3$^{\text{rd}}$ National Conference
Databases for Science
INFOBAZY 2002

24–26 June 2002
Gdansk-Sobieszewo, Poland

Selected Papers and Abstracts
guest-edited by Antoni Nowakowski

Database Systems for Tomorrow: New Challenges and Research Areas

Krzysztof Goczyła

Department of Applied Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdansk, Poland, kris@eti.pg.gda.pl

(Received 30 October 2002)

Abstract: Since the mid-80s, considerable progress has been achieved in relational database technology. The main achievements have been in high performance, high reliability and availability, scalability and development tools. However, the environment for database systems is rapidly changing. There are new challenges that originate from the present hardware technology achievements, as well as from new kinds of data resources that hardly conform to the well-established relational data model (*e.g.* data from the Web). In the paper, we present the new challenges and research areas, as well as motivations behind them.

Keywords: database systems, new architectures, Web technologies, non-relational models, integration

1. Introduction

Contemporary relational database management systems (RDBMS) are mature, sophisticated software systems that are supported by advanced hardware solutions. RDBMSs are commonly used in all real-life areas where computers are present. Database systems market is large – it is estimated that the annual volume exceeds US$ 10 billion and is expected to grow steadily, even in the face of the recession that the world's economy experiences. This huge amount of money engaged in the database systems business has both advantages and disadvantages. One apparent advantage is that the community of database systems users feel comfortable with technical support, stability and their systems maintenance. There is little chance that the software they purchased (usually quite expensive) will not be maintained and periodically upgraded by its vendor and become an expensive but useless gadget instead. Another advantage is that large RDBMS corporations invest enormous funds into research and development, resulting recently in remarkable advances in such database fields as replication and parallelism. On the other hand, big corporations tend to monopolize the database systems market, which results in narrowing down the development trends into several and quite restricted "safe" areas and prohibiting development of new "risky" technologies. A clear example is the way the object-oriented paradigm [1] is being introduced into the database world. However strange it is, contemporary database technology is apparently the only area of information technology where the principles of object orientation were practically not applied. Of course, some elements of the object-oriented paradigm do appear in object-relational database management systems (ORDBMS) [2], however, they considerably diverge from the full object-oriented model. The latter is fully implemented in object-oriented databases management systems (OODBMS) that still remain on the margins of commercial database applications.

It seems, however, that nowadays the database worlds – both the commercial one and the research one – face completely new challenges that will force changes in database technology much deeper than those which occurred at the end of the previous century. In the nearest future, we will face serious changes in computer systems technology for large database systems. At the same time, the unstoppable growth of the Internet and its informational resources accumulated in the ubiquitous World Wide Web create completely new requirements for functionality of data repositories and data analysis tools. In the following sections, we will take a closer look at these causes and prerequisites, referring them to research areas that should be explored in order to cope with these new requirements.

2. The challenges

One may formulate three main reasons for undertaking new research in the database systems technologies:

1. As a result of the rapid development of the Web technology, it has become quite easy and inexpensive to make information of any kind and quality available to millions of potential information "consumers".
2. Many new applications appear that require programs to be integrated with voluminous data of complex and heterogeneous structure.
3. The rapid development of hardware technology invalidates assumptions that various techniques currently applied in database systems are based upon.

2.1. Exploiting information resources of the Web

The Web can be seen as a large, distributed (and unmanaged) database. It would be extremely difficult to estimate its size in bytes (or terabytes), but certainly the number of Web users (*i.e.* people who have access to workstations connected to the Internet and at least occasionally make use of it) can be counted in hundreds of millions. The number of Internet hosts that run Web servers is of course much lower but, certainly, also large and rapidly growing into tenths of millions. However, the Web is not a "normal", fully functional database – it is not managed in any uniform way. For this reason, the Web is a huge storehouse of data rather than a real database. "Real" DBMSs are present in this storehouse, but play a secondary, back-end role: they are repositories of structured data dynamically used to fill predefined static HTML pages used to present information by Web sites. Although DBMSs are common in such Web services as e-commerce sites or informative sites, still, the majority of information delivered by Web sites is of static nature, with no explicit or implicit structure nor schema.

It is expected that in the near future the proportions will reverse. The majority of information delivered by Web sites will be dynamically created, based on the content of back-end specialised databases. This information will no longer be presented as HTML pages, but rather as XML documents [3], that are better suited for description of data with rich structure. This in turn requires better integration of XML and DBMSs. Nowadays some database systems vendors (IBM, Oracle, Microsoft) offer extensions to their systems that facilitate storing and processing XML data. However, there is no widely accepted standard in this area. Efficiency of XML processing is also not satisfactory, because XML features are implemented as a layer on top of a relational engine.

There is also another facet of dependency between the Web and database systems. Data acquired from the Web, whether created statically or dynamically, must be stored and eventually processed at client systems, so that they can be useful for a group of users they are addressed to (*e.g.* for businesses of some kind). Taking into account the possible size and complexity of data, this creates severe requirements on client tools for management and analysis of Web data. To sum up, users seek powerful and friendly tools for Web data warehousing [4].

2.2. Integration of programs and data

Traditional database systems used to store data only. In contemporary database systems, relational and object-relational, a possibility to store pieces of program code has been introduced. These code pieces take the form of *stored procedures* and *triggers*. Stored procedures may be written in a language native to a DBMS, a language that is a part of an SQL standard [5], or even in a common programming language like C or Java. In the latter case, however, a programmer must tackle with the *impedance mismatch* problem, *i.e.* the problem of data type conversions and transferring database entities into host language variables.

In database systems, a program is still a "second-class citizen": programs cannot be queried, outputs from programs cannot be directly presented as inputs to SQL queries, programs do not have their consistent model. In general, it results from the lack of tight integration between programs and the data they manipulate. Such an integration is postulated in the object-oriented approach to systems development: objects can be treated as modules that encapsulate data and operations. But, as mentioned above, object orientation enters the world of databases rather through the back door, making small and shy steps called "extensions", "cartridges" or "modules" of some kind.

Making the process of integration more active is essential for many big organisations that have at their disposal hundreds, or even thousands, of mission-critical applications that operate in separation, making their interoperability awkward. Such applications could be integrated if they had a common platform – a database that could store them together with the data they consume and produce. Software engineering is another area where the problem of integration is crucial. In this complex information technology area descriptions and specifications of applications being developed (models, dictionaries, interface definitions, *etc.*), their source code and testing data constitute an integral whole and need efficient environments for storing and processing programs and data. Computer aided software engineering (CASE) tools begin to provide such environments and rapid progress in their functionality and power is expected.

2.3. Keeping pace with hardware advances

No one knows where and when the progress in hardware will stop, or even decelerate. Processors become more productive, disks become larger, communication media become faster. Soon we will be faced with computer systems where the main memory of a terabyte will serve as a buffer pool for databases of petabytes. As a result, practically every table of a relational database will be capable of being kept in the main memory for faster access and data retrieval. This would demand new data structures and access algorithms, as well as new database tools that would accommodate the changed architecture of computer systems.

