z nazwą — jest jednostką powołaną przede wszystkim do konstruowania specjalistycznej aparatury elektronicznej, przewidzianej do produkcji przemysłowej. Specjalnością Katedry szybko stają się jednak nie zwyczajne „konstrukcje telekomunikacyjne i radiofoniczne”, lecz zaawansowane urządzenia z kręgu (jak wtedy mówiono) „techniki impulsowej”, a w szczególności przeliczniki elektronowe, produkowane na potrzeby badań jądrowych. Do 1960 roku wyprodukowano 642 sztuki tych urządzeń (z czego ponad 40 na eksport), a ich dalszą produkcję przejął przemysł.

POCZĄTKI ZAINTERESOWANIA KOMPUTERAMI W POLSCE

Już wcześniej, w 1948 r., z inicjatywy Kazimierza Kuratowskiego (dyrektora Państwowego Instytutu Matematycznego i uczonego o światowej sławie) powstaje Grupa Aparatów Matematycznych (GAM), skupiająca matematyków i inżynierów (zwłaszcza elektroników) zafascynowanych możliwościami komputerów. Kierownikiem GAM zostaje Henryk Greniewski, pierwszymi pracownikami są: Romuald Marczyński, Krystyn Bochenek, Leon Łukaszewicz. Sami absolwenci Politechniki Warszawskiej. Dołączą do nich wkrótce: Zdzisław Pawlak, Antoni Mazurkiewicz, Józef Winkowski — dziś profesorowie, dobrze znani w środowisku informatyków — i wielu innych. Podejmują oni prace koncepcyjne nad techniką programowania, a wkrótce również próby skonstruowania polskiej cyfrowej „maszyny matematycznej”. Pierwsza eksperymentalna konstrukcja (GAM 1, Zdzisław Pawlak, 1950 r.) nigdy nie była praktycznie użyta do obliczeń. Również następna (EMAL, Romuald Marczyński, lata 1953–55) nie została w pełni ukończona, przede wszystkim ze wspomnianych wyżej powodów niezawodnościowych. Dopiero maszyna EMAL 2 (lata 1957–58) miała dostateczną niezawodność i była pierwszym, naprawdę używanym komputerem nowo powstałego Centrum Obliczeniowego PAN.²

POCZĄTEK PRAC NAD KOMPUTERAMI W KATEDRZE PROFESORA KILIŃSKIEGO

Prace nad „maszynami matematycznymi” budzą zainteresowanie w polskim środowisku naukowym i akademickim. Nie omija ono Katedry Konstrukcji Telekomunikacyjnych i Radiofonii. W 1956 r. Gerard Kudelski buduje tu Programowany Automat Rachunków Krakowianowych (PARK), a cały zespół doskonali się w sztuce konstruowania niezawodnych złożonych cyfrowych urządzeń elektronicznych. Zebrane doświadczenia powodują, że w drugiej połowie lat pięćdziesiątych Antoni Kiliński podejmuje ze swym zespołem zadanie zbudowania własnej

² Więcej o polskich projektach informatycznych z tamtych lat można dowiedzieć się z dwuczęściowego artykułu Jana Madeya i Macieja M. Sysło: *Początki informatyki w Polsce*, Informatyka, nr 9 i 10, 2000.

„maszyny matematycznej”, a więc prawdziwego, programowanego komputera. Powstaje najpierw model laboratoryjny EMC (po prostu: Elektroniczna Maszyna Cyfrowa) — a następnie prototyp (1960 r.) i pięć egzemplarzy maszyny UMC 1 (Uniwersalna Maszyna Cyfrowa)³. Część elektroniczną i konstrukcję mechaniczną wykonano również całkowicie samodzielnie (Zdzisław Braun, Jerzy Połoński, Kazimierz Terlecki i inni), a prototyp i seria próbna okazały się tak udane, że władze polskiego przemysłu podjęły decyzję o podjęciu produkcji tych maszyn we wrocławskich zakładach elektronicznych ELWRO. Wkrótce, w latach 1962–64, w ELWRO wyprodukowano serię (aż!) 25 sztuk maszyn UMC 1. Był to pierwszy komputer wytwarzany w Polsce na skalę przemysłową.⁴ W ten sposób zespół prof. Kilińskiego wszedł w lata sześćdziesiąte ze znaczącym sukcesem w swym konstrukcyjnym dorobku.

LATA SZEŚCZDZIESIĄTE

POCZĄTEK „SKOKU TECHNOLOGICZNEGO” W LATACH SZEŚCZDZIESIĄTYCH W USA

W październiku 1957 roku Związek Radziecki wprowadza na orbitę pierwszego sztucznego satelitę Ziemi — Sputnika, wkrótce potem drugiego, tego z suczką Łajką na pokładzie. To znak, że wschodni konkurent, który do tej pory tylko doganiał Zachód w dziedzinie nauki i techniki, zaczął wyprzedzać Stany Zjednoczone w nowoczesnej technologii kosmicznej i raketowej. Wprawdzie Amerykanie natychmiast podejmują wyścig i również umieszczają swe satelity w przestrzeni kosmicznej, ale to Rosjanie (w kwietniu 1961 roku) odnotowują kolejny sukces: okrążenie Ziemi przez pierwszego w historii kosmonautę, Jurija Gagarina. Ameryka musi podjąć wyzwanie i odzyskać inicjatywę, inaczej przegra rywalizację. W 1961 roku prezydent USA — John F. Kennedy — rzuca rozpalające wyobraźnię hasło: zrobimy wszystko, żeby do końca tej dekady Amerykanie wylądowali na Księżycu i bezpiecznie powrócili na Ziemię. Nakłady na badania naukowe i rozwój nowych technologii rosną znacząco, a dekada lat sześćdziesiątych XX wieku staje się okresem bardzo wyężonej pracy i ogromnego rozwoju, naukowego i technicznego.

Obecnie, z perspektywy czasu, wiemy, że to właśnie wtedy uruchomiono procesy, które ćwierć wieku później przyczyniły się do upadku systemu komunistycznego i gruntownej zmiany politycznego porządku świata. Wiemy też, że to ani udane lądowanie na Księżycu w lipcu 1969 roku, ani późniejsze sukcesy amerykańskiej astronautyki nie były tymi czynnikami, które zadecydowały o ostatecznym wyniku rywalizacji między mocarstwami. Rolę tę odegrały przede wszystkim techniki informacyjne, w tym — informatyka i jej zastosowania.

TWORZENIE ZESPOŁU INFORMATYKÓW NA WYDZIALE

Już w końcu lat pięćdziesiątych pracownicy Katedry i Zakładu prof. Kilińskiego zaczynają prowadzić dla studentów ówczesnego Wydziału Łączności wykłady i laboratoria z programowania, organizacji maszyn cyfrowych, arytmetyki dwójkowej (Jerzy Zieliński, Wojciech Jaworski, Wincenty Balasiński, Zdzisław Braun) czy projektowania układów logicznych (Kazimierz Bieńkowski). W 1962 roku zostaje utworzona specjalność Maszyny Matematyczne. W tymże roku dyplom o tej specjalności otrzymuje pierwszych pięciu magistrów inżynierów, rok później — następnych kilkanaście osób. Co roku spora grupa absolwentów pozostaje w Uczelni, bądź jako nauczyciele akademicy w Katedrze, bądź jako konstruktorzy i progra-

³ Warto wspomnieć, że projekt logicznej organizacji maszyny był dziełem Zdzisława Pawlaka, który przeszedł do Katedry z PAN. Niedługo potem jednak z Katedry odszedł, by później, w 1989 r., a więc po 30 latach, już jako profesor zwyczajny, powrócić na stanowisko dyrektora Instytutu Informatyki.

⁴ Argumentem, który przeważał na korzyść UMC 1, była właśnie jej niezawodność, wysoka w porównaniu z pierwszymi maszynami serii ODRA 1000, zaprojektowanymi i budowanymi wówczas w ELWRO.

miści w Zakładzie Doświadczalnym. Wkrótce pierwsi asystenci z KBMM uzyskają stopnie doktora (jako pierwszy Kazimierz Bieńkowski, 1962 r.) i doktora habilitowanego (Konrad Fiałkowski, Jacek Bańkowski, 1966 r.). Pracę podejmują osoby, które do dziś są znanymi nauczycielami akademickimi w obecnym Instytucie Informatyki: Jerzy Mieścicki, Jan Zambrodzki, Jan Bielecki, Andrzej Skorupski. W 1963 roku Katedra i Zakład KTR zostają przekształcone w — odpowiednio — Katedrę Budowy Maszyn Matematycznych i Zakład Doświadczalny BMM. Prof. Kiliński pozostaje kierownikiem obu.

Tematyka prac naukowych, które owocowały tymi doktoratami i habilitacjami, dotyczyła typowych problemów ówczesnej techniki komputerowej: arytmetyki dwójkowej, projektowania elektronicznych układów cyfrowych, a także sieci logicznych i układów sekwencyjnych. Sam Antoni Kiliński był specjalistą z dziedziny niezawodności i jakości, więc rozprawy te poruszały zazwyczaj niezawodnościowe aspekty problemu: projektowanie układów z uwzględnieniem rozrzutu parametrów, konstruowanie niezawodnych układów z zawodnych elementów, odporność układów logicznych na losowe błędy itp. Zaznaczyliśmy już wcześniej, że była to tematyka w tamtym okresie bardzo ważna.

Mimo, że Katedra Budowy Maszyn Matematycznych już w końcu lat sześćdziesiątych zaczyna przypominać swą strukturą personalną typową akademicką „piramidę” — z profesorem, docentami, adiunktami, asystentami, jej styl działania nie jest typowo akademicki. „Akademicka” jest z pewnością dbałość o poziom nauczania: merytoryczną treść zajęć, poziom laboratoriów, organizację zajęć, materiały do laboratoriów. Jednocześnie Kiliński wyznawał pogląd, że wszyscy pracownicy, niezależnie od stopnia i stanowiska (a także od tego, czy są zatrudnieni jako nauczyciele akademicy, czy jako pracownicy Zakładu Doświadczalnego), są przede wszystkim inżynierami i mają kształcić inżynierów. Dlatego wszyscy powinni brać udział we wspólnych projektach badawczych, które są podstawowym źródłem kompetencji i doświadczenia, a inżynierowie z Zakładu Doświadczalnego powinni prowadzić zajęcia dydaktyczne na równi z asystentami i adiunktami. Tak też się działo. Pozycja i osobowość kierownika Katedry narzucały przekonanie, że teoretyczna działalność naukowa, publikowanie wyników w czasopiśmie i na konferencjach naukowych, a nawet naturalne w akademickiej karierze prace doktorskie i habilitacyjne w istocie ustępują co do „prawdziwej” wartości sprawdzalnym, namacalnym osiągnięciom konstrukcyjnym i programistycznym.⁵ Również (zwłaszcza w początkowym okresie, gdy studenci nie byli tak liczni) zajęcia laboratoryjne odbywały się najczęściej w roboczych pomieszczeniach Zakładu Doświadczalnego.

PRACE BADAWCZE I KONSTRUKCYJNE W LATACH SZEŚCZDZIESIĄTYCH

Całe lata sześćdziesiąte były dla tego zespołu okresem poszukiwań i inwencji, zarówno w dziedzinie organizacji i architektury maszyn, jak metod programowania i technologii budowania komputerów. W początku dekady inwencji tej nie ograniczały jeszcze (prawie) żadne standardy dotyczące formatów danych, kompatybilności języków programowania i systemów operacyjnych, interfejsów urządzeń zewnętrznych itd. Jednak, jeśli komputer miał być rzeczywiście zbudowany i sprawdzony w praktyce (a takie były aspiracje zespołu), projektanci i programiści nieustannie musieli borykać się z małą pojemnością pamięci operacyjnej, prymitywnością urządzeń zewnętrznych, trudnościami z uzyskaniem elementów o odpowiednich parametrach, a także z problemami niezawodnościowymi, cieplnymi i energetycznymi, wynikającymi ze złożoności układu.

W takich warunkach, wykorzystując precyzyjnie wytoczone w warsztacie mechanicznym Zakładu metalowe walce, głowice magnetyczne własnej konstrukcji i pistolet natryskowy, wyprodukowano ponad 50 sztuk pamięci bębnowej (Wojciech Łągwiński), która wówczas była jeszcze używana jako pamięć operacyjna. Opanowano własną technologię montażu i testowania układów logicznych (lampowych, a potem tranzystorowych) i zestawiania ich w panele, jednostki, szafy. Tworzyło to techniczną bazę realizacji projektów. Oprócz wspo-

⁵ Należy jednak przypomnieć, że sam prof. Kiliński był promotorem ponad 30 doktoratów, w początkowym okresie głównie o tematyce niezawodnościowej, wykonanych w uczelniach i instytutach całej Polski.

mnianej maszyny UMC 1, na początku lat sześćdziesiątych zbudowano prototyp maszyny do przetwarzania danych administracyjnych AMC 1 (Wincenty Balasiński, Aleksandra Wierusz i inni) i równoległe ulepszoną, tranzystorową wersję UMC 1, nazwaną UMC 10 (lata 1964–65, już z pamięcią ferrytową). Prace nad prototypem AMC 1 zakończono w 1966 r., bez kontynuacji, natomiast UMC 10 wykonano jeszcze w trzech egzemplarzach, w tym dla Wydziału Geodezji i Kartografii PW oraz Państwowego Instytutu Hydrologiczno-Meteorologicznego.

Organizacja projektowanych i budowanych komputerów oparta była na oryginalnych rozwiązaniach: minus-dwójkowej reprezentacji liczb (Zdzisław Pawlak) i konsekwentnym zastosowaniu mikroprogramowania poziomego (UMC 1 i 10), oryginalnej zasadzie kodowania danych numerycznych i alfanumerycznych oraz znacznej autonomii układów wejścia-wyjścia (AMC 1). Również własnej produkcji były elementy *software'u*: np. język programowania W 20 dla maszyn UMC, program zarządzający AMC, zawierający elementy współczesnych systemów plików (dla pamięci taśmowej).

Bardzo istotną rolę w dziejach Katedry i Zakładu odegrały specjalizowane maszyny cyfrowe ANOPS (do inicjatorów należeli Konrad Fiałkowski i Jacek Bańkowski), które w kolejnych, stale ulepszanych wersjach były wytwarzane w Zakładzie Doświadczalnym przez dwadzieścia lat. Ich zadaniem była rejestracja i cyfrowa obróbka sygnałów elektrycznych wytwarzanych przez organizm człowieka w toku eksperymentów naukowych i medycznej praktyki diagnostycznej. Prace nad maszynami ANOPS rozpoczęto w 1965 r., wspólnie z lekarzami i naukowcami z warszawskiej Akademii Medycznej (Klinika Neurologiczna Ireny Hausmanowej-Petrusewicz). W ich rezultacie w latach 1967–70 zbudowano 15 sztuk urządzeń ANOPS 1 (jeszcze w technice lampowej), a potem (w latach 1970–75) 13 egzemplarzy (tranzystorowych) maszyn ANOPS 10. Stały się one wyposażeniem diagnostycznym klinik i oddziałów neurologicznych większych szpitali w Polsce. Od 1972 r. wytwarzano te komputery z coraz większym udziałem układów scalonych średniej skali integracji i o coraz bogatszych funkcjach (ANOPS 100 i 101). Cieszyły się one dobrą opinią i zasłużonym zainteresowaniem świata medycznego. Dość powiedzieć, że do początku lat osiemdziesiątych w Zakładzie Doświadczalnym Instytutu wyprodukowano 84 egzemplarze ANOPS 101, z tego wiele na eksport: do ZSRR, Czechosłowacji, NRD, ale także do USA (4 sztuki), Kanady i RFN.

Zespół inżynierów Instytutu (Michał Wołyński, Michał Rawski i inni) znakomicie rozumiał się z klinicystami i lekarzami, ci z kolei coraz lepiej poznawali możliwości współczesnej techniki komputerowej i korzyści, jakie wynikają z jej stosowania w badaniach i w praktyce medycznej. Gdy potem — w latach 1981–82 — w klinice profesor Hausmanowej opracowano nową metodę analizy czynności biologicznej mięśni, znalazła ona szybko odbicie w nowej wersji maszyny ANOPS 105, której znów wyprodukowano ponad 30 sztuk. W drugiej połowie lat siedemdziesiątych doświadczenia te wykorzystano również w innej ważnej gałęzi medycyny, a mianowicie kardiologii. W szczególności zaprojektowano i wykonano w krótkich seriach urządzenia KARDIO 78 i KARDIO 80 do nieinwazyjnego badania układu przewodzącego serca. Potem, już w latach 1981–85, kolejne wersje wspomnianych urządzeń realizowano w technice mikroprocesorowej, opierając się na Modułowym Systemie Mikroprocesorowym (MSM), opracowanym pod kierownictwem Andrzeja Skorupskiego. Tak został wykonany ANOPS 205 (Aleksander Wigura, Witold Żaba), a potem wreszcie ANOPS-KARDIO 85. W sumie, przez wspomniane dwudziestolecie, wyprodukowano ponad 150 sztuk różnych urządzeń typu ANOPS, z czego ponad 80 sztuk na eksport. Kres tej działalności położyła w istocie dopiero zapaść ekonomiczna naszego kraju w drugiej połowie lat osiemdziesiątych. Ale to już inna, późniejsza historia.