Database systems equipped with such an advanced hardware will also have to be equipped with advanced, preferably automated, administration techniques and tools. They include automatic installation, configuring, tuning, failure recovery, and even programming. A human will simply not be able to manage such complicated systems due to the volume of information to be interpreted and the extremely fast response needed to react to rapidly changing workloads.

The progress in hardware efficiency will create new demands on scalability and accessibility of DBMSs. In the near future we will need database systems that will efficiently serve a hundred of thousands of concurrent users without trying their patience. Such systems will have to be ready to store and make accessible petabytes of data, processed in parallel by thousands of processors. These figures exceed by two orders of magnitude the parameters of the contemporary commercial database systems. Currently used techniques and technologies do not enable database systems to achieve such dramatic progress.

Another source of workload for database systems are miscellaneous devices and gadgets commonly used in everyday life, like mobile phones, home electronics appliances, magnetic chip ("smart") cards, *etc.* In the nearest future, most of such popular devices will be equipped with processors and the appropriate dedicated software – together referred to as *embedded systems.* We will live in intelligent buildings, drive intelligent cars, shop in intelligent vending machines, cook in intelligent utensils, and so on. Billions of such devices will be able to communicate with their environment via the Internet (or its successor), which will require millions of servers for these "gizmos" to operate. Traditional two- and three-tier software and hardware architectures will certainly not be able to serve small, but numerous applications of this kind. Moreover, the devices will not have a traditional user interface nor any administration interface – they will have to be self-managing, self-configuring, self-tuning, self-identifying, self-protecting, self-repairing, and, eventually, self-destroying.

3. The research areas

In the following sections we present detailed insights into the main research areas following from the above mentioned problems. These areas are expected to dominate the efforts of the database community in the first decade of the 21st century [6].

3.1. Plug and Play *database systems*

The first reason for development of *Plug and Play* DBMSs are computer systems embedded into everyday life appliances, as discussed above. The systems will not be parameterised by an administrator, so they will have to automatically adapt to ever-changing work conditions. The very first step towards this goal is the development of *self-tuning* DBMSs that will be "intelligent" enough to be able to set themselves hundreds of performance parameters that otherwise would have to be set manually (but they could not, for the reasons mentioned in Section 2.3) by database administrators (DBAs). The next step is automatic choice of physical database organisation, for instance automatically changing disk file structures for data storage and indexing. The next, and much more challenging, problem is automatic design of logical database schema with appropriate integrity constraints, followed by automatic binding of these schemas with – also automatically developed – applications like report generators or data presentation tools. One of the measures towards self-tuning is collecting and analysing parameters of workload generated by production environments together with internal system performance parameters. Based on the data collected, the system could choose appropriate values of configuration parameters and system settings, select proper algorithms and data structures, decide on optimisation strategy *etc.*

Another, but equally important, aspect of *Plug and Play* DBMSs is the problem of information discovery. As has been mentioned earlier, the Web is a huge storehouse consisting of heterogeneous information resources. Some of these resources are databases and their share in overall Web information capacity will grow. Future database systems, fully integrated with the Web, will have to co-operate with other database systems in the Web just after installation in a corporate intranet or in the global Internet. They will have to find the other "friendly" systems, ready to co-operate across the network. There is a clear analogy with discovery by an operating system of new hardware just attached to a computer. To achieve such a high degree of automatic integration, a comprehensive metadata standard is needed that would cover the structure and semantics of all the objects managed by a database system. In other words, databases must have a common language powerful enough to present their schemas and data to the outer world and to make them available to their "friends".

3.2. Large federated database systems

Autonomous databases are called *federated* if they have agreed to conform to a set of rules so that they are able to co-operate on behalf of one application (particularly, of one transaction). Nowadays, the Web may be treated as a huge federated information system, although the rules governing their components are quite loose (in practice they consist of a set of simple protocols like http or ftp). It is expected that in the near future billions of Web clients will make use of millions of co-operating database systems. Such a client (for instance, associated with an embedded system) will not bother about which particular database stores data needed for the device to fulfil its functions. Appropriate data should be reliably delivered by a federated system, which may possibly have to perform a database transaction over thousands of databases. This would pose quite new challenges for query optimisers. First, they must take into account that some of the federated database systems may not be available due to failure or may deny co-operation due to security constraints. Then, data needed for the transaction may be replicated on many database servers, located in various remote places of the whole system. Also, response time for a given query may remarkably depend on the current workload of the network, which traditional optimisers usually neglect. As a result, a cost-based query plan may have to be dynamically modified, reflecting changes in the workload of the federated system.

Another challenging problem in the processing of federated queries is balance between the expected accuracy of query results and the time allowed to perform the query. Let us assume that we need an average salary of a corporation consisting of 1000 departments, each running its own local database. We probably will be satisfied if we promptly receive an approximate value for the average, that can be gradually refined as time elapses. If we are satisfied with the current accuracy, we can stop the execution of the query, or else we can wait for the exact result (if one exists). In a traditional federated system we have to wait for the result of a distributed query until an exact result is produced, whether this accuracy is really needed or not.

The problem of imprecise, approximated data is inherent not only to queries addressed to federated systems, but also in the formulation of the queries themselves. Let us have another exemplary federated query: "Find good Italian restaurants that are located near my home". A system to which the query has been submitted must first refine the concepts of "good" and "near" criteria used in the query. Most probably, the user will not be happy to have to refine the criteria by himself/herself, so the system should be constantly taught by being "fed" with some kind of knowledge concerning how to interpret vague or ambiguous queries. The next problem is how to determine databases that could store data useful for the query. Under consideration there may be thematic databases on restaurants or on tourist places, as well as geographical databases, and others. The problem of integration surfaces once again: a federated system may be composed of databases of different kinds, possibly conforming to different standards and paradigms.

Another crucial aspect of integrating database systems into large federated systems is co-operation between applications running in the federated systems. Let us consider two e-commerce applications: one runs for a manufacturer, the other runs for a warehouse. It is important that the two applications could understand each other so that they start co-operation as soon as they have found each other. To this aim a common language is needed that could be used on the interfaces of the applications running in the federated system or in the Web. This language should be flexible and powerful enough for the applications to communicate their functions and to define their data. A step towards this goal is the *Unified Modelling Language* (UML) [7], primarily devised for modelling and developing applications rather than precise functional specifications. Together with XML for metadata definition, UML may be a promising tool for exchange of functions and data between applications.

3.3. New database system architectures

The progress in hardware technology, described in the previous section, enables developers to create more and more powerful database applications. The best performance is achieved when applications are run within parallel systems of a *shared-nothing* architecture, where each processor has its own main memory and own disk memory, and the only shared resource is a communication medium. We expect further intensive development of parallel systems that could co-operate in large high-performance and high-availability clusters. It is a great challenge for DBMS software developers. The tasks that have to be considered are: load balancing among parallel system nodes, developing partitioning and replication strategies, and creating optimal query plans for such complex working environments.

As has been pointed out before, soon database engines will be able to use main memory buffers of terabyte capacity. Relational tables crucial for a given application, together with the corresponding indices, will reside entirely in the main memory for a long time. It means, for instance, that traditional access methods based on B-trees will no longer be adequate. Indeed, B-trees are not the best index structures for in-memory access, as they are based on dividing data into chunks (disk pages), which is inappropriate for in-memory data. Other database mechanisms that require re-thinking are transaction handling, recovery, concurrency and all the other techniques that make use of main memory buffers.

The development of disk storage technology manifests itself in rapid growth of disk capacities and transfer rates. As a result, the seek time (*i.e.* the time needed to move disk arms to a disk cylinder needed) becomes the bottleneck of disk storage throughput. This requires development of new storage organisations and new access strategies (*e.g.* access requests scheduling) that would minimize disk arms activity.

Many database applications, in particular in the fields that much rely on visualisation techniques, require large volumes of data. Soon these volumes will attain hundreds of petabytes. This will come to reality when disks have achieved at a reasonable price so large capacities that manipulating such voluminous data will be feasible in the traditional 2-level storage architecture. Another possibility is the introduction of new storage media (*e.g.* based on holography) that would be mature enough to be applied in commercially available computer systems. These new media could be used as a third, the slowest but the most capacious, level of database storage. It is clear that the advent of 3-level storage of exabyte capacity offers novel capabilities in archiving and replication methods, thus increasing reliability and availability of database systems. This can lead directly to "never-fail" systems that, from the user's perspective, are always up and running and never lose data.

In this context, it is worth mentioning the steady interest of vendors and developers in 3-tier architectures of database applications, where only one program (DBMS) runs on the server and only one program (application server) runs in the middle tier. Both programs will have to be capable of serving thousands of concurrent client connections. The problem of scalability of such architectures is a serious challenge for researchers and developers in the information technology industry.

3.4. Unifying applications and data in databases

There are several important aspects of the necessity to uniformly handle data and applications in database systems. Firstly, we need uniform, universal and flexible application models similar to data models used in software engineering and databases. One possibility is to describe each application as a set of business rules and their flows. The flows can be formulated and visualised in the form of flow diagrams, as we formulate and visualise relational data as tables. Presently, there are some systems that support workflows. We can imagine that data from these systems are interpreted and compiled into database triggers to be defined in an active database schema. Executing triggers by a database engine is much more efficient than executing them outside the system, so we can expect considerable gains in performance and flexibility of applications. However, to this end, we need to define thousands of triggers in one database, which for now is not feasible. It is estimated that scalability of three orders of magnitude is required, but job appears to be worth the while: for each data item we could define a trigger that would execute a (probably tiny) action for an application running over a federated system. Let us imagine for instance a stock application that promptly informs stakeholders of any change in the quotation of a given set of shares.

Secondly, it is of crucial importance for software engineering that the component-based approach for application development should be integrated with databases. It would be desirable if we could build any database application (maybe not only a database application) from ready-to-use components stored in a database. Obviously, such applications would be easily and efficiently stored and executed on demand within a database system. Presently, there is no widely accepted uniform and consistent model of software components. Different vendors promote their own component systems: CORBA, OLE, COM, DCOM, EJB, JINI *etc.*, so maintaining them in one database, although theoretically possible, does not seem neither reasonable nor comfortable for use. One, but far from satisfactory, currently available solution is the possibility of defining *User-Defined Types* (UDT), postulated in the SQL-99 standard. Actually, UDTs correspond to classes in the object-oriented paradigm. However, the standard does not treat UDTs as active components for different applications, but rather as mere types of data to be stored in a database. Another possibility stemming from the SQL-99 standard is coding software components in a database procedural language like SQL/PSM [5], primarily aimed at coding stored procedures. It seems that further development and refinement of languages of this class could be fruitful, particularly if it is accompanied by efforts towards optimisation of execution, similar to the optimisation of SQL queries.

One consequence of realizing a component-based model of database application development will be a need for new application development tools. The tools should be able to help a user to find necessary components, to integrate them into an application, to test the result in a testing environment and finally to move it into a production environment. It seems that the first step – finding an appropriate component – is the most difficult one, as it requires a user to specify his/her needs. This in turn requires a standard, precise component specification language, unless we want to make a user browse through numerous descriptions of components that might be useful (but almost always are not). We hope that such environments will appear together with more mature and advanced object-relational systems.

3.5. Integrating structured and semi-structured data

Traditional database systems, as well as the "next generation" systems [2], store data that have a well-defined structure known a priori, called a database schema. Data of this kind are *structured data.* The uncontrolled growth of the Web causes that more and more information resources (useful anyway) contain data that are irregular, incomplete or of complexity that can hardly be described by relational or even much richer object-oriented data models [8]. Data of this kind are called semi-structured data. A convenient tool for the description of such data is *eXtensible Markup Language* (XML), that allows for alternatives, optional constructs, multivalued attributes and other means that go beyond classical database constructs. As a consequence, contemporary advanced database systems are capable of storing XML data and querying them in a way similar to querying structured data (*i.e.* using a declarative query language).

It is expected that the Web content coded in XML will soon prevail the content produced in HTML. This will allow for automatic analysis of Web pages, including extraction of semantics from Web data, which is hardly attainable in the case of unstructured, HTML data. Additionally, XML documents can be self-descriptive in the sense that they may be accompanied by metadata formulated as *Data Type Definitions* (DTD) that actually play the role of database schemas. The *Document Object Model* (DOM) [9] attemps to standardize the form of XML published documents.

Presently, commercial DBMSs handle XML data via middle layers that translate them into a relational form. There is, however, no standard for processing XML data, although XPath [10] language is a good step in this direction. Actually, there is no clear vision of how to integrate XML Web resources and database technology. There are also a lot of important problems to be attacked in this area, like efficient processing of deeply nested hierarchical objects, typical for XML documents, the development of appropriate transaction models, devising new access methods (including methods of updating XML data), versioning and configuration management.

4. Conclusions

The amount of information we have at hand grows exponentially. The information is of varying quality and structure – from highly structured, "clean" data, appropriate for controlling devices or managing big enterprises, to irregular, imprecise and inconsistent data, distributed via Web sites of varied origin. Such situation creates new challenges for database systems, whose main task has always been to organize, store and make available data in a way most adequate for a given application. The main challenge can be formulated as follows: The priority is to develop database systems that would be able to collect, organize, store, analyse and make available all information resources of humankind in such a way that the information could be used on-line by anyone.

It is clear that this general goal is strongly related to Web technologies. Firstly, in the nearest future, most of our knowledge will be digitised (in the form of *digital libraries*) and made available globally through the Web. Secondly, the number of Internet users grows so fast that very soon most of the Earth's citizens (at least in the more developed areas) will become consumers of this knowledge, and consequently, clients of the knowledge repositories. Many of them will also become producers of knowledge, which will also require access to knowledge repositories. An ideal database system that we aim at should be able not only to respond to any queries formulated by any user of the global network, but also to anticipate users' queries and actively present useful information. As a matter of fact, we should strive to transform the huge storehouse of data called the Web into an integrated, intelligent, global information system based on an advanced and mature database technology.