Drugą dziedziną, ważną dla praktycznej działalności konstruktorskiej i programistycznej zespołu, była geodezja i kartografia. Jej potrzeby obliczeniowe spowodowały, że już w pierwszej połowie lat sześćdziesiątych nawiązano współpracę z Katedrą Geodezji PW (prof. Jerzy Gaździcki). Wspólnie opracowane oprogramowanie numeryczne dla UMC 1 i UMC 10 zaowocowało projektem oryginalnego specjalizowanego urządzenia GEO 1 (Zbigniew Dudek). Choć „w środku” był to prawdziwy programowany komputer — z punktu widzenia użytkownika był to prosty automat obliczeniowy, wyposażony w zestaw typowych procedur numerycznych, najbardziej przydatnych w praktyce geodezyjnej (Zbigniew Zorski). Pracownik w przedsiębiorstwie geodezyjnym nie musiał więc znać się na komputerach czy programowa-

niu: uruchamiał owe programy za pomocą przejrzystego układu przycisków, zgodnie z instrukcjami udzielanymi mu za pośrednictwem swego rodzaju podświetlanego menu. W latach 1968–70 wykonano 11 sztuk maszyn GEO 1, potem wytwarzano ich ulepszoną wersję GEO 2. Współpraca z geodezją miała również swą dalszą historię: system GEO 20 oraz GEO 3. Powiemy o tym dalej.

Równocześnie były opracowywane i wykonywane inne unikatowe urządzenia, jak na przykład miernik ładunku bazy tranzystora (Marian Łakomy) czy tester układów scalonych TTL (Jan Zabrodzki, Andrzej Skorupski, Janusz Sosnowski).

OWOCE „SKOKU TECHNOLOGICZNEGO” LAT SZEŚĆDZIESIĄTYCH W USA

W Stanach Zjednoczonych dekada intensywnej pracy badawczej, ogromne nakłady na badania, prace rozwojowe i nowe technologie przyniosły znaczące efekty. W wielu dziedzinach techniki dokonała się prawdziwa rewolucja. Dotyczy to także (a może przede wszystkim) informatyki. Najbardziej widocznym sukcesem tamtych lat było opanowanie technologii układów scalonych, najpierw małej, a potem średniej skali integracji (SSI, MSI). Dzięki nim komputery stały się nie tylko mniejsze i zużywały mniej energii, ale przede wszystkim były znacznie bardziej niezawodne. Umożliwiło to realizację bardziej złożonych pomysłów, w szczególności — na bezawaryjne i bezpieczne kontrolowanie przebiegu wielodniowej misji statku załogowego Apollo. To był jednak tylko spektakularny wyczyn, pobudzający wyobraźnię społeczną. Ważniejsze, długofalowe konsekwencje owego skoku technologicznego odczuliśmy tu, na Ziemi.

W połowie lat sześćdziesiątych w światowej informatyce wyraźnie krystalizują się tendencje rozwojowe, które w istotny sposób zaważą na losach tej dziedziny. Duże systemy komputerowe (*mainframes*) przechodzą znamiennej ewolucję, której dobrą ilustracją jest IBM System/360. Oto System/360 nie jest już jednym komputerem: jest zbiorem pasujących do siebie modułów, cegiełek, z których użytkownik zamawia i konstruuje odpowiadającą mu *konfigurację*. Pasują one do siebie, gdyż — po pierwsze — są podporządkowane wspólnemu schematowi organizacji i mają standardowe interfejsy, po drugie — są zarządzane przez wspólny, również modularnie rozbudowywany system operacyjny OS/360.

Sukces koncepcji S/360 sprawił, że w krajach RWPG podjęto (w 1968 r.) zamiar skopionowania rodziny maszyn IBM S/360 przy zastosowaniu technologii dostępnej w krajach obozu socjalistycznego w taki sposób, aby były one zgodne z pierwowzorem na poziomie listy rozkazów języka maszynowego. Miało to umożliwić zastosowanie zarówno systemu operacyjnego OS/360, jak i całego oprogramowania opracowanego przez IBM (od razu gotowego) w praktyce po naszej stronie „żelaznej kurtyny”. Nie trzeba dodawać, że nikt nie pytał IBM o zgodę, nikt nie otrzymał (przynajmniej legalną drogą) żadnej dokumentacji sprzętu ani kodu źródłowego oprogramowania. Prace nad tym systemem (zwanym Jednolitym Systemem Elektronicznych Maszyn Cyfrowych — „Riad”), „rozpisane” na wszystkie kraje RWPG, na wiele lat obciążały potencjał projektowy i technologiczny tych krajów pracą żmudną, intelektualnie wtórną, spóźnioną i o małym prawdopodobieństwie sukcesu. Istotnie, przedsięwzięcie nie zakończyło się sukcesem i nie zrewolucjonizowało gospodarki ani zarządzania w krajach RWPG.

Bardzo ważną nowością jest również pojawienie się *minikomputerów*: maszyn z założenia małych i tanich, służących nie do obliczeń numerycznych i przetwarzania danych administracyjnych, lecz raczej do sterowania urządzeniami pomiarowymi, produkcyjnymi, telekomunikacyjnymi itp. oraz do pełnienia pomocniczych funkcji (np. wstępnej edycji danych wejściowych) w większych systemach komputerowych.⁶ Pod bokiem znanych komputerowych firm, takich jak: IBM, UNIVAC czy CDC, wyrastają liczne firmy produkujące gotowe systemy minikomputerowe, ale również oddzielne („luzem”) procesory, pamięci, dyski, urządzenia zewnętrzne, czujniki i elementy wykonawcze przystosowane do współpracy z minikomputerami. Pojawia się nowy typ producentów, tzw. OEM — *Original Equipment Manufacturers*, którzy kupują te elementy *hardware’u* komputerowego (być może u różnych

⁶ Za minikomputer uważano wtedy maszynę, która kosztuje mniej niż 10 000 (ówczesnych!) dolarów.

wytwórców), oprogramowują je i integrują w jeden „oryginalny” system: pomiarowy, laboratoryjny, telekomunikacyjny itd. Minikomputery zrewolucjonizowały całą zachodnią gospodarkę. Dzięki nim technika komputerowa wydostała się z ośrodków obliczeniowych, centrów badawczych i uniwersyteckich, najbogatszych przedsiębiorstw i banków — i trafiła do hal produkcyjnych, biur, laboratoriów, telekomunikacji, mediów, usług.

Oba naszkicowane wyżej zjawiska pozostają w ścisłym związku z postępami w dziedzinie technologii układów scalonych, najpierw małej, a potem średniej skali integracji. Bez nich nie opanowano by masowej, różnorodnej produkcji systemów i ich podzespołów. Jednak trzecim ważnym zjawiskiem drugiej połowy lat sześćdziesiątych jest wyraźny postęp w dziedzinie koncepcji, metod i narzędzi do tworzenia oprogramowania. Obok „zdroworozsądkowego” FORTRAN-u i COBOL-u (które powstały jeszcze w latach pięćdziesiątych) krystalizuje się paradygmat programowania strukturalnego, ucieleśniany przez ALGOL 68, potem Pascal. Znacznie rozwijają się metody kompilacji, pojawia się wiele nowych języków programowania, opartych na nowych koncepcjach, które po latach ewolucji stworzą podwaliny pod tak dziś powszechne programowanie obiektowe. Wiele uwagi poświęca się mechanizmom zarządzania zasobami systemu (np. mechanizm semafora, Edsger Dijkstra, 1968 r.), które mają zasadnicze znaczenie dla programowania współbieżnego i dla tworzenia nowoczesnych systemów operacyjnych. Wreszcie, w tymże 1968 r., w Bell Laboratories zostaje opracowany język C (Brian Kerningham, Dennis Ritchie), stanowiący do dziś (z późniejszymi modyfikacjami: C+, C++, również w znacznym stopniu Java) najbardziej powszechnie używane narzędzie programistyczne. Zespół w Bell Labs opracuje wkrótce i uruchomi system operacyjny UNIX (Ken Thompson), który do chwili obecnej jest źródłem inspiracji dla twórców systemów operacyjnych.

W 1969 roku miało miejsce jeszcze jedno wydarzenie, którego znaczenia w przyszłości dla całej naszej cywilizacji nie przewidywał wtedy nikt. Oto w Stanach Zjednoczonych uruchomiono pierwsze cztery węzły sieci komputerowej ARPA. Łączyła ona początkowo instytucje uczestniczące w wojskowych projektach zarządzanych przez Agencję Zaawansowanych Projektów Badawczych Departamentu Obrony USA. ARPA była od początku projektowana jako sieć zdolna do rozrastania się i istotnie — zaczęła się rozrastać bardzo szybko. Niebawem zaobserwowano (z zaskoczeniem i początkowo z wyraźną dezaprobatą), że znaczną część ruchu w sieci generują nie te duże komputery, dla których współpracy sieć zaprojektowano, lecz ich operatorzy i programiści, którzy wymieniają między sobą pozdrowienia, plotki i wiadomości o pogodzie, przesyłają pliki z dokumentacją i dzielą się pomysłami. Taki był początek usług pocztowych w sieci. Dziś wiemy, do czego to — po dwudziestu pięciu latach rozwoju — doprowadziło: do World Wide Web, pajęczyny oplatającej cały świat, uważanej za najważniejsze zjawisko cywilizacyjne przełomu XX i XXI wieku.

I, dodajmy, 20 lipca 1969 roku Neil Armstrong rzeczywiście dotknął stopą Księżycą, po czym wraz z resztą załogi Apollo 11 bezpiecznie powrócił na Ziemię. Szkoda, że prezydent John Fitzgerald Kennedy nie dożył chwili, gdy jego wizja zrealizowała się — wiemy, że w listopadzie 1963 roku zginął w Dallas od kuli zamachowca.

LATA SIEDEMDZIESIĄTE

ZMIANY ORGANIZACYJNE. UTWORZENIE INSTYTUTU INFORMATYKI

Początek lat siedemdziesiątych stoi pod znakiem zmian organizacyjnych w całym szkolnictwie wyższym. Wraz z wprowadzeniem w Uczelni struktury instytutowej Katedra Budowy Maszyn Matematycznych oraz Katedra Technologii Sprzętu Elektronicznego (prof. Stefan Okoniewski) zostają przekształcone w Instytut Budowy Maszyn Matematycznych. Potem, w roku 1975, nastąpi kolejna zmiana: prof. Okoniewski z zespołem swej dawnej Katedry przejdzie do Instytutu Technologii Elektronowej (obecny IMiO), natomiast Instytut Budowy

Maszyn Matematycznych zostanie przekształcony w Instytut Informatyki.⁷ Pod tą nazwą trwa do dziś. Nowy instytut przejmie także ośrodek obliczeniowy w Gmachu Głównym Politechniki (obecny COI: Centralny Ośrodek Informatyki). W toku tej reorganizacji z Instytutu odchodzą dwaj młodzi, samodzielni pracownicy: Konrad Fiałkowski i Jacek Bańkowski⁸, wraz z kilkoma współpracownikami i doktorantami. Utrzymują potem pewne kontakty naukowe z pracownikami Instytutu, są promotorami kilku rozpraw doktorskich (np. takich osób, jak: Jan Bielecki, Henryk Stelmasik, Włodzimierz Zuberek), zaś ich dwaj ówczesni asystenci: Henryk Rybiński i Mieczysław Muraszewicz, powrócą w latach pięćdziesiątych, sami już jako profesorowie, do Instytutu Informatyki.

Ważnym wydarzeniem było uruchomienie w 1975 roku studiów na kierunku Informatyka na Wydziale Elektroniki PW. Znaczną zasługę w organizacyjnym przeprowadzeniu tego złożonego przedsięwzięcia należy przypisać samemu Antoniemu Kilińskiemu, a także Janowi Zabrodzkiemu, który był autorem generalnego programu studiów dla tego kierunku. Niemniej, dla całego zespołu było to zadanie bardzo ambitne i trudne. Siłami jednego tylko instytutu (i to niezbyt dużego) o stosunkowo młodej i niezbyt „utyłowanej” kadrze trzeba było zapewnić wykłady, laboratoria, pomoce dydaktyczne itd. o tematyce obejmującej tak przecież nową i dynamicznie zmieniającą się dziedzinę. Na etapie organizowania nauczania na nowym kierunku ważną, porządkującą rolę odegrało powołanie w Instytucie (w 1976 r.) trzech zakładów dydaktycznych. Kierownictwo Zakładu Podstaw Informatyki objął Antoni Kiliński (do odejścia na emeryturę w 1978 r., potem Krzysztof Sapiecha). Zakład Budowy Sprzętu Informatyki poprowadził Stanisław Budkowski, zaś Zakład Organizacji i Oprogramowania — Jan Bielecki (do 1978 r., potem w latach 1978–80 Włodzimierz Zuberek i w 1981 r. — Jerzy Mieścicki).

Od strony liczby nauczycieli akademickich czy ilości zajęć dydaktycznych nowo powstały Instytut Informatyki jest wówczas jednostką o średnich rozmiarach, jednak wyróżnia się dużą liczbą pracowników inżynierskich i technicznych⁹ oraz obciążeniem zadaniami „poza-dydaktycznymi”. Zakład Doświadczalny Instytutu nadal prowadzi własną, małoseryjną produkcję stale udoskonalanych, specjalizowanych komputerów ANOPS i GEO. Nadal wspólnymi siłami zespołu nauczycieli akademickich i pracowników Zakładu Doświadczalnego realizowane są ambitne projekty informatyczne — będzie jeszcze o nich mowa poniżej. Jednocześnie, wraz z przejściem Ośrodka Obliczeniowego (z komputerem ODRA 1304) w Gmachu Głównym, na Instytut spadają obowiązki komputerowej obsługi „centrali” Uczelni: wprowadzanie oprogramowania do celów zarządzania Uczelnią (system kadrowy, finansowo-księgowy itd.), organizacja użytkowania uczelnianego terminalu systemu CYBER 73 (zdalna współpraca z dużą maszyną CDC 6400 zlokalizowaną w Instytucie Badań Jądrowych w Świerku pod Warszawą), doradztwo w sprawach polityki zakupów komputerów i organizacji ośrodków obliczeniowych, a także coroczna obsługa egzaminów wstępnych i procesu rekrutacji na studia. Znaczna część oprogramowania (dla komputera ODRA 1304) jest projektowana i eksploatowana siłami pracowników Instytutu, a system do obsługi rekrutacji na studia (Henryk Stelmasik, Włodzimierz Zuberek) wkrótce będzie wykorzystywany w innych polskich uczelniach (np. AGH w Krakowie).

TEMATYKA BADAŃ NAUKOWYCH W INSTYTUCIE

Znamienną ewolucję przechodzi tematyka prowadzonych w Instytucie badań. Nadal — jak w poprzednich latach — analizowane są niezawodnościowe aspekty systemów cyfrowych, jednak w ślad za tendencjami rozwojowymi w światowej informatyce w publikacjach i dok-

⁷ Warto dodać, że sam termin „informatyka” zaczął z wolna wchodzić w użycie w Polsce od 1968 roku z inicjatywy Romualda Marczyńskiego, który zgłosił taką propozycję na ogólnopolskiej konferencji w Zakopanem. Wcześniej mówiono raczej o „elektronicznej technice obliczeniowej”, „maszynach matematycznych”, „elektronicznych maszynach cyfrowych”, „cybernetyce technicznej” itp. Samo słowo „komputer” zadomowiło się zresztą na dobre w polskim języku jeszcze znacznie później. Trudno podać jakąś konkretną datę, lecz w każdym razie „minikomputer” (u nas — lata siedemdziesiąte) był wcześniejszy.

⁸ Jacek Bańkowski i Konrad Fiałkowski byli pierwszymi absolwentami specjalności „Maszyny Matematyczne”, którzy habilitowali się na Wydziale (1966), a potem — uzyskali tytuł profesora nadzwyczajnego (1973).

⁹ W 1976 r. Instytut Informatyki liczył łącznie 145 pracowników, w tym 30 nauczycieli akademickich.

toratach pracowników coraz wyraźniej pojawiają się nowe zagadnienia: teoria i metody projektowania systemów operacyjnych, modelowanie współbieżności i ocena wydajności systemów, a także projektowanie i weryfikacja układów mikroprogramowanych.

W tradycyjnej dla Instytutu tematyce niezawodnościowej ośrodek zainteresowań przesuwa się w kierunku metod testowania i diagnostyki układów, a także układów i systemów tolerujących uszkodzenia. Jan Zabrodzki uzyskuje (w 1978 r.) habilitację za rozprawę na temat projektowania układów elektronicznych z uwzględnieniem rozrzutu parametrów, a równolegle opracowuje metodykę testowania mikroprocesora 8080, przekazaną (w 1978 r.) do praktycznego zastosowania w produkcji układów scalonych w warszawskim CEMI. W tymże samym roku — 1978 — habilituje się Krzysztof Sapiecha (teorie i metody wyznaczania testów diagnostycznych dla układów asynchronicznych oraz projektowanie układów łatwo testowalnych). Stanisław Budkowski jest od 1975 r. regularnym członkiem komitetu programowego dorocznych międzynarodowych konferencji FTC (*Fault Tolerant Computing*) oraz FTSD (*Fault Tolerant Systems and Diagnostics*). Jest także — wspólnie z Przemysławem Prusinkiewiczem (doktorat w 1978 r.) — autorem kilku prac na temat kodów korekcyjnych. Zagadnień układów asynchronicznych o podwyższonej niezawodności dotyczy rozprawa doktorska Janusza Sosnowskiego (1976 r.), który będzie potem nadal pracował nad metodami testowania układów i systemami odpornymi na błędy, uzyska habilitację (1993 r.) i będzie kontynuował te badania do dnia dzisiejszego. Wszystko to sprawia, że tematyka niezawodności, diagnostyki i testowania, odporności na uszkodzenia itp. — a więc wiarygodności systemów — pozostaje niejako tradycyjną dziedziną badań w Instytucie.

Metodami opisu i projektowania systemów operacyjnych dla specjalizowanych maszyn cyfrowych zajęli się natomiast Wacław Iszkowski i Marek Maniecki. Rezultatami ich badań była nie tylko ich wspólna rozprawa doktorska (1977 r.), lecz również wykład i laboratorium wykorzystujące do celów dydaktycznych opracowane przez nich środowisko programowe i język JOS. Tematykę tę rozszerzą wkrótce o zagadnienia programowania współbieżnego, napiszą znany podręcznik (*Programowanie współbieżne*, 1981 r.) i będą kontynuowali te prace aż do odejścia z Uczelni na przełomie lat osiemdziesiątych i dziewięćdziesiątych.

Teoria i modele systemów współbieżnych stanowią w ogóle „gorący” temat w informatyce lat siedemdziesiątych. Szczególnego impulsu w ich rozwoju dostarczyło ponowne „odkrycie” w 1974 r. modelu formalnego, który już kilka lat wcześniej sformułował niemiecki teoretyk Carl Adam Petri. Chodziło o tzw. sieci Petriego, bliski intuicji i jednocześnie formalny model umożliwiający opisanie zachowań współbieżnych i koordynujących się procesów, konkurujących o wspólne zasoby. W Instytucie tematykę tę podjął Włodzimierz Zuberek, który w swej rozprawie doktorskiej (1978 r.) i w szeregu publikacji sformułował jeden z pierwszych, często potem cytowany w świecie, model *sieci Petriego z czasem*.

Problematyka ta wiąże się ściśle z *analizą wydajności*, to znaczy próbą formułowania odpowiedzi na *ilościowe* pytania dotyczące zachowania systemu: jaki jest czas odpowiedzi systemu, ile zadań jest on w stanie obsłużyć w jednostce czasu itd. Pytania tego typu zaczęto stawiać w informatyce już w latach sześćdziesiątych, wraz z pojawieniem się techniki podziału czasu (*time sharing*), wieloprogramowania (*multiprogramming*) i pamięci wirtualnej w dużych systemach komputerowych. W Instytucie Informatyki tematykę tę podjął w 1976 r. Jerzy Mieścicki ze współpracownikami, co w przeciągu kilku lat zaowocowało szeregiem doktoratów na temat analitycznych i symulacyjnych wydajnościowych modeli systemów współbieżnych (Andrzej Pająk, 1979 r.; Małgorzata Kalinowska-Iszkowska, 1980 r.; Anna Hać, Jacek Stochlak, Helena Szczerbicka, 1982 r.) i również znalazło kontynuację w latach osiemdziesiątych i później.

Nowe oblicze uzyskują także badania nad techniką mikroprogramowania, która była praktycznie stosowana już w komputerach EMC, UMC 1 i UMC 10. Wraz z pojawieniem się układów scalonych średniej skali integracji mikroprogramowanie zaczyna odgrywać rolę ważnego sposobu konstruowania układów sterowania, stanowiących poziom pośredni (*firmware*) pomiędzy sprzętem (*hardware*) a oprogramowaniem (*software*). Stanisław Budkowski wraz ze współpracownikami z Instytutu (Igor Hansen, Andrzej Papliński, Przemysław Prusinkiewicz) i z PAN (m.in. Jacek Blikle, Piotr Dembiński) podejmuje próbę zastosowania formalnych metod do specyfikacji i weryfikacji poprawności mikroprogramów. Zostaje opracowany język opisu mikroprogramów MIDDLE (lata 1977–78), powstają rozprawy doktorskie (Hansen, 1979 r.; Papliński, 1980 r.), oraz szereg wspólnych publikacji. Nieco później (lata 1981–82) powstaje

środowisko do symulacji i weryfikacji mikroprogramów zapisanych w MIDDLE. Niestety, wkrótce potem wszyscy członkowie tego zespołu rozjadą się po świecie, podobnie zresztą, jak wiele innych osób z Instytutu. Będzie o tym mowa dalej.

PROJEKT KRTM (UMC 20)

W 1972 r. Instytut rozpoczyna duży projekt, który na dobrych kilka lat zintegruje badania znacznej części zespołu. Chodzi o budowę własnego systemu minikomputerowego o roboczej nazwie KRTM 20, przewidzianego głównie do przygotowywania i wstępnej edycji danych w ośrodkach obliczeniowych. Wszystko wskazywało na to, że będzie to kolejny sukces Instytutu. Przewidywano, że system będzie produkowany w zakładach MERAMAT na warszawskim Służewcu.

Młodszym czytelnikom wypada uświadomić, że podstawowym nośnikiem danych i programów wprowadzanych do komputera były wówczas wciąż karty perforowane. Każdy ośrodek obliczeniowy zużywał ich całe tony. Programista (lub osoba przygotowująca dane do obliczeń) pisał swój program (lub dane) na odpowiednim formularzu i składał go w dziale dziurkowania kart, którego personel (posługując się klawiaturą dziurkarki kart) przenosił ciągi znaków z dokumentu źródłowego na owe karty dziurkowane. Operator dziurkarki nie widział zawartości całego wprowadzanego wiersza. O monitorach ekranowych czy choćby ciekłokrystalicznych znakowych wyświetlaczach w dziurkarkach kart nie było mowy: dane wprowadzało się praktycznie „na ślepo”, bez możliwości wykrycia i skorygowania choćby prostego manualnego błędu.

System KRTM 20 był pomyślany jako remedium na te kłopoty. Z informatycznego punktu widzenia miał to być uniwersalny system minikomputerowy (nazwany UMC 20), wykonany w technologii układów scalonych średniej skali integracji, przewidziany do jednoczesnej obsługi kilkunastu stanowisk z klawiaturami i monitorami ekranowymi, o współczesnej architekturze, z własnym, oryginalnym, wieloprotocowym systemem operacyjnym. Oprogramowanie aplikacyjne miało go jednak przystosowywać do roli wysoce ulepszanego systemu rejestrowania danych wprost na taśmie magnetycznej¹⁰, z pominięciem kart perforowanych. Operator terminalu widział wprowadzany tekst na znakowym monitorze ekranowym, system mógł kontrolować na bieżąco składniową poprawność tekstu, operator miał możliwość dokonywania poprawek i definiowania formatu pliku do zapisu na taśmie magnetycznej itd., co znacznie ułatwiało i przyspieszało współpracę z „właściwym” systemem obliczeniowym, a ponadto — powodowało wielką oszczędność papieru.

Dla zespołu Instytutu projekt UMC 20 (KRTM) był znakomitym, całościowym sprawdzianem profesjonalnej kompetencji i samodzielności. Rozpoczynał się od bardzo ogólnych założeń funkcjonalnych, a wymagał zaprojektowania i realizacji właściwie wszystkiego, co składało się na system komputerowy: architektury procesora, listy rozkazów, jednostki arytmetyczno-logicznej, układu sterowania, układu przerwań, jednostek sterujących urządzeń zewnętrznych itd., a także oprogramowania: od jądra systemu operacyjnego do systemu plików i sterowników urządzeń, od assemblera i innych elementów systemowego środowiska programistycznego do oprogramowania aplikacyjnego. W projekcie brało udział w sumie około 40 pracowników Instytutu. Całością prac kierował Jerzy Szewczyk. W rezultacie, w latach 1974–75, powstał bardzo udany, w pełni fizycznie zrealizowany prototyp sprzętu systemu (Andrzej Papliński, Marian Łakomy, Zbigniew Dudek i inni), a wkrótce potem uruchomiono system operacyjny MISS (Jan Bielecki, Aleksander Wigura i inni) oparty na pojęciu procesu, z konsekwentnymi mechanizmami zarządzania procesami i zasobami, a zatem swą nowoczesnością i możliwościami przypominający (nieprzypadkowo) znany od kilku lat UNIX, znacznie przewyższający dość prymitywne systemy operacyjne współczesnych mu minikomputerów. Zaprojektowano kilka wersji assemblera MAAS i inne elementy oprogramowania systemowego (konsolidator LINK, debugger, edytor itd.), a wreszcie program SFINX do kontroli poprawności wprowadzanych dokumentów źródłowych (Andrzej Pająk, Grzegorz Prochowski, Jan Sobolewski).

¹⁰ Stąd nazwa KRTM: Klawiaturowy Rejestrator na Taśmie Magnetycznej.

Młodsze pokolenie informatyków nie zdaje sobie chyba sprawy, jak koncepcyjnie i technicznie złożonym przedsięwzięciem była wówczas budowa „od zera” takiego systemu komputerowego. Gotowe, scalone mikroprocesory zaczęły się pojawiać dopiero kilka lat później; wtedy trzeba było zaprojektować i własnoręcznie wykonać z układów o małej i średniej skali integracji całe „logiczne wnętrze” procesora. Teraz, standardową klawiaturę można kupić za kilkadziesiąt złotych, wówczas — trzeba było zdecydować się na sposób kodowania znaków, zaprojektować i zamówić, a potem zmontować i plastikowe klawisze, i konstrukcję mechaniczną, i logiczne układy wchodzące w skład klawiatury. Nie było czegoś takiego, jak gotowy monitor ekranowy. W systemie KRTM rolę tę odgrywały odpowiednio przerobione radzieckie turystyczne telewizorki Junost’ 603. Był to jednak rarytas bardzo poszukiwany na rynku, więc zakup dziesięciu czy dwudziestu sztuk jednocześnie wymagał poważnej urzędowej korespondencji i zgody ze strony wysokich czynników. „Zorganizowanie” zachodnich dewiz na zakupy układów scalonych średniej skali integracji nie było łatwe; na szczęście Instytut sam zarabiał pewne kwoty dewiz na eksporcie urządzeń ANOPS. Ale nowsze typy układów, dobrze znane projektantom i potrzebne w projekcie, często niestety znajdowały się na tzw. „liście COCOM”¹¹, to znaczy były objęte strategicznym embargiem i zachodnim producentom nie wolno było ich sprzedawać do krajów obozu wschodniego. Na tej liście były na przykład również generatory znaków, bez których nie może się praktycznie obejść żaden monitor alfanumeryczny. Oczywiście, szybko udało się przywieźć z Zachodu „w kieszeni” kilka sztuk tych układów i wkrótce skonstruowano sterownik, dzięki któremu (w trybie podziału czasu i przy odpowiedniej synchronizacji) jeden generator znaków wyświetlał znaki alfanumeryczne na ośmiu (czy może nawet szesnastu) telewizorach Junost. Trochę to śmieszne, trochę straszne, ale takie były realia.

Ostatecznie, system KRTM 20 nie doczekał się jednak seryjnej produkcji. Przewidywany producent (MERAMAT) wycofał się ze wspólnych planów, gdyż otrzymał zgodę i fundusze na zakup licencji na zbliżony funkcjonalnie brytyjski system SEECHECK. Po odpowiedniej adaptacji do standardów RWPG, system ten był później produkowany jako MERA 9150.

Instytut Informatyki uratował natomiast projekt KRTM 20, wykorzystując dobre tradycje współpracy z polską geodezją. System minikomputerowy UMC 20 wyposażono w kompilator pełnego ANSI FORTRAN IV (Jan Bielecki, Aleksander Wigura, Marek Suchenek) oraz bibliotekę funkcji do obliczeń geodezyjnych. Następnie, na zlecenie Zjednoczenia Przedsiębiorstw Geodezyjnych i Kartograficznych (GEOKART), w Zakładzie Doświadczalnym Instytutu wykonano w sumie siedem egzemplarzy systemu, które (zainstalowane w latach 1979–83, tym razem pod nazwą GEO 20) w siedmiu (wszystkich siedmiu) okręgowych przedsiębiorstwach geodezyjnych i kartograficznych w Polsce stanowiły do końca lat osiemdziesiątych podstawowe komputery obliczeniowe dla tej dziedziny w naszym kraju.

INNE PROJEKTY BADAWCZO-ROZWOJOWE

Projekt UMC 20, choć największy, nie był jedynym przedsięwzięciem badawczo-rozwojowym tamtego okresu. Wytwarzano nadal specjalizowane komputery ANOPS i GEO (była o tym mowa wcześniej). Zespół Krzysztofa Sapiechy opracował ponadto i wykonał prototyp innego minikomputera (UMB — Uniwersalny Moduł Biomedyczny), przeznaczonego do rejestracji w czasie rzeczywistym i przetwarzania sygnałów bioelektrycznych z organizmu ludz-

¹¹ COCOM był to wspólny komitet powołany przez kraje NATO, mający na celu przeciwdziałanie sprzedaży do ZSRR i krajów socjalistycznych najnowszych technologii (w tym elektronicznych i informatycznych), mających znaczenie strategiczne. Na liście COCOM były w swoim czasie i komputery o pamięci RAM przekraczającej 256 kB (!) i — później — mysz komputerowa. Oczywiście, Obóz Wschodni odpowiedział na to gorączkowymi próbami zdobywania i kopiowania urządzeń i układów z listy COCOM. Odpowiedzią na strategiczne ograniczenia COCOM-u było również wprowadzenie tzw. „listy preferencyjnej”, to znaczy centralnego (w skali RWPG) wykazu układów scalonych i podzespołów, których wytwarzanie (oczywiście, z nieuchronnym kilkuletnim opóźnieniem) opanowano w krajach socjalistycznych. Tylko takich układów wolno było używać w urządzeniach elektronicznych produkowanych przemysłowo w krajach RWPG. Lista preferencyjna miała uniezależnić gospodarkę krajów RWPG i armię Układu Warszawskiego od Zachodu, ale jednocześnie stanowiła solidną, instytucjonalną gwarancję technologicznego zapóźnienia tych krajów.

kiego. UMB był wykorzystywany w badaniach nad medycznymi aspektami sportu wyczynowego w Akademii Wychowania Fizycznego w Warszawie. W 1979 r. podjęto konstrukcję nowej wersji tego specjalizowanego komputera (UMB 10). Bardzo interesująco zapowiadał się inny projekt, rozpoczęty w 1976 r. w Zakładzie Doświadczalnym, a mianowicie specjalizowane urządzenie WEGA, przeznaczone do rejestracji i przetwarzania sygnałów geofizycznych, powstających w wyniku odbić sygnału wymuszającego od wewnętrznych struktur geologicznych w toku poszukiwań złóż ropy i gazu ziemnego. Ukończony w 1979 r. prototyp (Grzegorz Malanowski) bardzo dobrze sprawdził się w próbach terenowych. Geologowie doskonalili algorytmy wnioskowania o prawdopodobnym położeniu złóż i weryfikowali je za pomocą tradycyjnych wierceń. Zgodność była bardzo dobra. Na podstawie tych doświadczeń, w latach 1981–84, skonstruowano nową wersję WEGA D-02 (Cezary Stępień, Jan Zabrodzki), która również była wykorzystywana w badaniach terenowych. Niestety, i to urządzenie poszło w niepamięć: podobnie jak wiele innych projektów pochłonął je kryzys ekonomiczny drugiej połowy lat osiemdziesiątych.

LATA 1978–1989

ZMIANY PERSONALNE

Jesienią 1978 roku, wraz z końcem kadencji, Antoni Kiliński przechodzi na emeryturę. Poprowadzi jeszcze kilka rozpoczętych pod jego opieką doktoratów, w kilku innych będzie recenzentem, ukończy i doprowadzi do ukazania się swej książki o pojęciu jakości. Wkrótce potem (w 1980 r.) odejdzie również z Instytutu dr Jerzy Szewczyk, Zastępca Dyrektora do spraw Zakładu Doświadczalnego, który był od 1962 r. „prawą ręką” Profesora. Doskonale się z nim rozumiał, w pełni identyfikował się z jego polityką integrowania działalności Instytutu wokół projektów realizowanych w Zakładzie i — jako dynamiczny i sprawny organizator — sam był w znacznej mierze współtwórcą sukcesu większości projektów badawczo-rozwojowych Instytutu. Odejście tych dwóch znaczących osób kończy pierwszy, pionierski okres w dziejach Instytutu Informatyki. Kierownictwo Instytutu obejmuje Jerzy Mieścicki (na czas kadencji 1978–81), potem przez sześć lat (1981–87) dyrektorem będzie Jan Zabrodzki, kierownikiem Ośrodka Obliczeniowego Henryk Stelmasik, a kierownikiem Zakładu Doświadczalnego Andrzej Skorupski.

TEMATYKA BADAWCZA INSTYTUTU

Na początku tego okresu wszystko wskazuje na to, że Instytut ma już dobrze sprecyzowaną tematykę badawczą, dydaktyczną i konstrukcyjną, wypracowaną w latach siedemdziesiątych. Obok zaawansowanych projektów o tradycyjnym dla Instytutu inżynierskim charakterze (ANOPS, GEO, WEGA, UMB, KARDIO itd.) rozpędu nabierają prace o tematyce bardziej teoretycznej: teoria i projektowanie systemów operacyjnych, modele współbieżności, weryfikacja mikroprogramów, diagnostyka i testowanie układów o podwyższonej niezawodności. Nie obywa się przy tym bez dyskusji, kontrowersji, a nawet napięć wokół generalnej polityki naukowej Instytutu, zwłaszcza dotyczącej proporcji między dominującymi dotąd projektami o charakterze praktycznym a badaniami prowadzonymi do wyników bardziej teoretycznych, ukierunkowanymi przede wszystkim na ich publikowanie.

Badania te zaczynają zresztą już owocować, co jest widoczne w postaci zwiększonej liczby doktoratów, podręczników i opracowań o tematyce „teoretyczno-systemowej” w końcu lat siedemdziesiątych i na początku osiemdziesiątych. Pisaliśmy o tym wcześniej. Dotyczy to również dziedziny projektowania sprzętu. Marian Łakomy i Jan Zabrodzki publikują podręczniki, używane potem przez wiele roczników studentów (*Cyfrowe układy scalone*, PWN, 1974 r.; *Liniowe układy cyfrowe w technice cyfrowej*, PWN, 1979 r.). Jednocześnie zespół pod kierownictwem Andrzeja Skorupskiego projektuje Modułowy System Mikroprocesorowy (MSM, lata 1979–81), który stanie się podstawą laboratorium projektowania systemów

mikrokomputerowych, a także wielu projektów wykonywanych w Zakładzie Doświadczalnym. MSM jest potem nieustannie unowocześniany, w ślad za rozwojem światowej techniki mikroprocesorowej. Na przykład, od 1983 r. są w nim uwzględniane mikroprocesory 16-bitowe i systemy operacyjne CP/M i ISIS, a w 1984 r. rusza sieciowe połączenie *on-line* między laboratorium MSM a komputerem ODRA 1304 w Ośrodku Obliczeniowym. Powstają także inne rozwiązania, np. Analizator Stanów Logicznych ASL-80 (Jan Zabrodzki, Witold Żaba), monitor szyny (wykonany potem w latach 1982–85 w ośmiu egzemplarzach dla różnych instytutów Wydziału) czy programator pamięci ROM (Marek Pawłowski, Andrzej Woźniak, 1983 r.), który — stale unowocześniany — będzie produkowany w Instytucie przez 10 lat na zamówienie wielu instytucji (m.in. CEMI, Zakłady w Błoniu itp.).