References

[1] Cattell R G G and Barry D K (Eds) 2000 *The Object Data Standard: ODMG 3.0*, Morgan Kaufmann Pub.

[2] Stonebraker M and Brown P 1999 *Object-Relational DBMSs: Tracking the Next Great Wave*, Morgan Kaufmann Pub.

[3] *Extensible Markup Language (XML) 1.0*, Second Edition, W3C Recommendation, October 2000, www.w3.org

[4] Hackathorn R D 1999 *Web Farming for the Data Warehouse*, Morgan Kaufmann Pub. Inc.

[5] Gulutzan P and Pelzer T 1999 *SQL-99 Complete, Really*, R&D Books

[6] Bernstein P, Brodie M, Ceri S, De Witt D, Franklin M, Garcia-Molina H, Gray J, Held J, Hellerstein J, Jagadish H V, Lesk M, Maier D, Naughton J, Pirahesh H, Stonebraker M and Ullman J 2000 *The Asilomar Report on Database Research*, www.acm.org/sigmod/record

[7] *OMG Unified Modeling Language Specification, Version 1.4*, September 2001, www.omg.org

[8] Florescu D, Levy A and Mendelzon A 1998 *ACM SIGMOD Record* **27** (3) 59

[9] *Document Object Model (DOM) Level 2 Specifications*, November 2000, www.w3.org

[10] *XML Path Language (XPath) Version 1.0*, W3C Recommendation, November 1999, www.w3.org

Provision of Databases in the Poznan Supercomputing and Networking Center

Sławomir Niwiński[1], Iwona Pujanek[2] and Maciej Stroiński[1]

[1] *Poznan Supercomputing and Networking Center,
Noskowskiego 10, 61-704 Poznan, Poland,
{niwinski,stroins}@man.poznan.pl*

[2] *Poznan University of Technology, Main Library, Plac
Skłodowskiej-Curie 5, 60-965 Poznan, Poland,
iwona@ml.put.poznan.pl*

(Received 16 October 2002; revised manuscript received 13 January 2003)

Abstract: The article presents a concise report on the experience gained in the last three years in the scope of network provision of bibliographic databases by the Institute for Scientific Information (ISI) and full-text humanities and medical databases by EBSCO Publishing. The authors emphasise the importance and impact of the programme and the databases, co-financed by the State Committee for Scientific Research, on the initiation and continuation of organisational activities and efficient database access management. The paper contains a short review of the information resources presently available, including the titles of bibliographic and full-text databases, the scope of licences, subscription periods, and the volumes of archival resources. It provides statistics illustrating the distribution and extent of the bibliographic database usage by the scientific community, including, active institutional and individual, users and discusses the hardware and software used to provide the network database access services, the availability conditions, as well as the rules of license renewal and co-financing by the interested institutions. The report also deals with the access conditions and access abilities to the electronic versions of humanities, economics and medical databases offered by EBSCO Publishing. It is vital to show the structure of the service, the scope of subjects available in the databases, title estimation and the access to html, as well as image source types. Moreover, the report focuses on the modes of access to single database and to several of them simultaneously, as well as on multi-aspect searching with the use of the author and subject indexes and keywords.

Keywords: bibliographic databases, full-text databases, network access, EIFL Programme

1. Introduction

The activity of the Poznan Supercomputing and Networking Center (PSNC) is concentrated on five main areas:

- operating the POL-34/622 Polish National Scientific and Education Network as well as international connections,
- operating the Poznan Metropolitan Area Network – POZMAN,
- an HPC Centre,
- an System and Network Security Centre,
- R&D on new generation networks, grids and portals.

Since 1996 PSNC has maintained and provided users with access to bibliographic databases [1], and since 2000, to electronic versions of the full-text databases of EBSCO Publishing.

Literature databases containing basic data concerning publications in the various fields of science and general knowledge are a significant source of information used in scientific work. Apart from the basic bibliographic data on publications, these databases more and more frequently contain abstracts created by authors and publishing houses, as well as full-text articles, and also faithful images of publications with graphical elements and photographs. Information run in databases is usually organised in a specific way, *e.g.* as subject indexes. A vital criterion of database usefulness is the availability of proper searching mechanisms, both simple and advanced, one- and multi-criterial, but easy-to-use, intuitive and user-friendly.

Such access is offered by EBSCO Publishing within the Electronic Information for Libraries – EIFL Direct Project. It provides online access to literature databases, available at *http://search.epnet.com/*.

2. Bibliographic databases

2.1. Starting the database access service

At the end of 1996, the Poznan Supercomputing and Networking Center obtained network licenses for the following bibliographic databases for the scientific community:

1. **SCI/CDEA** (Science Citation Index, Compact Disc Edition with Abstracts);
2. **CCCD** (Current Contents on CD with Abstracts, Six Editions), containing 6 series: Engineering, Computing & Technology, Life Sciences, Clinical Medicine, Social & Behavioral Sciences, Agriculture, Biology & Environmental Sciences, Physical, Chemical & Earth Sciences;
3. **A&HCI/CDE** (Arts and Humanities Citation Index).

All databases have been purchased from one vendor – the Institute for Scientific Information (ISI) from Philadelphia. The licenses for the aforementioned databases are of network and regional character and, when purchased, referred to the whole of the Poznan scientific community, *i.e.* universities, Polish Academy of Sciences institutes and R&D institutions. Purchasing a one-year license of any database means that it remains the property of PSNC (and the whole community) also after that period and will continue to be available in the following years [2, 3].

On purchasing the license to use the bibliographic databases, a database server and access Info Ware CD/HD UltraNet Software were installed. The system makes it possible to reload information recorded on CD-ROMs to disc memory and make it available via a computer network, so that over 100 users can be managed at the same time, to simultaneously store and provide several databases on-line, to provide management and access via several network protocols (TCP/IP, IPX, NetBios) concurrently.

2.2. Financing

Three one-year licenses for bibliographic databases: Current Contents (CC), Science Citation Index (SCI) and Arts and Humanities Citation Index (AHCI) with the subscription period starting from 1997 were purchased for the funds granted in 1996 by the State Committee for Scientific Research. The subsidy covered 90% of the license price, with PSNC covering the remainder. The first licenses were purchased on preferential conditions within an ISI grant. To renew the license in 1998, a similar amount was granted by the Polish Foundation of Science Dissemination, which made possible the purchase of licenses for SCI and CC bases at an extra charge from PSNC (approximately 13% of the license value). The subsidy from the State Committee for Scientific Research (approximately 35% higher than in 1997) for the renewal of the SCI license made it possible to purchase all three databases (SCI, CC, AHCI) for the year 1999 and, additionally, the SCI database license for the year 1996 (used for 2 years at no cost) as well as the AHCI database license for the year 1998.