NIEPOKOJE, NADZIEJE I ZWĄPIENIA POCZĄTKU DEKADY

Ale czas niezbyt sprzyja spokojnej pracy naukowej i regularnemu pomnażaniu dorobku zawodowego. Przełom lat siedemdziesiątych i osiemdziesiątych, pierwsze lata osiemdziesiąte — to w Polsce czas szczególnie. W społeczeństwie wyczuwa się narastające niezadowolenie. Żyje się coraz trudniej. Cukier kupuje się „na kartki” już od 1976 roku, mięsa brakuje od dawna, teraz już zaczyna brakować benzyny, masła... Praktycznie wszystkiego. Najprostsze, codzienne zakupy trzeba opłacić wielogodzinnym oczekiwaniem w kolejkach. Telewizja i prasa promieniują urzędowym optymizmem, ale ludzie czują niepokój, napięcie, znużenie. Wreszcie, w lipcu i sierpniu 1980 r. frustracja i potrzeba zmian wybucha w postaci masowych strajków. Powstaje NSZZ Solidarność. Długo tłumiona aktywność ludzka ujawnia się w organizowaniu Związku, w dyskusjach, zebraniach, postulatach. Po kilkunastu miesiącach gorączkowej aktywności, huśtawki nerwów i nadziei — następuje niespodziewane wprowadzenie stanu wojennego (grudzień 1981 r.). Wojskowe patrole na ulicach pokrytego śniegiem miasta, głuchoe telefony, oba programy telewizyjne pełne nachalnej propagandy, godzina milicyjna, wieści o internowanych znajomych. I kolejki, kolejki, i kartki na wszystko. *Point de réveries, messieurs*; koniec marzeń, panowie. Zapowiada się długa, długa zima.

EXODUS

W takiej sytuacji, gdy tylko złagodzenie przepisów stanu wojennego przywraca możliwość podróżowania za granicę — wielu Polaków podejmuje decyzję o opuszczeniu kraju, w poszukiwaniu lepszych perspektyw zawodowego rozwoju, lepszego losu dla swoich rodzin. Dotyczy to oczywiście nie tylko informatyków, jednak specjaliści z tej dziedziny szczególnie łatwo znajdują miejsce za granicą. Dodatkowo, absolwenci informatyki kończący studia na naszym Wydziale i pracownicy Instytutu cieszą się opinią osób twórczych, dobrze przygotowanych zawodowo, bardzo samodzielnych i łatwo dostosowujących się do nowych wyzwań. To nie przypadek: ten, kto potrafił „od zera” zbudować procesor, fragment wieloprotocowego systemu operacyjnego, assembler czy kompilator Fortranu, nie przeleżnie się nowej wersji systemu, mikroprocesora czy nowego języka programowania, zwłaszcza, jeśli ma luksusowy (w stosunku do tego, do czego przywykł) dostęp do dokumentacji, komputera, dobrze wyposażonego laboratorium. Dlatego w pierwszej połowie lat osiemdziesiątych wiele osób odchodzi z zespołu Instytutu i rozprasza się po całym świecie. Co najmniej sześćdziesięciu kolegów ze stopniem doktora, którzy opuścili Polskę w tamtym właśnie okresie, jest obecnie profesorami wyższych uczelni na kilku kontynentach — od Australii po Hawaje. Ich odejście zubożyło potencjał intelektualny Instytutu. Decyzję o odejściu z Uczelni (choć niekoniecznie o wyjeździe z kraju) podejmują też inne osoby. Dodatkowo, nowa ustawa o szkolnictwie wyższym likwiduje (od 1982 r.) stanowisko tzw. docenta kontraktowego, które zajmował w Instytucie Jerzy Mieścicki. Dawało ono (choć na czas określony) wszystkie uprawnienia przysługujące „samodzielnemu pracownikom naukowym”, w tym — prawo kierowania pracami doktorskimi, których Jerzy Mieścicki zresztą wypromował sześć. W ten sposób, w omawianym krótkim okresie, z personelu Instytutu ubywa z różnych powodów łącznie trzech „samodzielnymi” pracowników (Kiliński, Budkowski, Mieścicki). Utrudnia to proces rozwoju kadry, tak pożądany w sytuacji, gdy jednocześnie ze stosunkowo nielicznego zespołu odchodzi ośmiu doktorów.

PRACE INSTYTUTU W POŁOWIE LAT OSIEMDZIESIĄTYCH

Mimo tych trudności Instytut kontynuuje swą misję. Zespół zgromadzony wokół Stanisława Budkowskiego wprawdzie rozpadł się, jednak badania nad formalną weryfikacją mikroprogramów (język i środowisko MIDDLE) toczą się nadal (Marek Gondzio, doktorat w 1989 r.). Jerzy Mieścicki, mimo uszczuplenia zespołu o troje świeżo wypromowanych doktorów, formułuje oryginalny model systemu cyfrowego: Sieci Sterowane Zdarzeniami, który będzie przedmiotem badań przez kilka lat, a następnie w drodze ewolucji przekształci się w formalny model systemu współbieżnego (CSM — *Concurrent State Machines*). Także Wacław Iszkowski, Marek Maniecki i Małgorzata Kalinowska-Iszkowska kontynuują pracę nad projektowaniem systemów operacyjnych. W 1986 r., nakładem PWN, ukazuje się ich książka pt. *Projektowanie systemów operacyjnych w ujęciu syntetycznym*. W tymże samym roku, również nakładem PWN, ukazuje się monografia Krzysztofa Sapiechy pt. *Testowanie i diagnostyka systemów cyfrowych*, która wkrótce otworzy autorowi drogę do tytułu profesorskiego. To oczywiście kontynuacja uprawianej od dawna „niezawodnościowej” tematyki Instytutu. Osiągnięcia na tym polu mają także inne osoby: Krzysztof Walczak przygotowuje rozprawę habilitacyjną, poświęconą hierarchicznemu wyznaczaniu testów, którą obroni w 1988 r. Znaczny dorobek naukowy gromadzi Janusz Sosnowski, również zmierzający do habilitacji.

Zespół zgrupowany wokół Jana Zabrodzkiego coraz głębiej angażuje się w problematykę zaawansowanej grafiki komputerowej i związanych z nią urządzeń, zwłaszcza do celów generowania obrazów w czasie rzeczywistym. Nabyta kompetencja i doświadczenia spowodują, że zespół Jana Zabrodzkiego rozpocznie wkrótce pracę nad projektem SLOT: wizualizacja w czasie rzeczywistym na potrzeby symulatora lotniczego, mającego służyć treningowi pilotów. Równolegle w tymże zespole kontynuowane są badania nad projektowaniem układów cyfrowych i systemów mikroprocesorowych. Marian Łakomy i Jan Zabrodzki wydają nowe podręczniki: *Scalone przetworniki analogowo-cyfrowe i cyfrowo-analogowe* (PWN, 1985 r.); *Układy scalone CMOS* (PWN, 1991 r.). Dwie osoby uzyskują stopień doktora (Cezary Stępień w 1983 r., Michał Rudowski w 1986 r.). Głównie dzięki zespołowi zgrupowanemu wokół J. Zabrodzkiego w roku akademickim 1986/87 zostaje uruchomione Studium Podyplomowe Systemów Mikroprocesorowych, którego kierownikiem jest Cezary Stępień.

Kontynuując wspomniane już poprzednio projekty badawczo-rozwojowe, Instytut podejmuje nowe tematy. Oto w latach 1982–83 powstaje projekt koncentratora danych KD 02 dla Kopalni Siarki w Tarnobrzegu (Janusz Sosnowski). Z wykorzystaniem tego urządzenia jest budowana sieć zbierania danych o przebiegu procesu technologicznego kopalni, rozłożonej na rozległym obszarze. To duży rozproszony system przemysłowy. Co więcej, urządzenia muszą pracować w trudnych warunkach atmosferycznych i w agresywnym chemicznie środowisku. Projekt zostaje wykonany z sukcesem. Do 1987 r. Instytut wytworzy 11 sztuk koncentratorów, dalsze 11 wykonano do 1989 roku. Ponadto opracowano oryginalny zestaw testerów przemysłowych.

Ośrodek Obliczeniowy w Gmachu Głównym PW, który wciąż jest częścią Instytutu Informatyki, doprowadza wreszcie (w 1987 r.) do całkowitego wyeliminowania technologii opartej na zastosowaniu kart dziurkowanych. Staje się to możliwe dzięki wprowadzeniu systemu przygotowania danych MERA 9150 — tego samego, który trzynaście lat wcześniej „zagroził drogę” oryginalnej konstrukcji Instytutu (KRTM 20) do seryjnej produkcji w warszawskich zakładach MERAMAT. Teraz przed Ośrodkiem staje nowe zadanie. Uczelnia otrzymuje (w 1986 r.) dar z Uniwersytetu w Darmstadt (RFN): komputer (*mainframe*) IBM S/370. Komputer jest wprawdzie nie najnowszy (z lat siedemdziesiątych), używany i dla ofiarodawców już przestarzały, lecz sprawny i funkcjonalnie znacznie przewyższający zarówno podstawowy komputer ośrodka, ODRA 1305, jak i inne eksploatowane w Instytucie maszyny (ODRA 1304 oraz klon minikomputera PDP 11, czyli SM-4, jeśli nie liczyć własnych konstrukcji). Zespół Instytutu ma zainstalować podarowany komputer, poznać nowy system i prowadzić jego eksploatację.

Równocześnie, od 1983 r., w Zakładzie Doświadczalnym prowadzone są prace nad nowym, oryginalnym minikomputerem GEO 3 (Zbigniew Dudek, Aleksander Wigura, Janusz Skolimowski). Ma to być mały, podręczny komputer obliczeniowy, wyposażony w specjalizowane urządzenia zewnętrzne do celów kartograficznych i we własny system programowania (SIGMAL), zbliżony do BASIC-u. Dobra renoma Instytutu w środowisku geodezji pozwa-

la przypuszczać, że przedsiębiorstwa geodezyjne i kartograficzne będą chętnie kupować to urządzenie. Mówi się, że małoseryjna produkcja powinna ruszyć za dwa, trzy lata.

Trudno było przewidzieć, że za owe dwa — trzy lata gotowy, bardzo funkcjonalnie zbliżony komputer (może jedynie bez specjalizowanych urządzeń) będzie można już sobie kupić prywatnie, do domu. Będzie się nazywał zapewne Amstrad, Amiga lub wręcz IBM PC.

NOWA REWOLUCJA W INFORMATYCE

Istotnie, za oceanem dojrzeła tymczasem nowa rewolucja informatyczna. W kalifornijskim garażu Steve Wozniak i Steve Jobs już skonstruowali swój prototyp komputera Apple. Technologicznie — niby nic nowego: znane mikroprocesory, układy LSI, języki programowania. My też je dobrze znamy, ale tam są one łatwo dostępne, nawet dla amatorów. Jednocześnie tworzy się nowy, oszałamiający potencjalną wielkością krąg użytkowników: prywatni, nieprofesjonalni nabywcy, którym trzeba dać prosty, obrazkowy, łatwy w użyciu sposób porozumiewania się z systemem — okna na ekranie, rozwijające się menu, mysz. W Wielkiej Brytanii sir Clive Sinclair już zbudował swój ZX Spectrum: ośmiobitowy mikroprocesor Z80, dziwaczna klawiatura, domowy telewizor jako monitor, magnetofon kasetowy jako pamięć zewnętrzna, wbudowany Basic, prymitywna grafika, pierwsze gry... Ale cena już dostosowana do kieszeni prywatnego użytkownika. Wkrótce ruszy lawina: Atari, Amstrad, Sharp, Commodore, Amiga...

Błękitny gigant, „Big Blue”, IBM, z początku pobłażliwie patrzy na harce tej drobnicy, gdy jednak orientuje się, że to wcale nie żarty — podejmie kontrofensywę i wkrótce wyjdzie na czoło. Jego „komputer osobisty”, IBM PC, przechodzi ewolucję od pierwszych XT i AT z połowy lat osiemdziesiątych do obecnych maszyn, z których każda ma szybkość *prawie* tysiąc razy większą, a pamięć operacyjną *ponad* tysiąc razy większą od pierwszych domowych mikrokomputerów. Już na samym początku IBM podejmuje znakomitą strategiczną decyzję: przemyka oko na fakt, że jego komputery osobiste są bezcennie kopiowane przez setki firm elektronicznych z całego świata. IBM decyduje tym samym, że nie musi zarabiać na sprzedaży domowych komputerów. W zamian jednak osiąga to, że powstaje ogromne, masowe zapotrzebowanie na *software* — na tym *dopiero* się zarabia. To zresztą dodatnie sprzężenie zwrotne: masowość produkcji sprzętu i podzespołów prowadzi do obniżenia ceny, zatem liczba użytkowników — a więc również zapotrzebowanie na oprogramowanie — wzrasta. Producent oprogramowania oferuje nowe wersje swych produktów, wymagające z zasady rozbudowy sprzętu: pamięci, pojemności dysku, szybszego procesora. Buduje to popyt na sprzęt, więc napędza masową produkcję podzespołów. Tak koło się zamyka i obraca coraz szybciej. Wyrastają na tym całe gałęzie gospodarki, nie tylko w USA i Japonii, lecz także na Tajwanie, w Korei, Hongkongu, Singapurze. Pewnego dość przeciętnego studenta, który niezbyt daleko w naukach się zapędził, produkcja *software’u* czyni w ciągu kilku lat najbogatszym człowiekiem świata. Kluczem do sukcesu staje się nie funkcjonalna doskonałość, lecz agresywny marketing i bezwzględna polityka rynkowa: odbiorcami są przecież nie profesjonalni informatycy, lecz prywatni ludzie, skłonni do zachwyty nad wszystkim co nowe i komputerowe. Drobniejsze płotki po prostu giną; także prekursor i główny konkurent na rynku komputerów osobistych, Apple, nie wytrzymuje trudów wyścigu i popada w kłopoty, trwające do dziś. Dzieje się tak, pomimo że Apple’a Macintosh wyprzedza o kilka lat inne komputery tego typu nowatorskimi pomysłami systemowymi, zaś o jego oprogramowaniu mówi się, że jest stabilniejsze, bardziej niezawodne i prawdziwie profesjonalnej jakości.

NARASTAJĄCY KRYZYS EKONOMICZNY I POLITYCZNY „OBOZU SOCJALISTYCZNEGO”

W naszej części politycznego świata mamy problemy inne, niż rewolucja informatyczna. W naszej podróży przez historię dotarliśmy do lat 1986–87: zjawiska kryzysowe, których objawy można było obserwować już od dawna, teraz nasilają się i nieuchronnie prowadzą do wniosku, że dalej tak już być nie może. Konieczna jest radykalna zmiana dotychczasowego modelu działania. Na najszerszej płaszczyźnie politycznej — prowadzi to Michaiła Gorba-

czowa (od 1985 r. Sekretarza Generalnego Komunistycznej Partii Związku Radzieckiego) do prób naprawiania ustroju komunistycznego. W Polsce stan wojenny odwołano już dawno i władza podtrzymuje pozory normalności. Jednak — w przeciwieństwie do Michaiła Gorbaczowa — między Bugiem a Odrą wszyscy zdają sobie sprawę, że „ta czaszka się już nie uśmiechnie”. Narasta wyraźny kryzys ekonomiczny, rusza inflacja, która wkrótce osiągnie rozmiary klęski. Dziś wiemy, do czego to w końcu doprowadzi: do prób uzyskania społecznego porozumienia wokół „odgórnice” wprowadzanych wątych zmian, do „okrągłego stołu”, wreszcie — do faktycznej zmiany ustroju, najpierw w Polsce, a potem jesienią 1989 r. — lawinowo — w innych krajach dotychczasowego obozu socjalistycznego.

TRUDNOŚCI DRUGIEJ POŁOWY LAT OSIEMDZIESIĄTYCH

Na razie jednak o „okrągłym stole” nikt nawet nie myśli. Coraz wyraźniej daje się za to we znaki kryzys ekonomiczny, a przede wszystkim inflacja. Oczywiście, jej skutki odczuwają wszyscy, zarówno w życiu prywatnym, jak i w działalności różnych instytucji. Instytut Informatyki jest nią jednak dotknięty chyba bardziej niż inne instytucje. Do tradycji Instytutu należały silne związki z praktyką. Oznaczało to realizację wielu interesujących projektów badawczych, wykonywanych na zamówienie zewnętrznych jednostek: służby zdrowia, geodezji, służb poszukiwań ropy i gazu itd. Z zasady prace były planowane na kilka lat — ponieważ zawierały znaczny element nowości i oryginalności, więc wymagały czasu na dopracowanie koncepcji, a następnie jej techniczną realizację. Tymczasem w warunkach inflacji nikt nie był w stanie przewidzieć, ile środków będzie miał na finansowanie takich badań za rok czy dwa. Zamawiającym instytucjom zaczyna brakować pieniędzy, strumień nowych zamówień zanika, z czasem zamawiający zaczynają wycofywać się nawet z już zawartych umów. Ten los spotka w ciągu kilku lat właściwie wszystkie poprzednio wymienione projekty, które były wykonywane na zamówienie zewnętrznych zleciennodawców.

FALA KOMPUTERÓW OSOBISTYCH

Równocześnie w Polsce już od kilku lat narasta fala zainteresowania komputerami osobistymi. Coraz więcej osób prywatnych przywozi sobie z zagranicy ZX Spectrum, Amigę czy Atari. Niektórzy robią sobie z tego źródło utrzymania. Rozwija się prywatny, z początku półlegalny import małych komputerów. Wkrótce powstają firmy handlujące komputerami, drukarkami, oprogramowaniem, gramami komputerowymi. Popyt na nie jest tak duży, że zostaje zniesione cło na sprzęt i oprogramowanie komputerowe. To decydujący moment: fala rośnie i zalewa całą Polskę taną, najczęściej dalekowschodnią elektroniką. A te małe komputery są coraz lepsze: po ZX Spectrum i Amstradach zaczynają się masowo pojawiać klony IBM PC, najpierw XT, potem AT, 286, 386, a do nich oprogramowanie, różnorodne karty, moduły pamięci. Zaczynają się w nie wyposażać także instytucje. Coraz mniej jest chętnych na oryginalne, unikalne rozwiązania. Cóż z tego, że urządzenie będzie profesjonalnie zaprojektowane i dobrze dostosowane do potrzeb użytkownika, skoro trzeba będzie na nie czekać rok, dwa, a może więcej? Czy wtedy jeszcze będziemy mieć na to środki? Poza tym, kusi niska cena. Urządzenia, choćby najlepsze, wykonywane w pojedynczych egzemplarzach lub co najwyżej małych seriach nie są w stanie wygrać konkurencji z dobrze zorganizowaną, przemysłową produkcją. Za kilka lat z tym problemem boleśnie zderzy się cały polski przemysł elektroniczny, na razie jeszcze chroniony przepisami celnymi.

NOWA KADENCJA 1987–1989

W październiku 1987 r., wraz z początkiem nowej kadencji władz Uczelni, stanowisko dyrektora Instytutu Informatyki obejmuje Krzysztof Sapiecha. Nowa dyrekcja przejmuje Instytut w niełatwej sytuacji. Zespół nauczycieli akademickich, uszczuplony wcześniej o kilka znaczących osób, nie szczędzi wysiłku, aby wykonywać swe podstawowe zadania dydaktyczne i naukowe. Wyposażenie laboratoriów staje się jednak coraz bardziej anachroniczne.

Dość powiedzieć, że w 1987 r., oprócz dotychczas eksploatowanych, starych maszyn ODRA 1305, 1304 oraz SM 4, Instytut posiada jedynie sześć komputerów PC XT. Wprowadzie w 1987 r. w Ośrodku Obliczeniowym rusza podarowany przez Uniwersytet z Darmstadtu *mainframe* IBM S/370, jednakże nie jest to komputer o szczególnych walorach dydaktycznych, a jego opanowanie do celów dydaktycznych, przygotowanie zajęć laboratoryjnych itd. wymaga ogromnego (i nie wiadomo, czy opłacalnego) wysiłku.

Informatycy wiedzą już także, że Ośrodek Obliczeniowy będzie musiał wkrótce podjąć jeszcze znacznie trudniejsze wyzwanie niż eksploatacja wystuzonego S/370. Na całym świecie w uczelniach komputery łączą się w sieci, najpierw lokalne, następnie w ramach wydziałów, następnie w skali całej Uczelni, wreszcie — w sieć o zasięgu światowym. To samo z pewnością czeka Politechnikę Warszawską, zwłaszcza że praktycznie wszystkie instytuty i wydziały zaczynają się wyposażać w sprzęt komputerowy i lokalne sieci już powstają. Osłabiony Instytut Informatyki nie podejmuje się zadania organizacji ogólnouczelnianej sieci. Dlatego w 1988 r. Ośrodek Obliczeniowy w Gmachu Głównym PW zostaje wyłączony ze struktury Instytutu Informatyki i podejmuje działalność jako samodzielna jednostka Uczelni: Centralny Ośrodek Obliczeniowy PW.

Jeszcze trudniejsza jest sytuacja instytutowego Zakładu Doświadczalnego. Do tej pory nie tylko zarabiał on na swoje utrzymanie, lecz stanowił dodatkowe źródło dochodów Instytutu, teraz zaczyna obciążać instytutowy budżet. Zamówienia wynikające z dawniejszych zobowiązań są już nieliczne: koncentratory KD 02 dla Siarkopolu, pewna liczba urządzeń ANOPS i KARDIO. Z nowych udaje się jedynie podjąć prace nad watomierzem całkująco-rejestrującym WATCAR (Zbigniew Dudek, 1988 r.) oraz konwerterem sygnalizacji Nr5/R2 (Zdzisław Braun, lata 1987–92). Były to jednak nieduże projekty, dające zatrudnienie właściwie pojedynczym osobom. W 1988 r. został jeszcze przekazany do produkcji w Zakładzie Doświadczalnym ostatni model urządzenia do przetwarzania sygnałów biomedycznych ANOPS-KARDIO 85 (Andrzej Skorupski, Ryszard Kott, Piotr Smażyński i inni). Popyt na te urządzenia dobiegał jednak kresu.

NOWE MOŻLIWOŚCI

Masowe pojawienie się komputerów w instytucjach, bankach, usługach, szkołach stworzyło informatykom zupełnie nowe możliwości zatrudnienia. To wprowadzie jeszcze nie lata dziewięćdziesiąte, kiedy prywatne firmy komputerowe będzie się liczyć na tysiące, niemniej już w latach 1987–88 zapotrzebowanie na informatyków jest duże. Oprócz firm i instytucji po prostu sprowadzających komputery i instalujących je u użytkownika, powstają pierwsze *software houses*: firmy projektujące oprogramowanie dla odbiorców w Polsce i za granicą, sprzedające oprogramowanie, dostosowujące zachodnie oprogramowanie do warunków polskich itd. Gwałtownie wzrasta także zapotrzebowanie na różne formy szkolenia i nauczania z zakresu elementarnej informatyki, na podręczniki o programowaniu, o bazach danych itd. Uczelnia, z jej anachronicznym wyposażeniem, trudnościami finansowymi, niskimi płacami i atmosferą frustracji, staje się mało atrakcyjnym miejscem pracy dla wielu młodych informatyków.

DRUGI EXODUS

I znowu wiele osób podejmuje decyzję o odejściu z Uczelni i rozpoczęciu nowego etapu życia zawodowego. Z Instytutu odchodzą kolejni doktorzy: Marcin Nowicki (w 1987 r.) i Marek Maniecki (w 1988 r.) do polsko-szwedzkiego *software house'u*, Henryk Stelmasik (w 1988 r.) do niemieckiej firmy MAKOMAT. Krzysztof Walczak uzyskuje w 1988 r. habilitację (*Hierarchiczne wyznaczanie testów dla układów cyfrowych*) i odchodzi, by poświęcić się nauczaniu informatyki w AWF i pisaniu podręczników z dziedziny programowania.¹² Nieco później z grona doktorów Instytutu Informatyki znikną: Wacław Iszkowski (od 1991 r. będzie wieloletnim prezesem Polskiej Izby Teleinformatyki) i Marek Gondzio (po doktoracie uzyskanym w roku 1989 przechodzi w 1991 roku do tej samej firmy co Marek Maniecki). Opuszczają

¹² Krzysztof Walczak powróci do Instytutu w 1999 r.

cych Instytut wartościowych pracowników, choć bez doktoratu, trudno by tu nawet wliczyć. Z grona zarówno nauczycieli akademickich, jak i pracowników Zakładu Doświadczalnego odchodzi w sumie kilkanaście osób. Odnajdują swe miejsce w bardzo różnych instytucjach i firmach, w kraju i za granicą: np. Aleksander Wigura w Kuwejcie, a Ryszard Kott w centrali *software’owego* giganta — Microsoftu — w Redmont, w USA.

Jak gdyby tego jeszcze było mało, na przełomie 1988 i 1989 roku dyrektor Instytutu, Krzysztof Sapiecha, decyduje się (z powodów osobistych) wyprowadzić z Warszawy do Kielc, gdzie obejmuje kierownictwo Katedry Informatyki w Politechnice Świętokrzyskiej. Kilka osób z jego zespołu odchodzi z nim, reszta jego współpracowników rozprasza się. Przez kilka miesięcy Instytut pozostaje pod rządami zastępców dyrektora: Jerzego Mieścickiego i Andrzeja Pająka.

To najtrudniejszy moment w całych dziejach Instytutu Informatyki. Jest oczywiste, że polityka naukowa Instytutu, rodzaj prowadzonych prac badawczo-rozwojowych, mechanizmy rozwoju naukowego pracujących w nim osób — muszą ulec radykalnej zmianie. Tradycyjny model działania — ukształtowany jeszcze w latach sześćdziesiątych i siedemdziesiątych, będący wtedy źródłem sukcesów Instytutu — nie wytrzymuje konfrontacji ani z nową sytuacją w światowej informatyce i z zachodnimi produktami komputerowymi, ani z wyjątkowo trudną sytuacją ekonomiczną zarówno Uczelni, jak i całego kraju. Jednocześnie, zespół jest bardzo osłabiony. W Instytucie pozostało jedynie dwóch „samodzielnych” pracowników: profesor Jan Zabrodzki i docent Kazimierz Bierkowski. Grono adiunktów i asystentów jest również bardzo zdekompletowane. Poczucie niepowodzenia, powtarzające się wiadomości o kolejnych kolegach decydujących się opuścić Instytut — pozbawiają wielu spośród tych, którzy pozostali, tej odrobiny optymizmu, który jest niezbędny do przetrwania trudności.

Po kilku miesiącach poszukiwań i konsultacji pojawia się nowa szansa: stanowisko Dyrektora Instytutu Informatyki zgadza się objąć Zdzisław Pawlak.

LATA DZIEWIĘDZIESIĄTE: ODBUDOWA I BUDOWA OD NOWA

KADENCJA ZDZISŁAWA PAWLAKA

Zdzisław Pawlak obejmuje formalnie stanowisko Dyrektora Instytutu z początkiem roku akademickiego 1989/90. Tak, to ten sam Zdzisław Pawlak, który w latach pięćdziesiątych jako konstruktor w Katedrze prof. Kilińskiego projektował UMC 1, a wcześniej — jeszcze w ramach grupy GAM — pierwszy „aparat matematyczny” GAM 1. Teraz jest profesorem zwyczajnym, członkiem PAN, uczynionym o ogromnym dorobku i o niekwestionowanym autorytecie w środowisku informatyków nie tylko w Polsce, ale również poza jej granicami. W chwili obejmowania stanowiska Dyrektora Instytutu Informatyki Zdzisław Pawlak zajmuje się zagadnieniem wnioskowania z niepewnych lub niedokładnych danych. Jest to tematyka zaliczana do kręgu sztucznej inteligencji. Do najnowszych osiągnięć Zdzisława Pawlaka w tej dziedzinie należy teoria zbiorów przybliżonych (*rough sets*) oraz liczne prace (których był inicjatorem i współautorem) zmierzające do praktycznego wykorzystania tej teorii, m.in. w diagnostyce medycznej, w układach sterowania, systemach wspomagania decyzji itd. Tak więc, oprócz swej wiedzy, pozycji i autorytetu Zdzisław Pawlak wnosi do Instytutu nową, aktualną tematykę, której we współczesnej informatyce poświęca się wiele uwagi. Zdzisław Pawlak będzie Dyrektorem Instytutu aż do 1996 r., to znaczy do chwili, gdy wraz z końcem kolejnej kadencji władz Uczelni przejdzie na emeryturę.

Nowe kierownictwo Instytutu rozpoczyna od przemyślenia od nowa i zrewidowania naukowego profilu i organizacyjnej struktury Instytutu. Muszą one — z jednej strony — odpowiadać współczesnym kierunkom badań w światowej informatyce, z drugiej — wykorzystywać dotychczasowe atuty i mocne strony zespołu. Drugim bardzo pilnym zadaniem jest unowocześnienie i rozbudowa zasobów komputerowych, wykorzystywanych zarówno w ba-

daniach, jak i w zajęciach dydaktycznych. Trzecim — rozwiązanie problemów kadrowych, zarówno tych, które są związane z uszczupleniem zespołu nauczycieli akademickich (zwłaszcza z małą liczbą profesorów i docentów), jak i z zatrudnieniem pracowników Zakładu Doświadczalnego, który coraz trudniej jest utrzymać z realizowanych w nim projektów.

Już jesienią 1989 r. w Instytucie zostaje utworzony jeden zakład — Zakład Grafiki Komputerowej (kierowany przez Jana Zabrodzkiego), oraz sześć pracowni, odpowiadających głównym kierunkom specjalizacji naukowej Instytutu. Zespół nauczycieli akademickich dzieli się pomiędzy wymienione jednostki. Oprócz nich, w Instytucie działa Laboratorium Komputerowe (kierowane przez Marka Pawłowskiego, mające pod opieką kilka pracowni komputerowych wykorzystywanych głównie do celów dydaktycznych), Zakład Doświadczalny oraz — co oczywiste — grupa pracowników administracyjnych.

Powołane pracownice wypracowują sobie tematykę nowych projektów, podejmują udane starania o granty (zarówno rektorskie i dziekańskie, jak i finansowane przez nowo powstały Komitet Badań Naukowych), kompletują wyposażenie własnych laboratoriów badawczych. Tematyka ich badań w naturalny sposób ewoluuje, towarzyszą temu zmiany sytuacji personalnej. Janusz Sosnowski uzyskuje w 1993 r. habilitację (*Analiza i testowanie błędów przemijających w systemach cyfrowych*), a w 1994 r. — Nagrodę Wydziału IV Nauk Technicznych PAN za całokształt działalności naukowej. Henryk Rybiński przenosi się do Instytutu na stałe, wkrótce współpracę z jego zespołem podejmie (początkowo na pół etatu) Mieczysław Muraszkiewicz. Kilka osób uzyskuje stopień doktora, z drugiej strony kilku doktorów odchodzi z Instytutu. W 1994 r. Instytut spotyka dotkliwą stratę: umiera nagle Kazimierz Bienkowski, pierwszy doktor w historii Instytutu i jeden z jego najstarszych pracowników.

W 1994 r. sytuacja dojrzeje do kolejnej zmiany struktury organizacyjnej Instytutu. Zakład Grafiki Komputerowej będzie kontynuował działalność pod kierownictwem Jana Zabrodzkiego, lecz w miejsce sześciu pracowni zostaną utworzone dwa dalsze zakłady: Zakład Systemów Informacyjnych, którego kierownikiem będzie Henryk Rybiński, oraz Zakład Oprogramowania i Architektury Komputerów, kierowany przez Janusza Sosnowskiego.

Od samego początku kadencji Zdzisława Pawlaka Instytut podejmuje także starania o nowe zamówienia na projekty i prace techniczne, które pozwoliłyby utrzymać zespół Zakładu Doświadczalnego. Niestety, czas nie sprzyja działalności tego Zakładu. Początek lat dziewięćdziesiątych — to przecież okres bardzo burzliwych zmian ustrojowych w Polsce. W całej polskiej gospodarce, a zwłaszcza w przemyśle elektronicznym, następuje gwałtowna zmiana: dawne — „socjalistyczne” — maszyny, urządzenia i technologie okazują się w bolesny sposób przestarzałe. To samo dotyczy form zarządzania. Na ich miejsce wchodzi rozwiązania importowane, zachodnie lub azjatyckie. Pojawiają się przedstawicielstwa wielkich i małych światowych firm wytwarzających komputery, oprogramowanie i gotowe systemy do celów zarządzania, sterowania procesami, obsługi urządzeń medycznych, obliczeń inżynierskich... Doświadczenie i fachowość zespołu Zakładu Doświadczalnego nie wystarcza do podjęcia rywalizacji z nimi, zwłaszcza że jego techniczne wyposażenie pochodzi w znacznej mierze również z ubiegłej epoki. W rezultacie, po ponad półtorarocznych bezowocnych staraniach o nowe zamówienia, w 1991 r. Zakład Doświadczalny Budowy Maszyn Matematycznych¹³ zostaje zlikwidowany. To trudny, gorzki moment rozstania z ważną częścią własnej historii. Wielu młodszych pracowników Zakładu odeszło już wcześniej z własnej woli, teraz na wcześniejsze emerytury odchodzi duża grupa doświadczonych inżynierów, techników i robotników, którzy nierzadko przez trzydzieści i więcej lat tworzyli komputery i inne urządzenia, stanowiące kiedyś o sukcesach Instytutu.

W 1989 r. Instytut podejmuje (wspólnie z Instytutem Telekomunikacji) organizację podyplomowego studium teleinformatyki i zarządzania CITCOM. Jest to inicjatywa francuskiego partnera, Institut Telesystemes, który ma doświadczenie w prowadzeniu tego typu szkoleń. Oprócz korzyści merytorycznych (wzbogacenie oferty Wydziału o nowoczesną, bardzo aktualną tematykę i przeszkolenie we Francji grupy pracowników obu instytutów) laboratorium komputerowe Instytutu otrzymuje darowany przez stronę francuską system komputerowy DPX 2/210 (produkcji firmy Bull), z systemem operacyjnym zbliżonym do UNIX-a i systemem

¹³ Tę wzruszająco niedzisiejszą nazwę Zakład Doświadczalny zachował do końca.

zarządzania bazą danych ORACLE, wyposażony w osiem terminali. Będzie on wykorzystywany zarówno w ramach Studium CITCOM, jak i w innej działalności Wydziału.