Starting from 2000, the State Committee for Scientific Research has subsidised only the SCI database license, up to 25% of its value. The Poznan scientific community have decided to pay the remaining 75% of the SCI database license value and to fully finance the Current Contents database on their own. The amount of the fee paid by the institutions interested in the access to database titles is based the total annual usage time. PSNC does not charge users for the costs of maintaining the service.

2.3. Users

In December 1999, the following scientific and R&D institutions of the Poznan community were actively using all the available databases: Academy of Economics, University of Medical Science, Agriculture University, Academy of Physical Education, Institute of Bioorganic Chemistry of the Polish Academy of Sciences (PAS), Institute of Dendrology (PAS), Institute of Energetics, Institute of Molecular Physics (PAS), Department of Human Genetics (PAS), Institute of Plant Genetics (PAS), Institute of Plant Protection, Institute of Natural Fibres, Poznan University of Technology, Adam Mickiewicz University, Industrial Institute of Agricultural Machines, and R&D Center of Machines and Special Devices.

2.4. Provision of bibliographic databases to the Poznan scientific community

The server of the bibliographic databases is available at *baza1.man.poznan.pl*.

Using the databases run on the server is possible after installing client software with access to the Internet. The software is free and available from the ftp server at *ftp://ftp.man.poznan.pl/pub/windows/ultranet/*.

Also, a www service containing detailed information on client software installation was prepared and made available at *http://www.man.poznan.pl/software/databases/*.

3. EBSCO Publishing literature databases

3.1. Subject characteristics, service structure and periodical rank

EBSCO Publishing offers access to two sets of literature databases: EBSCOhostWeb and EBSCO Medline, containing a total of 18 literature databases referring to: socio-economic sciences, the humanities, education, technology, business, information technology, bio-physics, bio-chemistry and medicine. The databases are of bibliographic-abstract character and comprise publications from over 8 000 periodicals, including over 6 000 in full text (in html, pdf, and xml) and represent such publishers as: Springer, Swets & Zietlinger Publishers, American Institute of Physics, Scandinavian University Press International Division, Blackwell Publishers, Taylor & Francis, B.C. Decker. Inc., Academy of Management, University of California Press, Arnold Publishers, Harvard Business School Publishing, Industrial & Labor Relations Review, Lawrence Erlbaum Associates, and many others. A detailed subject characteristics of the currently available databases follows [4]:

- **Academic Search Premier:** Full text of more than 3 460 scholarly publications covering academic areas of study including social sciences, humanities, education, computer sciences, engineering, language and linguistics, arts and literature, medical sciences and ethnic studies;
- **Business Source Premier:** Provides full text of over 2 800 scholarly business journals covering management, economics, finance, accounting, international business, *etc.*;
- **MasterFILE Premier:** Full text of over 1 900 periodicals, general reference, business, health;
- **Newspaper Source:** Full text of regional U.S. newspapers, international newspapers, newswires, newspaper columns, indexing and abstracts for national newspapers;
- **USP DI Volume II, Advice for the Patient:** Patient-oriented drug information in lay language;
- **Regional Business News:** Full text newswire database that incorporates business wires from all over the world, including A&G Information, Africa News Service, Inter Press Service, Resource News International, South American Business, M2 Communications, PR

Newswire, Business News Wire, Canadian Corporate News;

- **Health Source-Nursing/Academic Edition:** Nearly 580 full text scholarly journals focusing on many medical disciplines. Also featured are abstracts and indexing for over 615 journals;
- **Medline:** Authoritative medical information on medicine, nursing, dentistry, veterinary medicine, the health care system, pre-clinical sciences, *etc.* Created by the National Library of Medicine, allows to search abstracts from over 4 600 current biomedical journals;
- **ERIC:** Contains citations and abstracts from over 980 educational and education-related journals, as well as full text of more than 2 200 digests;
- **Health Source-Consumer Edition:** Health topics including the medical sciences, food sciences and nutrition, childcare, sports medicine and general health;
- **INSPEC:** Bibliographic information from the world's leading scientific and technical literature on physics, engineering, electronics, computers, and information technology;
- **Econlit:** Source of references to economic literature, includes journal articles, essays, research papers, books, dissertations, book reviews and working papers on accounting, consumer economics, monetary policy, labor, marketing, demographics, modeling, economic theory;
- **GeoRef:** Geoscience database on mineralogy and crystallography;
- **PsycINFO:** Citations and summaries of journal articles, book chapters, books, dissertations and technical reports, all in the field of psychology;
- **CINAHL:** Current nursing and allied health journals and publications;
- **Cochrane Database of Systematic Reviews (CDSR):** Reviews of the effects of healthcare;
- **Database of Abstracts of Reviews of Effectiveness:** Abstracts of published research reviews on the effects of health care from around the world;
- **Cochrane Controlled Trials Register:** Bibliographic listing of controlled trials in health care.

Over 2 000 titles from the full text periodicals are journals of high scientific rank, with high Journal Impact Factor. The EBSCOhostWeb service software provides users with access and the possibility to search one or several databases simultaneously.

3.2. Database organization, identification of information

An important element of data structure organisation in databases is a clear and detailed record of data identifying each literature item. It is vital to unambiguously and quickly identify the record position in a database and to construct effective algorithms of reviewing its content and rendering it accessible. It should be noted, that in the case of EBSCO Publishing databases such unambiguous data identification common for all databases has not been introduced or made possible. In some databases, the record structure is, however, the same or very similar and, therefore, simultaneous searching of such databases is really effective. The basic information identifying a publication on a database is a bibliographic record created for the sake of identification needs.

In some databases, the record may be more detailed and contain, *e.g.* abstracts by the publication's authors. The record concerning a publication source is composed of a number of elements and contains the following information: the title of the periodical, year, volume, issue number, the number of the first page of the issue, the number of pages of the original, the number and type of graphical elements (pictures, diagrams, charts *etc.*). The richer the content of the bibliographic record, the easier the identification of a publication on the database.

3.3. Access mechanisms, searching criteria, work modes

The interface provides a searching form in one of three available searching modes. The layout of each searching form depends on the context in which it was required. The layout is strictly defined if the search concerns one database only. The content of the form is defined mainly by fields identifying the bibliographic record. The following searching modes are possible: Basic Search, Guided Search and Expert Search. The Basic Search Mode is the simplest and requires neither experience in searching modes and strategies nor knowledge of the data structures. However, it is frequently used by experts, especially when they are beginning to learn a new subject, in order to narrow down the search scope. The other two modes of different complexity are chosen by more advanced users, as they help narrow down the searching scope more effectively and quickly to the expected literature list.

3.4. EIFL Direct Programme in Poland

On 15 May 2000, the Poznan Foundation of Scientific Libraries and Adam Mickiewicz University Library signed an agreement with the Open Society Institute (OSI) – Electronic Publishing Development Programe in Budapest (a branch of the George Soros Foundation) concerning co-ordination of the Electronic Information for Libraries (EIFL) Programme in Poland. The agreement assumes:

a) creating an open National Consortium of Libraries interested in access to electronic databases, mainly referring to the humanities, economic and medical sciences,
b) collecting funds of 50% of the license value, constituting the Consortium's own costs of the project, from all the Consortium participants,
c) acting to maintain and develop the Consortium,
d) subsidizing the project by the Soros Foundation.

On 31 August 2000, the Institute of Bioorganic Chemistry (PAS) – Poznan Supercomputing and Networking Center signed an agreement with the Poznan Foundation of Scientific Libraries and the Adam Mickiewicz University in order to create a National Consortium of Libraries. The consortium is supposed to provide authorised libraries in Poland with access to the electronic versions of EBSCO Publishing databases. PSNC has hoped to receive from the State Committee for Scientific Research a subsidy constituting 50% of the license value, and provided technology and organisation for the project realisation. Within the Consortium, PSNC is responsible for the installation, updating and maintenance of the local database copy (the Polish archive of EBSCO Publishing databases), as well as the development, introduction and maintainance of software providing access to the databases during the realisation period.

Information concerning the possibilities of network access to the electronic versions of EBSCO Publishing databases is available from company pages at *http://search.global.epnet.com* or *http://search.epnet.com/*, as well as from the pages of the Poznan Foundation of Scientific Libraries (*http://www.pfsl.poznan.pl/*).

3.5. Polish archive of EBSCO Publishing databases

Copies of two largest databases, Academic Search Premier and Business Source Premier, are available in the Polish archive of EBSCO Publishing databases. The copies do not contain Full Text Images. The Polish database archive is available for the participants of the Polish Consortium of Libraries at *http://ebsco.man.poznan.pl/*.

A Polish language graphic interface has been created, and a search mode similar to the Guided (formerly Advanced) Mode available in the EBSCO Publishing service has been implemented for the needs of the archive.

4. Conclusions

The process of organising new community services, their maintenance and further development is obviously complicated, expensive and time-consuming. It was undoubtedly the significant subsidy from the State Committee for Scientific Research in the first years of disseminating access to the bibliographic databases [5] and the electronic versions of EBSCO Publishing databases [6] that has greatly contributed to the success of the aforementioned undertaking. The current databases are mostly co-financed or fully financed by the interested institutions.

References

[1] Niwiński S and Stroiński M 1997 *Proc. "INFOBAZY'97 – Databases for Science"*, TASK Computer Center in Gdansk, Poland, pp. 348–352

[2] Niwiński S and Pujanek I 2001 *Advanced Methods in Accessing Bibliographic Database SCIENCE CITATION INDEX on CD with Abstracts*, Second Edition, Poznan Supercomputing and Networking Center, Report no. RA-002/2001

[3] Niwiński S and Pujanek I 2001 *Advanced Methods in Accessing Bibliographic Database CURRENT CONTENTS on CD with Abstracts*, Second Edition, Poznan Supercomputing and Networking Center, Report no. RA-001/2001

[4] Niwiński S 2001 *Polish Graphical Interface for the EBSCO Publishing Databases Archive Project and implementation*, Poznan Supercomputing and Networking Center, Report no. RA-005/2001

[5] Niwiński S and Stroiński M 2002 *Proc. "INFOBAZY'2002 – Databases for Science"*, TASK Computer Center in Gdansk, Poland, pp. 159–162

[6] Pujanek I and Niwiński S 2002 *Proc. "INFOBAZY 2002 – Databases for Science"*, TASK Computer Center in Gdansk, Poland, pp. 151–158

Development of INFOCAST: Information System for Foundry Industry

Grzegorz Dobrowolski[1], Robert Marcjan[1], Edward Nawarecki[1], Stanisława Kluska-Nawarecka[2], Joanna Dziaduś[3] and Teresa Wójcik[3]

[1] *Department of Computer Science, University of Mining and Metallurgy, Mickiewicza 30, 30-059 Cracow, Poland, grzela@agh.edu.pl*

[2] *Department of Industrial Informatics, University of Mining and Metallurgy, Mickiewicza 30, 30-059 Cracow, Poland, nawar@iod.krakow.pl*

[3] *Foundry Research Institute, Zakopianska 73, 30-418 Cracow, Poland, jdziadus@iod.krakow.pl*

(Received 30 November 2002; revised manuscript received 18 March 2003)

Abstract: The article presents the current state of development of INFOCAST – an information decision system supporting technological problems of the foundry industry. Two aspects of the system are related: enrichment of its information and knowledge resources and changes in architecture, both oriented towards improvement of its applicability. The reported stage of development of INFOCAST is characterized by a close integration of its data and knowledge bases by means of the agent-based technology. Due to decentralization, the knowledge and information can be used not only in research and design activities, but also in the exploitation of technological processes.

Keywords: multi-agent systems, expert systems, databases, foundry industry, diagnosis

1. Introduction

The information resources available in the databases currently operating within the INFOCAST system [1, 2] are of great importance for research work and projects, as well as for production planning and marketing in the field of the foundry industry. The information is the property of the Foundry Research Institute in Cracow, who make it available and keep it up-to-date. The computer implementation of the system has been developed by Department of Computer Science of University of Mining and Metallurgy, also in Cracow. The core of the INFOCAST system consists of four currently operating and continually updated data and knowledge bases: SINTE, NORCAST, CASTSTOP and CAST-EXPERT.

Computerization of the bases of the INFOCAST system was carried out within the framework of a research project sponsored by KBN (State Committee of Scientific Research) in the years 1997–2001, in two stages. Information stored in its bases has been collected since 1977.

During the last stage, in 1999–2001, two kinds of activities were carried out:

- One was oriented towards the implementation of a new version of the INFOCAST system on a supercomputer at ACK CYFRONET AGH Centre, based on DBMS ORACLE. An immediate effect of this activity is that the system is easily available on the net and operates more efficiently.
- The other was related to designing and building a decentralized version of INFOCAST, based on the agent technology. There are programming tools (an agent software platform along with the relevant programmer's environment) designed specifically for the construction of decentralized information decision systems distributed over the net.

All the time, enrichment of INFOCAST's information resources has continued.

The decentralized version of the system (its agents), like CASTEXPERT, takes advantage of rule-based knowledge representation. Its implementation makes use of the JAVA language.

INFOCAST in its decentralized structure is expected to offer direct (network) access to foundry mills along with the possibility to utilize information coming from the currently running production processes.

The article is organised as follows. Section 2 is a recapitulation of the current state of resources of INFOCAST and a brief introduction to the system. Section 3 presents basic information regarding the idea of a decentralized version of the system. An outline of its overall architecture, designed according to the agent-based technology, is given. Information about the general structure of an agent and the inter-agent communication platform completes the picture.