System DPX (Bull) jest bardzo potrzebny Instytutowi. Odłączenie Centralnego Ośrodka Informatyki znacznie uszczupliło wyposażenie instytutowego Laboratorium Komputerowego: dość wspomnieć, że w 1989 r. Instytut dysponował jedynie czternastoma komputerami PC AT. Jednakże, już w 1991 r. Instytut pozyskuje — w dużej mierze dzięki aktywności Zdzisława Pawlaka — nowoczesny komputer IBM RS/6000 (RISC) oraz pierwsze cztery stacje robocze Sun SparcStation. Potem, w 1994 r., przybędą jeszcze 32 stacje robocze Sun, które zasilą laboratorium obsługujące potrzeby dydaktyczne całego Wydziału. Stale zwiększa się liczba komputerów typu PC. W 1991 r. łączone są one w sieć Novell, a w 1992 r. powstaje ogólnoinstytutowa sieć LAN. Środki na ten cel pochodzą z wszelkich możliwych źródeł: zasobów instytutowych, dziekańskich, Komitetu Badań Naukowych, a także projektów badawczych wykonywanych przez Zakład Grafiki Komputerowej i sześć pracowni naukowych. Później, po uruchomieniu w 1995 r. Wieczorowych Studiów Zawodowych — także ze środków pochodzących z opłat za studia.

W sumie, należy stwierdzić, że w ciągu lat dziewięćdziesiątych Instytut włożył wiele środków, pomysłowości i wysiłku organizacyjnego w rozbudowę i nieustanną modernizację swej bazy komputerowej. Należy podkreślić, że oprócz zasobów zlokalizowanych w instytutowym Laboratorium Komputerowym, poszczególne jednostki wyposażały się w bardziej specjalistyczny sprzęt i oprogramowanie. Stanowią one obecnie podstawę laboratoriów zakładowych, będących miejscem badań i zaawansowanej dydaktyki (pracownie problemowe, prace dyplomowe itd.). Dzięki temu wysiłkowi osiągnięto także znaczną różnorodność dostępnych dla badań i dydaktyki platform sprzętowych, systemów operacyjnych, narzędzi do tworzenia oprogramowania itd.

ROZWÓJ I DZIAŁALNOŚĆ ZAKŁADÓW INSTYTUTU

Jak wspomniano, od 1994 roku działają w Instytucie Informatyki trzy zakłady. Stanowią one do dnia dzisiejszego podstawowe jednostki organizacyjne, na których opiera się zarówno dydaktyczna, jak i badawcza działalność Instytutu. Każdy z zakładów ma także własne laboratorium, uzupełniające instytutowe zasoby komputerowe o różnorodny, specjalistyczny sprzęt i oprogramowanie.¹⁴

■ **Zakład Grafiki Komputerowej** (kierowany przez Jana Zabrodzkiego) został utworzony już w 1989 r.¹⁵ Od początku podstawowa tematyka badań tego zespołu obejmowała specjalizowane architektury komputerowe do celów grafiki komputerowej, modelowanie i wizualizację zjawisk naturalnych (zwłaszcza w czasie rzeczywistym), a także specjalistyczne metody i algorytmy grafiki komputerowej. Zakład kilkakrotnie uzyskuje środki (granty KBN) na realizację dużych projektów badawczych, integrujących działania zespołu. W drugiej połowie lat dziewięćdziesiątych tematyka ta rozszerza się o metody łączenia obrazów generowanych komputerowo z obrazem z kamery. W 1998 r. Przemysław Rokita uzyskuje z KBN grant na badania nad tym zagadnieniem. W wyniku badań przedstawia rozprawę habilitacyjną i w roku 2000 uzyskuje habilitację. W ostatnim okresie Zakład zajmuje się również techniką wirtualnej rzeczywistości. Jednocześnie kilka osób kontynuuje prace związane z dawną specjalnością Instytutu: przetwarzanie sygnałów biomedycznych.

Zespół Zakładu, w znacznej mierze dzięki aktywności kierownika, wyrobił sobie pozycję znanego polskiego ośrodka badań nad grafiką komputerową. Przyczyniła się z pewnością

¹⁴ Szczegółowe informacje o organizacji Instytutu, aktualnej tematyce naukowej, prowadzonych projektach, wyposażeniu laboratoriów itd. można znaleźć na instytutowej stronie www.ii.pw.edu.pl.

¹⁵ Warto tu dodać, że pierwsze prace związane z grafiką komputerową miały miejsce już na przełomie lat siedemdziesiątych i osiemdziesiątych. W 1982 r. ukazała się pierwsza polska praca poświęcona grafice komputerowej. Był to skrypt pt. *Wybrane zagadnienia cyfrowej techniki graficznej* napisany przez Przemysława Prusinkiewicza i Cezarego Sępnia. W późniejszych latach Przemysław Prusinkiewicz, pracując już w Kanadzie, stał się jednym z czołowych naukowców w skali światowej, pracujących w obszarze grafiki komputerowej.

do tego zbiorowa praca pt. *Grafika komputerowa, metody i narzędzia* (pod redakcją Jana Zambrodzkiego, WNT, 1994), a także fakt, że od 1993 roku Zakład prowadzi wspólnie z Komitetem Informatyki PAN ogólnopolskie seminarium *Grafika komputerowa i przetwarzanie obrazów*.

Wśród specjalistycznego wyposażenia Laboratorium Zakładu Grafiki Komputerowej na szczególną uwagę zasługują stacje graficzne firmy Silicon Graphics, sprzęt do tworzenia wirtualnej rzeczywistości oraz bogate specjalistyczne oprogramowanie do modelowania graficznego.

■ **Zakład Systemów Informatycznych**, którym od początku kieruje Henryk Rybiński, koncentruje się na badaniach z zakresu baz danych, baz wiedzy, systemów ekspertowych oraz teoretycznych i praktycznych aspektach wnioskowania na podstawie niepewnych danych. Ostatnie z wymienionych zagadnień jest przedmiotem naukowych zainteresowań Zdzisława Pawlaka. W 1991 r. Pawlak publikuje monografię pt. *Rough Sets. Theoretical Aspects of Reasoning about Data* (Kluwer, 1991 r.). Do badań nad teorią zbiorów przybliżonych Pawlak organizuje zespół badawczy, w którego pracach uczestniczą osoby nie tylko z Instytutu Informatyki, lecz z innych instytutów Wydziału, z innych polskich uczelni i ośrodków zagranicznych. Rezultaty badań są szeroko publikowane i wzbudzają ogromne zainteresowanie środowisk zajmujących się sztuczną inteligencją. Zdzisław Pawlak uczestniczy w licznych krajowych i zagranicznych konferencjach naukowych poświęconych tej tematyce; wreszcie w 1998 roku (wspólnie z Instytutem Matematyki Uniwersytetu Warszawskiego) zorganizuje na terenie naszego Wydziału wielką międzynarodową konferencję: *First International Conference on Rough Sets and Current Trends in Computing (RSCTC'98)*. Mimo odejścia (w 1996 r.) na emeryturę Zdzisław Pawlak utrzymuje do dziś niezwykłą aktywność naukową, a jego opublikowany dorobek naukowy z ostatnich lat budzi prawdziwy podziw.

Badania dotyczące baz danych, baz wiedzy i systemów ekspertowych dotyczyły głównie problemów synchronizacji transakcji w bazach danych, semantycznych zagadnień reprezentacji wiedzy w systemach informacyjnych oraz multimedialnych baz danych. W końcu lat dziewięćdziesiątych tematyka prac zespołu wzmocnionego przez Mieczysława Muraszkiewicza, ewoluowała w kierunku metod pozyskiwania wiedzy (*knowledge discovery*) i eksploatacji danych (*data mining*). W 1999 roku zespół kierowany przez Mieczysława Muraszkiewicza rozpoczął duży projekt badawczy, realizowany na zlecenie i we współpracy z Polską Telefonią Cyfrową, operatorem sieci telefonii komórkowej ERA GSM, dotyczący wspierania procesów zarządzania w ERA GSM za pomocą technik z kręgu *data mining*. Projekt jest wieloetapowy; wykonane do dziś etapy dostarczyły obu stronom ciekawych doświadczeń i wniosków. Wszystko wskazuje na to, że projekt będzie kontynuowany, stanowiąc źródło inspiracji i doświadczeń dla znacznej części Zakładu.

■ Tematyka badawcza trzeciego z zakładów, mianowicie **Zakładu Oprogramowania i Architektury Komputerów**¹⁶, obejmuje zagadnienia diagnostyki i testowania systemów komputerowych, modelowanie zachowań systemów współbieżnych i rozproszonych oraz inżynierię oprogramowania, ze szczególnym uwzględnieniem problematyki projektowania systemów działających w czasie rzeczywistym.

Badania nad testowaniem systemów, odpornością na uszkodzenia i błędy przemijające są domeną Janusza Sosnowskiego i jego współpracowników. Grupa ta ma w dorobku oryginalne metody testowania systemów mikroprocesorowych i analizy odporności na uszkodzenia (*fault tolerance*) systemów wielokomputerowych. W ostatnim czasie tematyka ta rozszerzyła się o metody zaawansowanej syntezy układów logicznych (Paweł Kerntopf). Rezultaty są dobrze znane i wysoko oceniane w międzynarodowym środowisku naukowym. W znacznej mierze dzięki temu, we wrześniu 2001 r., na wydziale odbyła się znana międzynarodowa konferencja EUROMICRO 2001, której organizatorami byli Janusz Sosnowski i Tadeusz Łuba (z Instytutu Telekomunikacji).

¹⁶ Zakładem kieruje Janusz Sosnowski, który jest także od 1996 r. do chwili obecnej Dyrektorem Instytutu Informatyki.

Zespół kierowany przez Jerzego Mieścickiego kontynuuje badania nad modelami systemów współbieżnych i rozproszonych. W początku lat dziewięćdziesiątych sformułowany został oryginalny model teoretyczny (CSM — *Concurrent State Machines*), który po kilku latach rozwoju stał się podstawą dużego projektu COSMA, zakładającego wykorzystanie formalnych metod weryfikacji poprawności w procesie projektowania zarówno współbieżnego obiektowego oprogramowania, jak i systemów wbudowanych (*embedded systems*). Tematyka ta należy do najbardziej aktualnych wyzwań we współczesnej informatyce, wiąże się z rozwojem metod inżynierii oprogramowania i narzędzi programowych wspomagających proces projektowania (CASE — *Computer-Aided Software Engineering*).

Należy podkreślić, że zespół Zakładu stara się o praktyczne wykorzystanie swych umiejętności w zaawansowanych projektach wykonywanych na zlecenie zewnętrznych instytucji. W szczególności, w latach 1994–95, we współpracy z Instytutem Techniki Ciepłej PW (zespół Janusza Lewandowskiego) grupa pracowników Zakładu kierowana przez Wiktora Daszczyka zaprojektowała i wdrożyła złożony system monitorowania i nadzoru bloków energetycznych dla warszawskiej Elektrociepłowni Siekierki. W 1996 roku Zespół uzyskał pierwsze miejsce w konkursie o Nagrodę Siemens w kategorii projektów badawczych mających zastosowanie w praktyce. Od tego czasu system był stale modyfikowany, ulepszany i rozwijany, a w chwili obecnej pracuje w ruchu ciągłym już w kilku polskich elektrowniach (EC Siekierki i Żerań, elektrownie w Połańcu, Żarnowcu i Kozienicach). Drugim przykładem projektów wykonywanych dla zewnętrznych zleceniodawców są prace wykonywane w latach 1995–98 dla wielkiego niemieckiego operatora telefonii komórkowej T-Mobil GmbH (części Deutsche Telekom). Obejmowały one najpierw organizację eksperymentalnej sieci telefonii cyfrowej GSM w regionie suwalskim, a następnie projekt organizacji rozproszonego systemu komputerowego, wykorzystującego złożony protokół komunikacyjny zbudowany na usłudze SMS (Wiktor Daszczyk, Jerzy Mieścicki).

IMPONUJĄCA DZIAŁALNOŚĆ JANA BIELECKIEGO

Na oddzielną wzmiankę zasługuje niezwykle dorobek wydawniczy jednego z długoletnich pracowników Instytutu Informatyki, mianowicie Jana Bieleckiego. Bielecki należy do najbardziej doświadczonych polskich programistów, ma w swym dorobku znaczące profesjonalne osiągnięcia, jak choćby — u nas — współautorstwo systemu operacyjnego MISS dla komputera UMC 20 czy liczne publikacje na temat różnych implementacji języka C, które zapewniły mu pozycję eksperta międzynarodowego komitetu dla opracowania standardu tego języka. Jest jednak najbardziej znany jako autor niezwykle liczby podręczników i monografii z dziedziny technik, systemów i języków programowania. Dziedzina ta rozwijała się bardzo szybko, lecz Bielecki śledził jej rozwój, trafnie oceniał wagę nowych rozwiązań i niezwykle sprawnie dostarczał czytelnikom kompetentne podręczniki, w których rozważania ilustrował sporządzonymi i zanalizowanymi przez siebie przykładami. Już w 1988 r. miał w swym dorobku 32 pozycje książkowe o tym charakterze. Od tamtego czasu, wraz z rozpowszechnianiem się komputerów, popyt na podręczniki z zakresu programowania znacznie się zwiększył, zwiększyła się też autorska produktywność Jana Bieleckiego. Dość powiedzieć, że dziś jego dorobek obejmuje 113 książek (słownie: sto trzynaście!), wydanych w łącznym nakładzie przekraczającym 800 000 egzemplarzy. Są one znane w Polsce, były tłumaczone i wydawane za granicą, przyniosły autorowi niezliczone nagrody Ministra i Rektora, a w 2000 roku — również prestiżową (w środowisku polskich informatyków) nagrodę „Krzemowego Oskara”, którą przyznano mu w wyniku tajnego głosowania internetowego w ramach wystawy „Infosystem 2000” odbywającej się w Poznaniu.

MEDAL „COMPUTER PIONEER” DLA ANTONIEGO KILIŃSKIEGO

W 1997 roku miało miejsce wydarzenie o charakterze symbolu, zamykającego klamrą całą kilkudziesięcioletnie dzieje polskiej informatyki. Oto IEEE Computer Society, największe międzynarodowe stowarzyszenie informatyków, przyznało honorowy medal Pioniera Informatyki (*Computer Pioneer*) dwóm uczonym z Polski: Antoniemu Kilińskiemu i Romualdowi Mar-

czyńskiemu¹⁷. Obaj Profesorowie znaleźli się w ten sposób w gronie kilkudziesięciu postaci dla naszej dziedziny historycznych, obok konstruktorów ENIAC-a i twórców pierwszych języków programowania, a na awersie przyznanego im medalu widnieje podobizna Charlesa Babbage'a, konstruktora pierwszej programowanej mechanicznej maszyny liczącej sprzed przeszło stu sześćdziesięciu lat.

W uzasadnieniu decyzji o przyznaniu medalu Antoniemu Kilińskiemu przypomniano nie tylko jego rolę w zbudowaniu pierwszego komputera produkowanego przemysłowo w Polsce, lecz również jego zasługi dla kształcenia polskich informatyków. Instytut Informatyki brał udział w obu wspomnianych dziełach swego założyciela i wieloletniego kierownika, mamy więc prawo uważać, że jakaś część tego zaszczytu przypada wszystkim jego współpracownikom, uczniom i następcom, którzy przez pięćdziesiąt lat konstruowali komputery i oprogramowanie, prowadzili badania w dziedzinie informatyki i przekazywali swą wiedzę studentom Wydziału Elektroniki.

Przyznanie Antoniemu Kilińskiemu i Romualdowi Marczyńskiemu medalu Pioniera Informatyki stanowi nie tylko wyraz uznania dla ich osobistych zasług. To także gest przyjęcia polskiego środowiska informatycznego do światowego bractwa informatyków. Całe lata istnieliśmy i działaliśmy jak gdyby równoległe, oddzieleni odmiennością systemów politycznych i rywalizacją między mocarstwami. Dzieliły nas możliwości technologiczne, wyposażenie naszych laboratoriów, trudności w wymianie doświadczeń... Ale naszą pomysłowość, czas, wysiłek intelektualny poświęciliśmy wspólnej sprawie. Zaliczenie dwóch polskich uczonych do grona pionierów światowej informatyki można odczytywać jako symboliczny gest zamykający epokę podziałów.

To zobowiązuje: nie żyjemy już na dalekich peryferiach rozwijającego się świata i mamy obowiązek wnieść swój wkład także w dzisiejszy i jutrzejszy rozwój naszej dziedziny: informatyki.

PRÓBA SPOJRZENIA W PRZYSZŁOŚĆ

Staraliśmy się pokazać w subiektywnym skrócie pięćdziesięcioletnią historię badań, prac konstrukcyjnych i nauczania w dziedzinie informatyki na naszym Wydziale, a przede wszystkim — losy Instytutu Informatyki i tworzących go ludzi. To pouczające doświadczenie: uświadomić sobie, jak dramatyczne zmiany były udziałem tej dyscypliny na przestrzeni zaledwie dwóch pokoleń i — jednocześnie — jak jej rozwój na świecie i u nas był powiązany z globalnymi procesami historycznymi i cywilizacyjnymi, z wielkimi konfliktami i osiągnięciami ostatniego półwiecza.

Te związki trwają nadal. Wszyscy powtarzają, że ludzkość przechodzi właśnie od ery społeczeństwa przemysłowego do informacyjnego. Powszechny i prawie nieskrępowany dostęp do informacji, wszechobecność Internetu, nowe formy komunikowania i samoorganizowania się grup społecznych i całych społeczeństw, nowe formy przedsiębiorczości... Dla jednych — to wizja nowego, wspaniałego świata, dla innych — przede wszystkim zagrożenia: upadek tradycyjnych instytucji i autorytetów, zniszczenie odmienności kulturowych, nowe formy przestępczości, swoboda szerzenia także destrukcji i demoralizacji. Tak czy inaczej, informatyka i telekomunikacja należą do tych nielicznych dziedzin nauki i praktycznej działalności człowieka, które wciąż silnie wpływają na losy naszej cywilizacji.