2. Utilitarian characteristic of INFOCAST

2.1. Field of application

The system under consideration has been designed to serve as a multi-purpose tool of technical assistance, an aid in designing new technologies and quality assurance for foundry products. The following can be mentioned as typical tasks solved by the system:

1. assistance in reference literature from the field of foundry practice,
2. analyses in the field of manufacturing technologies,

3. expertise in the field of casting defects diagnosis,
4. analyses for casting products – a module of knowledge to be designed in the future.

It is assumed that the services provided by the system can cover the analyses ordered by:

- research and development centers and specialized laboratories,
- design and trade offices,
- production plants and supervising staff,
- potential users of produced castings.

The system, which is accessible[1] free of charge to the domestic research institutions, serves as a basis for many research projects, including MSc and PhD theses. At present, it is the only Internet source of information on foundry practice in Poland.

2.2. Information and knowledge resources

INFOCAST is composed of four data and knowledge bases (SINTE, NORCAST, CASTSTOP, CASTEXPERT) [1, 2]. They contain a wide range of information of foreign and Polish literature (the SINTE database), European, international, Polish and EU standards (the NORCAST database), standard grades of cast alloys (the CASTSTOP database) and casting defects with reasons of their occurrence (the CASTEXPERT system with its knowledge base). The databases are relational.

SINTE thematically covers the following foundry-related problems: metallurgy and metal science of cast alloys; heat treatment of castings; iron castings; steel castings; non-ferrous metals castings; metal matrix composite castings; melting processes and melting installations; technological processes of moulding sand preparation; mould and core making; casting technologies; environmental protection; fettling and finishing of castings; use of computers to aid foundry production; foundry machines and equipment; mechanization and automation; quality control and quality systems; marketing; management; organization and cost of production.

Information included in the database is taken from all the leading Polish and international (American, English, French, German, Czech, Slovenian, Russian, Portuguese, Swedish) journals issued in the years 1977–2001, available in the Library of the Foundry Research Institute.

There are about 32 000 records now, while an average 1 000 is added each year. Two special ways of searching are available: according to descriptors using the thesaurus or using a dedicated classification system.

NORCAST comprises 4 058 records containing up-to-date information on foundry standards: Polish, European, international, and foreign national from selected countries (UK, Germany, France, Norway, Finland, Sweden, Italy, USA), as well as information on PN, ISO and EN standards related to a variety of problems concerning quality assurance systems, environmental management and certification.

Recorded aspects of the standards include: general issues, charge materials, moulding materials and methods of their testing, auxiliary materials, machines and equipment, castings, examination of cast alloys and testing of castings, occupational health and safety, quality assurance systems and certification.

CASTSTOP database gives information on Polish, European and foreign standard grades of cast alloys, including their chemical composition, mechanical properties and heat treatment, augmented whenever possible with some additional information. At present, its 1 098 records contain information taken from the international standards (ISO), European standards (EN), French (NF), German (DIN), Norwegian (NS), Swedish (SS, SIS), British (BS) and Italian standards (UNI).

CASTEXPERT is an expert system for diagnosing casting defects. Its user is aided by a knowledge base in the form of decision rules with an appropriate inference engine and a database of photographs of castings defects, as an exemplification. Based on the knowledge comprised (7 200 rules) and information acquired from process engineers (by way of dialogue), CASTEXPERT supports identification of possible reasons of occurrence of defects in castings with an indication of actions to be undertaken to eliminate these defects in the future. The current version of the system embraces defects of castings made of grey iron, ductile iron and non-ferrous metal alloys, as well as defects in steel castings.

CASTEXPERT data may be also regarded as an important source of information on casting defects that may be used in quality control of

[1] All the bases of INFOCAST system are available at the following address: `http://czapla.iod.krakow.pl/infocast/`

castings, or as reference material for training engineers and students.

3. Agent architecture of the decentralized INFOCAST

Apparently, the idea of multi-agent systems [3–5] is an extension or modification of the idea of decentralized systems, so that they may be even regarded as synonymous.

A typical system of this kind is composed of a set of subsystems/nodes (agents) that cooperate with each other, linked to form an organizational structure. The links are dependent on the functional characteristics of subsystems and can be created in a dynamic way in order to attain a global goal (of the whole system). Because agents are assumed to have some autonomy, the process of searching for a solution is decentralized. The potential of this system (its functionality) result from the functions of its individual agents and a certain *added value*, which is created in their cooperation.

The main nodes or subsystems of the decentralized version of INFOCAST are its computerized information or knowledge sources. Assuming the future development of INFOCAST, it is possible to enumerate many possible types of sources:

- Relational databases
  Information is accessed using an appropriate query language (SQL). Handling of information (making a query) requires knowledge of the database structure.
- Knowledge bases of reasoning systems
  Expert systems contain fragments of the domain knowledge. In the case described here, these are systems with rule-based knowledge representation. Communication with systems of this type consists in exchange of facts and rules.
- Experts – humans
  It is assumed that an expert is a source of the domain knowledge. Communication is in an off-line mode. Adequate understanding of questions addressed to an expert should be assured by the system.
- Other sources of information
  It is allowed to use other sources of information (*e.g.* other information systems or control and measuring instruments). Including such a source of information into the system requires the creation of special communication software.

In the solution described here, access to the various types of knowledge and information sources mentioned above is through agents assigned to these sources.

It should be mentioned here that the goal is not to create a decentralized diagnostic system from the very beginning, but to design a technology which would enable utilization of the existing information systems available via the internet and their arrangement into a system capable of solving complex tasks.

3.1. Agents and their functions

Given the above described application field of INFOCAST and the idea of its operation in a decentralized regime, the main functions which are to be performed by agents, can be defined as follows:

- reasoning based on local information resources,
- communication with other agents,
- communication with humans acting within the system (user, expert),
- making available various sources of knowledge created using various technologies,
- creation of an information network through identification of agents necessary to employ to solve a specific problem,
- assuring the conformity of knowledge components used in the diagnostic process,
- integration of information, synthesis and propagation of results, and arbitrage in the case of conflict.

A full set of agents proposed for the system architecture is shown schematically in Figure 1.

AG-DB: an agent for a source of knowledge of the database type – possesses a description of the source of knowledge (definition of the database, characteristic of the tables) and is capable of translating the contents of KQML messages [6] into the corresponding phrases (queries) written in SQL. A majority of the systems of this type enable communication through a JDBC/ODBC interface.

AG-SE: an agent for a source of knowledge of the expert system type – both the expert system and the agent use the same representation of information (facts and rules written in the JESS language [7]). Interactions consist in direct start ups of the reasoning process and appropriate data exchanges.