Jednocześnie informatyka też się zmienia. Wszechobecność światowej sieci narzuca unifikację i standaryzację formatów danych, protokołów komunikacyjnych, oprogramowania. Wydaje się, że mało pozostaje miejsca na własne, oryginalne rozwiązania. Produkcja układów scalonych, podzespołów, całych komputerów i oprogramowania systemowego skupia się w rękach bardzo nielicznej grupy globalnych producentów: tu też pole dla własnej twór-

¹⁷ Sędziwy prof. Romuald Marczyński, bardzo wzruszony, odebrał medal osobiście. Antoniemu Kilińskiemu, który zmarł w 1989 r., medal przyznano pośmiertnie, w drodze wyjątku, choć to odznaczenie do tej pory było przyznawane wyłącznie osobom żyjącym.

czości wydaje się bardzo ograniczone. Tworzeniem oprogramowania aplikacyjnego zajmuje się natomiast tysiące firm, grupujących się w silne konsorcja i oferujących gotowe oprogramowanie do wszystkich możliwych — zdawałoby się — zastosowań. Konkurencja na rynku jest niezwykle silna, a warunkiem „przebicia się” są wielkie nakłady na marketing.

W tej sytuacji pojawia się pytanie, kim jest dzisiaj informatyk? Czy ma coś jeszcze twórczego do zrobienia, jeśli nie jest pracownikiem zamożnej firmy w Krzemowej Dolinie? Czy w takim kraju, jak Polska są w ogóle potrzebni informatycy z wyższym wykształceniem? Czy — zamiast finansować studia na poziomie akademickim, badania naukowe i oryginalne projekty rozwojowe — nie byłoby lepiej i taniej skupić się na kupowaniu gotowych rozwiązań?

Istotnie, rozpowszechnienie się komputerów osobistych w domach, urzędach, sklepach, szkołach, coraz powszechniejszy dostęp do Internetu — wszystko to sprawia, że coraz więcej ludzi budzi się każdego ranka z przyjemnym przeświadczeniem, że już są informatykami. Umieją nadać e-mail, znaleźć coś za pomocą wyszukiwarki, napisać raport czy podsumować wydatki korzystając z arkusza kalkulacyjnego. Niektórzy umieją nawet zainstalować nowe oprogramowanie i wymienić kartę graficzną w jednostce centralnej. O, ci to już z całą pewnością informatycy, zwłaszcza jeśli jeszcze mają za sobą jakiś kurs programowania. Ponadto, na świecie jest coraz więcej młodych entuzjastów, którzy komputerem umieją się posługiwać wcześniej niż mówić i już w wieku kilku czy kilkunastu lat, bez specjalnych studiów, zadziwiają rodziców oryginalnymi projektami gier, biegłością w korzystaniu z Internetu czy wyczynami hakerskimi.

Ponieważ jednocześnie te umiejętności wcale nie są trudne albo można je osiągnąć bez studiowania — można spotkać się z powątpiewaniem, czy informatyka w ogóle zasługuje na status dyscypliny akademickiej. Może gdzieś w Ameryce, ale tu, w Polsce? A po cóż kształcić tylu tych inżynierów i magistrów informatyki? Komputerów przecież budować nie będziecie: robi to parę firm na świecie, u nas najwyżej będzie się składać podzespoły w całość. Odkręcić parę śrubek, wetknąć co trzeba i gdzie trzeba, przykręcić śrubki z powrotem — to każdy potrafi. Wiedza o systemach operacyjnych? Po co? Ich również nie będziecie projektować: na całym świecie są w użyciu raptem dwa czy trzy. Trzeba się zdecydować na jeden, wziąć podręcznik, nauczyć się kilkudziesięciu dyrektyw i już można zatrudnić się jako administrator systemu. Oprogramowanie? Przecież to się kupuje gotowe, na CD, albo ściąga z sieci. Ono się nawet samo instaluje: jedyne, co trzeba zrobić — to co jakiś czas „kliknąć” na przycisk „dalej”. Więc cóż pozostaje? Projektowanie stron WWW? Do tego jest potrzebny jakiś poradnik na początek i odrobina pomysłowości, bardziej zresztą plastycznej niż technicznej. Więc może wpisywanie zawartości formularzy do bazy danych przedsiębiorstwa? Tak, ale czy do tego jest rzeczywiście potrzebne wyższe wykształcenie? Co innego — zastosowania informatyki. To zupełnie inna historia. Już wiemy, że komputery będą potrzebne w technice, produkcji, zarządzaniu, usługach. Kształćmy więc inżynierów mechaników, elektryków, businessmanów, ekonomistów, tak aby umieli się posługiwać komputerem i Internetem. Dziś jeszcze może warto poświęcić temu ze dwie godziny na pierwszym roku studiów, jutro już wszyscy przyniosą tę umiejętność ze szkoły lub z domu i problem rozwiąże się sam.

Zdarza się, że poglądy takie powtarzają nawet osoby z tytułami naukowymi, na ogół z tym większym przekonaniem, im płytszy i bardziej amatorski jest ich kontakt z informatyką. Tymczasem informatyka jako akademicka dyscyplina (zarówno w sensie nauczania, jak i badań) naprawdę istnieje. Co więcej, jest w istocie dopiero na początku swej drogi. Cóż znaczy pięćdziesiąt lat istnienia *computer science* w porównaniu z wielowiekowymi tradycjami takich nauk, jak matematyka, mechanika, filozofia albo filologia klasyczna? Jeszcze nawet nie ma jednego, ugruntowanego kanonu wyższego wykształcenia informatyka. Próbami jego określenia zajmują się dwa największe światowe stowarzyszenia informatyków: *IEEE Computer Society* oraz *ACM (Association for Computing Machinery)*. Mniej więcej co dziesięć lat powoływany jest wspólny komitet, który opracowuje rekomendacje dotyczące treści programu studiów wyższych z dziedziny informatyki. Jeśli więc ktoś nie wie, czym (i jak) można zapełnić program kilkuletnich studiów wyższych na tym kierunku — powinien zapoznać się z najnowszym opracowaniem tego komitetu¹⁸.

¹⁸ *Computing Curricula 2001*, The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, <http://computer.org/education/cc2001/report/index.html>.

Pozostaje jednak pytanie, *po co* tym się zajmować, skoro właściwie wszystko albo już jest gotowe, albo zaraz będzie gotowe, zrobione przez jedną z kilku dominujących zachodnich firm. Odpowiedź brzmi: problem jest pozorny, gdyż jego przesłanka nie jest prawdziwa. W istocie, zrobiono jeszcze niewiele. Era społeczeństwa informacyjnego dopiero się rozpoczyna: z pewnością będziemy wkrótce świadkami zjawisk, wynalazków i osiągnięć, których dziś nie jesteśmy w stanie przewidzieć ani sobie nawet wyobrazić. Pole do działania jest niezwykle szerokie i miejsca starczy dla wszystkich. Ważne, by nie popełnić grzechu niewiary we własne możliwości i nie powtórzyć tego błędu, który w naszym kraju popełniono w latach sześćdziesiątych i siedemdziesiątych XX wieku, kiedy to założono, że stać nas co najwyżej na naśladowanie Zachodu, kopiowanie IBM System/360 i przestrzeganie „listy preferencyjnej”.

Po pierwsze, informatyka — to przede wszystkim *algorytmizacja*: sztuka przekształcania problemów naszego życia w zadania obliczeniowe i podawanie sposobów ich skutecznego rozwiązywania. Ta praca nie ma końca: to tworzenie modeli zjawisk ekonomicznych, społecznych, biologicznych, medycznych, geofizycznych, ekologicznych, technologicznych, modeli procesów komunikowania się, organizowania, podejmowania decyzji, rozpoznawania i produkowania obrazów i dźwięków... Trzeba wiedzieć, jak takie modele się tworzy i rozwiązuje, co na ich temat ma już dziś do powiedzenia współczesna informatyka, matematyka, fizyka, technika, jak przekształcić je w algorytmy i odpowiednie struktury danych, jak ocenić, które algorytmy są obliczeniowo „łatwe”, a które „trudne” (i co to znaczy), jakie środki techniczne: paradygmaty projektowania, języki programowania, systemy komputerowe — są najważniejsze dla danego typu zadań. Badania z tej dziedziny — to miejsce styku informatyki i innych nauk, działów wiedzy i działalności praktycznej człowieka.

Tworzeniu modeli zjawisk otaczającego nas świata towarzyszą badania nad systemami informacyjnymi oraz zupełnie nowymi technikami i algorytmami przetwarzania informacji. Szczególne miejsce mają tu algorytmy probabilistyczne i ewolucyjne, a także teorie i metody z kręgu sztucznej inteligencji, w tym sieci neuronowe i inne techniki zautomatyzowanego gromadzenia wiedzy, klasyfikowania faktów, wnioskowania. Szybko rozrastające się systemy informacyjne gromadzą ogromną ilość danych. Jeśli owe składowiska danych mają nie przekształcić się w monstrualny śmietnik, w którym wszystko jest, ale niczego nie można znaleźć — ich zawartość musi być tak zorganizowana, by użytkownik miał dostęp do danych, które będą mu potrzebne. Jak to skutecznie zrobić? Dane te mają być na ogół podstawą pewnego wnioskowania. Jak jednak prowadzić takie wnioskowanie, jeśli dane są o tak ogromnej objętości, a przy tym niepewne, niekompletne, nieprecyzyjne albo wzajemnie sprzeczne? Człowiek posługuje się w takich przypadkach inteligencją, doświadczeniem i intuicją, ale jak ma to zrobić system komputerowy? Co więcej, w ogromnych składowiskach danych ukryte są nieznanne nam dotąd zależności między różnymi, często pozornie odległymi faktami i zjawiskami. Czy można je jakoś stamtąd „wykopać”, przeanalizować i wykorzystać? Stąd badania nad systemami informacyjnymi, modelami baz danych, mechanizmami gromadzenia wiedzy, systemami klasyfikacji, wnioskowania na podstawie niepewnych danych, stąd takie terminy, jak sztuczna inteligencja, *data mining* czy *knowledge discovery*. To ważne działy współczesnej informatyki, obiecujące i dopiero na początku swej drogi.

Informatyka — to również sposoby na radzenie sobie z ogromną różnorodnością form występowania informacji: analogowej i cyfrowej, wizualnej, tekstowej i dźwiękowej, produkowanej przez urządzenia pomiarowe i przez człowieka. Informatyka sprowadza je wszystkie do sekwencji symboli o możliwie dokładnie określonej składni i semantyce, oferuje metody kompresji i kontroli poprawności, organizuje je w hierarchiczne struktury danych wygodne do przechowywania, przesyłania i przetwarzania. Oczywiście, istnieje wiele standardów reprezentowania różnego rodzaju danych: tekstowych, graficznych, muzycznych i multimedialnych. Równie wielka jest różnorodność metod i narzędzi ich przetwarzania: języków programowania, protokołów komunikacyjnych, platform sprzętowych, systemów operacyjnych, systemów zarządzania bazą danych. To nieprawda, że systemy operacyjne są dwa: UNIX i Windows, protokół komunikacyjny jeden: TCP, a języki programowania — ze trzy: C, C++ i Java. Są ich setki. Oparte są one jednak na wspólnych teoretycznych zasadach. Znajomość tych wspólnych zasad (a także zrozumienie istoty *różnic* między standardami, systemami, ję-

zykami) odróżnia profesjonalnego informatyka od przyuczonego amatora¹⁹. Dzięki temu „prawdziwy” informatyk potrafi *integrować* różne rozwiązania w system, to znaczy potrafi sprawić, by różne formaty danych, systemy operacyjne, protokoły komunikacyjne, platformy sprzętowe współistniały i współdziałały ze sobą w ramach systemu. Ta teoretyczna wiedza, uzupełniona o praktyczną znajomość kilku różnych standardów danych, języków, systemów, powinna mu wystarczyć również w przyszłości: wiadomo bowiem, że każde z dzisiejszych rozwiązań kiedyś też odejdzie do historii, tak jak odszedł ENIAC czy UMC 1.

Działem informatyki, któremu można wróżyć lata rozwoju, jest szeroko rozumiana *grafika komputerowa*. Zmysł wzroku, wraz z zadziwiającymi możliwościami panoramicznego i stereoskopowego widzenia, rozpoznawania obrazów, ich zapamiętywania i kojarzenia, stanowi jeden z najsilniejszych atutów człowieka. Dlatego dwu-, trój- i czterowymiarowa (tj. zmienna w czasie, animowana) informacja o charakterze wizualnym jest tak pożądana we współpracy człowieka z systemami komputerowymi. Stąd badania nad generowaniem realistycznych obrazów, animacją w czasie rzeczywistym, łączeniem obrazów generowanych komputerowo z obrazem z tradycyjnej kamery itd. — aż do łączenia wrażeń wzrokowych z innymi wrażeniami zmysłowymi tak, by otrzymać złudzenie sztucznej rzeczywistości (*virtual reality*). Zastosowania tych badań nie ograniczają się do gier, filmu, rozrywki, wirtualnych podróży do odległych miejsc czy do świata fantazji. To także możliwość trenowania pilotów, kierowców, operatorów urządzeń pracujących w niebezpiecznych warunkach, możliwość bezpiecznego ćwiczenia nowych technik chirurgicznych. Oczywiście, i tutaj już wiele zrobiono: wszyscy widzieliśmy choćby stada dorodnych wirtualnych dinozaurów przemierzających *Jurassic Park*. Przypomnijmy jednak, że każda klatka tego filmu — to kilka godzin pracy najszybszych ówczesnych komputerów, nie licząc poprzedzających to przetwarzanie setek godzin projektowania i „ożywiania” sztucznych stworów. Czy nie powinno się tego robić szybciej i lepiej?

Wyzwaniem dla informatyków są również zjawiska *rozproszenia* i *współbieżności* występujące we współczesnych sieciach i systemach komputerowych. Technologie sieciowe (w tym także internetowe) umożliwiają tworzenie *systemów rozproszonych*, w których liczne komputery, wykonujące pewne wspólne funkcje, są rozrzucone na dużym obszarze geograficznym, wywołują się wzajemnie na odległość i współużytkują również rozproszone zasoby: dane, elementy oprogramowania, moc obliczeniową itd. Nieuchronne są przy tym opóźnienia komunikacyjne. Zwłaszcza w dużych systemach należy się liczyć również z losową niesprawnością łączy i samych węzłów sieci. Stwarza to problemy ze spójnością danych oraz aktualnością informacji o lokalizacji zasobów. Ponadto, system rozproszony jest z natury *współbieżny*. Znaczy to, że wiele procesów biegnie w nim jednocześnie, a więc muszą się w pewien sposób koordynować i synchronizować. Jest to szczególnie niełatwe wobec opóźnień i niepewności co do stanu partnera, z którym mamy współpracować na odległość. Informatycy wiedzą również, że ze *współbieżnością* wiąże się możliwość występowania szkodliwych zjawisk, które nigdy nie mają miejsca w przypadku obliczeń nawet złożonych, ale sekwencyjnych: zakleszczenia (*deadlocks*), „martwe pętle” (*livelocks*), naruszenie reguł wzajemnego wykluczania itd. Teoria i praktyka tworzenia systemów rozproszonych wolnych od takich błędów ma przed sobą bardzo wiele problemów do rozwiązania.

Gatunkiem systemów, z jakim przeciętny użytkownik komputera osobistego na ogół nie ma do czynienia, są *systemy czasu rzeczywistego*. Są to zazwyczaj systemy *reaktywne*, to znaczy takie, których zadaniem jest reagowanie na zdarzenia nadchodzące do nich z otoczenia. W przypadku systemów czasu rzeczywistego na zachowania systemu są dodatkowo nałożone ostre ograniczenia czasowe. Nie jest wszystko jedno, czy (jak w internetowej wyszukiwarce) system raz odpowie w ciągu ułamka sekundy, a kiedy indziej w ciągu pół minuty. Systemy tego typu są budowane do celów sterowania procesami technologicznymi, liniami produkcyjnymi, zespołami robotów, ruchem kolejowym czy ulicznym, zespołami wind i zabezpieczeń przeciwpożarowych w wieżowcach itd. Stosuje się w nich inne zasady organizacji sieci, inne

¹⁹ Profesjonalny informatyk powinien być trochę rozbawiony, a trochę urażony, gdy go ktoś spyta, jaki zna język programowania. To tak, jak gdyby spytać kierowcę o najwyższej kategorii uprawnień zawodowych, jakiej marki samochodem umie kierować. Każdej, oczywiście. Proszę mi dać jedynie trochę czasu na oswojenie się, potem — to już tylko kwestia praktyki.

protokoły komunikacyjne, elementy sprzętowe, systemy operacyjne, oprogramowanie aplikacyjne niż w najpowszechniejszych zastosowaniach: domowych, biurowych i w powszechnie dostępnych sieciach wchodzących w skład Internetu. Najczęściej systemy takie są najpierw projektowane w wersji próbnej, uruchomieniowej, a następnie „dopasowywane” i „strojone” na miejscu tak, by dobrze obsługiwały dany, specyficzny obiekt. Na etapie takiego dopasowywania głównym problemem jest *analiza wydajności* systemu, wyszukiwanie i poprawianie jego wydajnościowych „wąskich gardeł”. To też nie jest praca dla informatyka — amatora, który nie wie, co system ma „w środku”.

Szczególnym rodzajem systemów reaktywnych (także najczęściej czasu rzeczywistego) są *systemy wbudowane (embedded systems)*. Z wyglądu nie przypominają komputera, z jego klawiaturą, monitorem, jednostką centralną. Prawdę mówiąc, niczego z wyglądu nie przypominają, gdyż kryją się pod tablicą wskaźników samochodu, wewnątrz telefonu komórkowego, w wieży CD, aparaturze wideo, telewizorze, aparacie fotograficznym i tysiącach innych urządzeń. Są tam właśnie *wbudowane* i jako układ sterowania zapewniają użytkownikowi bogaty zestaw wygodnych funkcji, a czasem (jak w przypadku cyfrowych telefonów komórkowych, odtwarzaczy CD czy telewizorów cyfrowych) jako jednostka o ogromnej mocy obliczeniowej realizują podstawowe funkcje całego urządzenia. Większość z nich to dość standardowe systemy mikroprocesorowe, przystosowywane do specyficznych funkcji przez oprogramowanie, wpisane na stałe do ich pamięci.²⁰ Zdarzają się jednak zastosowania, w których konieczne jest opracowanie specjalizowanych układów scalonych. Ich projektowanie — to miejsce styku informatyki i mikroelektroniki. Jeśli przewiduje się, że dany system wbudowany będzie częścią urządzenia produkowanego w tysiącach sztuk (jak np. w przypadku telefonów komórkowych), to może opłacać się opracować nowy, specjalizowany układ scalony (ASIC — *Application Specific Integrated Circuit*). Jeżeli skala produkcji nie będzie tak duża — lepiej wykorzystać układy bardziej uniwersalne (np. FPGA — *Field Programmed Gate Arrays*), dostosowywane tylko do specyficznych funkcji.

Można przewidywać, że liczba urządzeń wyposażonych w systemy wbudowane będzie gwałtownie rosła. Co więcej, wydaje się niewątpliwe, że ich rozwój pójdzie nie tylko w kierunku zwiększania ich złożoności²¹ i funkcjonalnej różnorodności, lecz także w kierunku ich integracji w lokalne sieci urządzeń współpracujących ze sobą w ramach mieszkania, domu, osiedla, wreszcie w ramach światowej sieci informatycznej. Podobnie w przypadku pojazdów: zapewne każdy samochód będzie wyposażony wkrótce nie tylko we wbudowane cyfrowe układy sterowania silnikiem czy hamulcami, lecz także system lokalizacji współpracujący z GPS, „inteligentny” alarm i mobilny interfejs do bazy danych dróg, ulic i hoteli całego świata. Będzie sam przypominał o przeglądach, wymianie opon itd. Ta wizja zresztą nie każdego zachwyca: uzależnimy się od komputerów i nie będziemy się potrafili bez nich obejść, a wielkie bazy danych będą gdzieś gromadziły informację o każdej stacji benzynowej, którą odwiedziliśmy i o każdym otwarciu drzwi w naszym mieszkaniu.²² Wielu zapewne ucieknie w Bieszczady lub na Borneo, ale wydaje się, że dla informatyków i elektroników pracy nie zabraknie.

Tematyką, której stosunkowo od niedawna poświęca się coraz więcej uwagi, są zagadnienia *bezpieczeństwa danych* i *autoryzacji dostępu* do sieciowych zasobów informatycznych. Wiadomo, że zgromadzone przez nas i naszą instytucję dane, programy, treść korespondencji itd. — mogą być przedmiotem zainteresowania przestępcy, konkurenta w interesach, terrorysty, szpiega lub choćby złośliwego maślaka, który będzie miał ochotę dla dowcipu „wyczyścić” nasz dysk za pomocą wirusa albo dla zabawy włamać się poprzez światową sieć do systemu sterowania strategicznych rakiet balistycznych wielkiego mocarstwa. Stąd intensywne badania nad nowymi, bezpieczniejszymi protokołami komunikacyjnymi, technikami kryp-

²⁰ Szybko rozpowszechniający się w ostatnich latach język Java ma pewne cechy, które sprawiają, że wyjątkowo dobrze nadaje się właśnie do programowania tego typu systemów.

²¹ Mikroelektronika potrafi już w tej chwili wytworzyć układy zawierające cały system, wraz z dość bogatym oprogramowaniem systemowym, w ramach jednego układu scalonego.

²² Ta deprymująca wizja nie jest bynajmniej sprawą odległej przyszłości. Już teraz awaria systemu komputerowego praktycznie paraliżuje każdy większy sklep czy urząd, a częste płacenie za zakupy kartą płatniczą pozostawia w jakichś bazach danych ślad naszych upodobań, zapis trasy naszych podróży itp.

tograficznymi, zabezpieczeniami i informatycznymi zaporami (*firewalls*), chroniącymi systemy państwowe, wojskowe, przedsiębiorstwa, a także osoby prywatne przed niepożądanym dostępem. Od ich rozwoju i skuteczności zależy nie tylko bezpieczeństwo obecnych zasobów sieciowych, lecz także przyszły los nowych technik przedsiębiorczości, handlu i usług zakładających wykorzystanie Internetu (*e-business, e-commerce*). Wszystkie one bowiem opierają się na przekazywaniu poprzez sieć zamówień, płatności, numerów kart kredytowych, podpisów itd., które powinny pozostać poufne.

Wszystkie te systemy, zarówno ich sprzęt, jak i oprogramowanie, powinny działać w sposób *wiarygodny*. Już na etapie tworzenia oprogramowania oraz projektowania i produkcji sprzętu trzeba zadbać o to, by nie było w nich (lub było jak najmniej) błędów. Dlatego ważnym kierunkiem badań są prace nad teorią i praktyką *testowania i diagnostyki* systemów cyfrowych. Co więcej, nawet dobrze zaprojektowany i wstępnie sprawdzony system w czasie pracy podlega (jak każde urządzenie na świecie) uszkodzeniom, awariom, zaburzeniom mechanicznym lub elektromagnetycznym itd., które mogą w sposób trwały lub przemijający spowodować niesprawność jego całości lub części. W przypadku domowego komputera użytkownik co najwyżej cicho zaklnie, wyłączy system i uruchomi go ponownie. W przypadku systemu bankowego, światowego systemu telekomunikacyjnego, systemu sterującego samolotem, linią produkcyjną — mogłaby to być katastrofa o nieobliczalnych skutkach. „Prawdziwe” systemy informatyczne powinny więc być tak projektowane, by były *fault-tolerant*, to znaczy odporne na awarie. Muszą być wyposażone w mechanizmy wykrywania błędów i uszkodzeń. Muszą mieć zapasowe zasoby: łącza komunikacyjne, procesory, serwery, bazy danych zawierające stale aktualizowaną replikę danych roboczych. Muszą umieć samoczynnie zrekonfigurować się, odtworzyć stan sprzed uszkodzenia i „wstać” po awarii tak, by było to prawie niezauważalne dla bieżących w nich procesów obliczeniowych.

Badania nad tymi zjawiskami były zawsze bardzo ważne dla informatyki. Widzieliśmy, że w latach pięćdziesiątych i sześćdziesiątych XX wieku dążenie do zapewnienia komputerom odpowiedniej niezawodności było w ogóle jednym z podstawowych motorów postępu technologicznego w dziedzinie układów scalonych. Opanowanie produkcji układów scalonych i szybki postęp w tej dziedzinie pozwoliły — z kolei — na budowanie niezawodnych systemów o coraz większej złożoności, w których można było wreszcie pokusić się o implementację skomplikowanych pomysłów teoretycznych i *software’owych*. Tematyka ta wraca więc niejako przy wyższym poziomie wymagań: chodzi już nie tylko o niezawodność (a więc o to, by wydłużyć prawdopodobny czas do wystąpienia uszkodzenia), lecz również o samoczynne radzenie sobie z niesprawnością, która już wystąpiła. Wydaje się, że znaczenie tych badań będzie rosło wraz z postępującym uzależnianiem się różnych dziedzin życia i działalności człowieka od wszechobecnych systemów komputerowych. Wiele spośród tych systemów będzie *safety-critical*, co znaczy, że od ich działania będzie w krytyczny sposób zależeć bezpieczeństwo wielu ludzi. W tym przypadku zagadnienia testowania, diagnostyki, tolerowania uszkodzeń — generalnie: wiarygodności — będą zawsze szczególnie ważne.

Ważną częścią profesjonalnej informatyki jest *inżynieria oprogramowania (software engineering)*. Zajmuje się ona metodologią i technologią tworzenia złożonego *software’u*. Współczesne produkty *software’owe* są najbardziej funkcjonalnie skomplikowanymi tworem ludzkiego umysłu. Nawet ktoś, kto osobiście napisał i uruchomił program komputerowy liczący kilkaset czy nawet kilka tysięcy wierszy kodu źródłowego, nie zdaje sobie w pełni sprawy jaka przepaść komplikacji dzieli jego pracę od procesu tworzenia oprogramowania mającego na końcu rozmiar kilkaset tysięcy czy kilku milionów wierszy, zwłaszcza jeśli projektowany system jest rozproszony i mają z niego korzystać tysiące ludzi. Takiego zadania nie wykona jeden człowiek, który byłby w stanie intelektualnie panować nad całością projektu. Zatrudnienie zespołu kilkudziesięciu czy kilkuset programistów oznacza natomiast konieczność zaplanowania nad monstrualnym chaosem, który grozi całemu przedsięwzięciu. Najpierw trzeba możliwie precyzyjnie wyrazić i możliwie formalnie opisać założenia i wymagania dotyczące oczekiwanych funkcji systemu. Nie można założyć, że się będzie je mieć po prostu „w głowie” lub zapisane w postaci kilku ogólnikowych zdań. Do tej specyfikacji wymagań będzie się potem odwoływać wiele osób i zespołów projektowych, musi być więc ona „świętą księgą”, podstawowym dokumentem projektu. Specyfikacja wymagań powinna być uzgodniona z przyszłym użytkownikiem i przez niego zatwierdzona. Musi być więc wyrażona tak, by

użytkownik ją zrozumiał, nie będąc zawodowym informatykiem. Projekt trzeba podzielić na części czy moduły, a ich wykonanie powierzyć równolegle pracującym zespołom programistów. Jest oczywiste, że należy przy tym szczegółowo określić strukturę i nazewnictwo danych, budowę komunikatów itd., które znajdują się „na styku” między zespołami: jedna litera niezgodności w nazwie zmiennej spowoduje, że owe moduły nie będą dobrze współpracowały między sobą. Każda poprawka czy modyfikacja musi dotrzeć do świadomości wszystkich zainteresowanych, w przeciwnym przypadku praca projektantów pójdzie na marne, a projekt szybko przekształci się w jedno wielkie nieporozumienie. Równolegle z implementacją należy opracować testy pozwalające możliwie gruntownie zbadać poprawność poszczególnych modułów, a także testy integracyjne, sprawdzające poprawność współpracy między modułami na bolesnym zazwyczaj etapie składania systemu w całość. Pozostają jeszcze do opracowania testy funkcjonalne całości, odpowiadające na pytanie, czy użytkownik rzeczywiście może zrobić wszystko to, co sobie wymarzył, zamawiając wykonanie systemu i płacąc zań niemałe na ogół pieniądze. A jeszcze opracowanie dokumentacji systemu, jeszcze podręczniki dla użytkownika, szkolenie personelu, ewaluacja systemu w okresie próbnym, instalacja w (być może) setkach węzłów sieci...

Inżynieria oprogramowania dzieli ten złożony proces na dobrze określone etapy, precyzuje procedury postępowania na każdym z nich, oferuje sposoby oceny złożoności *software'u*, przewidywania prac- i czasochłonności projektu, a także ryzyka ewentualnego niedotrzymania terminów. Co więcej, są dostępne na rynku produkty *software'owe*, które pomagają w pracy zespołom projektantów oprogramowania. Te tzw. narzędzia CASE (*Computer-Aided Software Engineering tools*) ułatwiają sporządzanie specyfikacji systemu, pilnują spójności dokumentacji, przechowują kolejne wersje oprogramowania, pomagają w organizacji pracy zespołowej, projektowaniu testów, produkują fragmenty kodu programu, dokumentację itd. Inżynieria oprogramowania jest bardzo ważnym działem współczesnej profesjonalnej informatyki. Warto podkreślić, że niedoceniecie jej zasad przy realizacji dużych projektów może prowadzić do kompromitujących niepowodzeń, dezorganizujących później życie milionów ludzi czasem na taką skalę, że musi się tym zajmować rząd, parlament i NIK.

Na styku matematyki i logiki z praktyką projektowania systemów cyfrowych leży dział informatyki, którego znaczenie wyraźnie rośnie na przestrzeni ostatnich kilkunastu lat: zastosowanie *metod formalnych* do weryfikacji poprawności systemów. Pytanie, czy dany system, program, urządzenie jest poprawny (a więc, czy istotnie umie zrobić to, do czego został zbudowany) dręczyło zawsze twórców zarówno sprzętu, jak i oprogramowania. Najpowszechniejszym sposobem radzenia sobie z tą wątpliwością jest oczywiście testowanie systemu: sprawdzanie jego zachowania dla różnych danych wejściowych, różnych bodźców zewnętrznych itp. Wykonanie takich testów, wielogodzinna próbna eksploatacja lub symulacja modelu urządzenia zwiększa zaufanie do projektu, niestety nie stanowi jednak *dowodu* jego poprawności. Po pierwsze, najczęściej nikt nie jest w stanie eksperymentalnie zbadać *wszystkich* możliwych układów danych. Po drugie, system nie tylko powinien robić to, do czego był budowany, ale także *nie powinien* robić tego, czego mu nie wolno: na przykład gubić komunikatów, wykonywać kilkakrotnie operacji, które powinny być wykonane tylko raz, „zawieszać się” itd. Tego rodzaju niepożądane zachowania bardzo często nie mają charakteru systematycznego błędu, spowodowanego pomyłką programisty lub projektanta (co ujawniłoby się w toku testowania), lecz są (zwłaszcza w przypadku systemów współbieżnych) rezultatem rzadkiego, losowego zbiegu okoliczności, który w symulacji (lub próbnej eksploatacji) systemu może wystąpić równie dobrze już w pierwszych sekundach, po — powiedzmy — stu tysiącach godzin, albo wcale.²³ Tak czy inaczej, testowanie (lub symulacja) układu sprzętowego czy oprogramowania oznacza w istocie jedynie zademonstrowanie, że w owych kilku zbadanych przypadkach (lub przez owych kilkadziesiąt godzin próbnej eksploatacji) system

²³ Wydawałoby się, że tak rzadkimi błędami można by się w ogóle nie przejmować. Kłopot polega jednak na tym, że jeśli w pewnym urządzeniu błąd pojawia się średnio raz na sto tysięcy godzin, to w zbiorze liczącym sto tysięcy sztuk tych urządzeń będzie się ujawniał — statystycznie rzecz biorąc — raz na godzinę. Tak więc, gdyby był to na przykład wbudowany system cyfrowego sterowania zainstalowany w stu tysiącach samochodów — to mógłby powodować średnio 24 wypadki drogowe na dobę. To straszna cena za zlekceważenie błędu o tak pomijalnie małym — zdawałoby się — prawdopodobieństwie wystąpienia.

zachował się poprawnie. Badania nad formalnymi metodami weryfikacji systemów zmierzają natomiast do tego, by w matematycznie ścisły sposób *udowodnić*, że poprawne zachowanie jest (lub nie jest) właściwością projektu, niezależną od konkretnych wartości danych lub od zbiegu okoliczności.

Badania nad zastosowaniem metod formalnych do opisu i weryfikacji poprawności systemów mają już bardzo długą historię. W ich toku powstały całe nowe gałęzie logiki, nowe metody modelowania zachowań systemów i ich danych oraz narzędzia programowe wspomagające proces weryfikacji. Rezultaty te, pojęciowo i matematycznie dość trudne, przenikają jednak do praktyki stosunkowo powoli. Znajomość tych wyników (i narzędzi informatycznych wspomagających ich stosowanie w praktyce) jest przy tym stosunkowo lepsza w kręgu zespołów i firm produkujących sprzęt, niż wśród projektantów oprogramowania. Przyczynia się do tego z pewnością fakt, że formalna weryfikacja projektów sprzętowych jest nieco prostsza. Ponadto — co dobrze ilustruje przypis do poprzedniego akapitu — masowa produkcja układów scalonych nakłada na producenta szczególną odpowiedzialność. Tymczasem w przypadku oprogramowania najbardziej masowo wytwarzane produkty są przeznaczone dla nieprofesjonalnego odbiorcy, nie są *safety critical*, a ich tworzeniem rządzi często nie tyle zawodowa odpowiedzialność, co pośpiech i terminarz kampanii reklamowej. Z tych zapewne powodów wykrycie błędu w działaniu komercyjnie dostępnego układu scalonego stanowi sensację na skalę światową, zaś skandaliczny błąd w komercyjnie dostępnym programie należy wciąż do codzienności. Jeśli jednak mamy zmierzać do ery społeczeństwa informacyjnego, które nie tylko zabawia się komputerami, lecz korzysta z osiągnięć informatyki w prawie wszystkich dziedzinach życia, to sytuacja ta musi ulec zmianie.

Wymienione wyżej i krótko omówione działy informatyki są wybrane subiektywnie i zapewne nie wyczerpują obrazu całej dziedziny. Wspomniano o nich przede wszystkim dlatego, że właśnie one stanowią obecną specjalność badawczą i dydaktyczną Instytutu Informatyki. Uprawiana w Instytucie tematyka badawcza ma wiele punktów wspólnych z badaniami prowadzonymi w innych Instytutach Wydziału. Oczywiście, nie jest to przypadek: to konsekwencja naturalnego rozwoju elektroniki, mikroelektroniki i telekomunikacji, które od lat coraz szerzej korzystają z dorobku informatyki, a jednocześnie dla tejże informatyki są źródłem inspiracji oraz nowych rozwiązań teoretycznych i technicznych. Każda z tych dziedzin ma swoją tożsamość, własne — specyficzne — problemy, z drugiej strony w wielu miejscach zrasta się z sąsiednią dziedziną tak dokładnie, że nie jest nawet celowe zastanawianie się nad granicami między nimi. Przy uszanowaniu tego, co odmienne i specyficzne, badania prowadzone w różnych instytutach powinny konkurować między sobą i wspierać się wzajemnie tam, gdzie tematyka jest podobna.