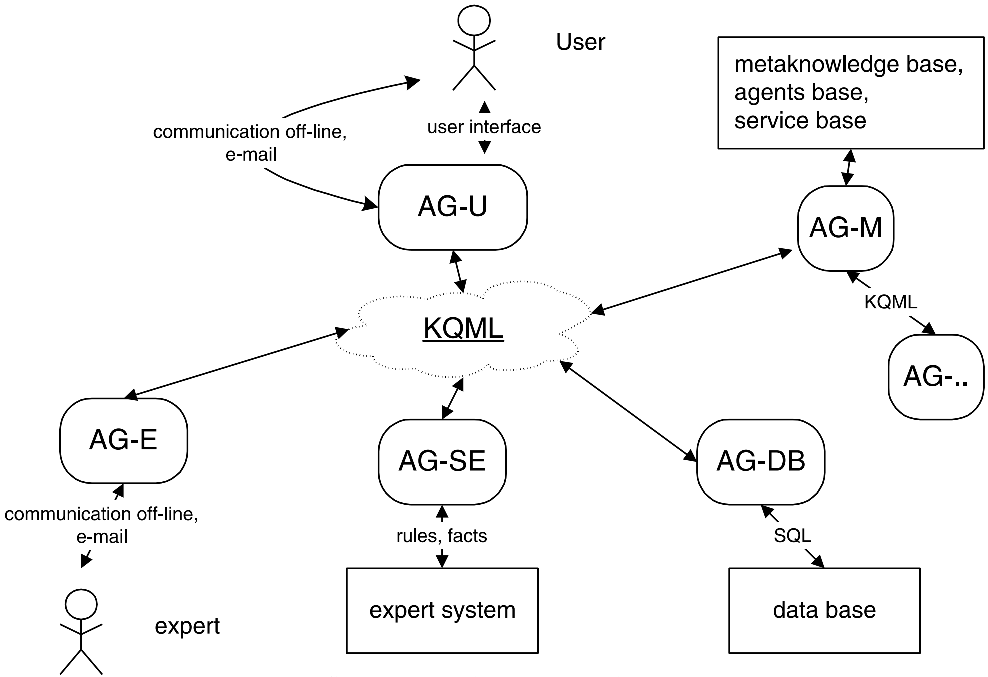

Figure 1. General scheme of decentralised information system

AG-E: an agent for a source of knowledge of the expert type – assists an expert in his communication with the system, receives messages in KQML, generates their HTML forms and sends them by electronic mail to the expert's address. When a replay comes, the contents is translated back to KQML.

AG-U: the user's agent – is responsible for communication with the user from the moment an analysis is initiated, through generation of auxiliary queries, until results are presented. These functions can be executed via the standard windows interface (based on a WWW browser), or by means of electronic mail. In the latter case, communication is carried out in the same way as it is with an expert.

AG-M: an agent managing meta-knowledge. This is also the agent allowing access to the source of knowledge/information, but of special functions in the system. There are:

- a database of the agents – information about the system structure (identification, addresses, *etc.*),
- ontologies – sets of notions (facts) together with relations between them which are used by the agents; reasoning in the system is made according to the relevant ontology (in foundry technology, the ontology is based on a thesaurus developed by the Foundry Research Institute),
- a database of services offered individually by the agents – information which can be provided by the given source of knowledge (the agent that operates it).

The system structure described above and the organization of access to the sources of information enable the process of reasoning (preparing an analysis) to be carried out irrespective of the structure of a given source (a coherent set of notions/facts, uniform methods of communication). They also make it an open system and create possibilities of easy adaptation of new sources of knowledge (extension of the system domain, new architectures of knowledge sources, *etc.*).

3.2. Agent as an inference engine

A sketch of the internal structure of an agent and its functional blocks are shown in Figure 2. This is a general agent architecture in that it involves all the modules (elements) which can appear among the system agents, while in specific cases some of them may be unnecessary (restricted configuration is applied).

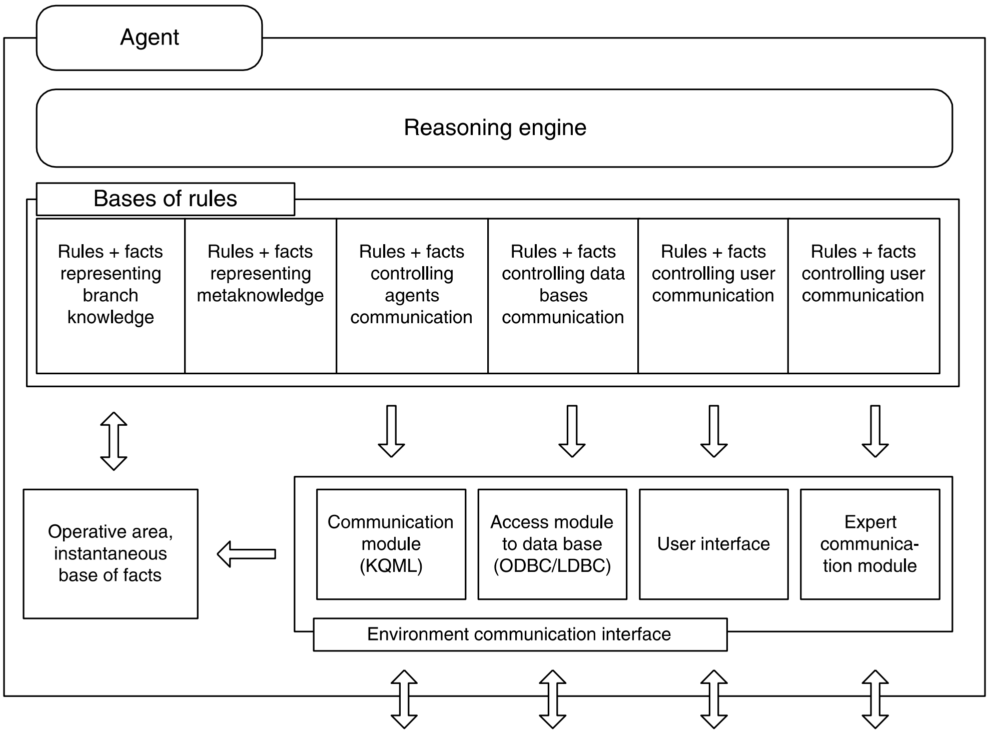
Agent
Reasoning engine
Bases of rules
Rules + facts representing branch knowledge
Rules + facts representing metaknowledge
Rules + facts controlling agents communication
Rules + facts controlling data bases communication
Rules + facts controlling user communication
Rules + facts controlling user communication
Operative area, instantaneous base of facts
Communication module (KQML)
Access module to data base (ODBC/LDBC)
User interface
Expert communica-tion module
Environment communication interface

Figure 2. General architecture of an agent

The agent's core is an inference engine that accepts a set of rules in the form of *premise* ⟶ *action* and facts that can be divided into several categories depending on what these facts refer to [8]. This results in differentiation of the functional blocks (modules) of an agent. Each block consists of a sub-base of rules/facts and an appropriate module, which ensures communication with the environment (*i.e.* with other agents or sources of knowledge). Communication among an agent's functional blocks is achieved through an instantaneous base of facts (an operational area or agenda, in the terminology of rule-based systems). Appearance of new facts in this area activates the rules for which the premises become true. Thus, modifications are introduced into the instantaneous base of facts, *i.e.* new facts appear, which in turn activate the rules of other modules. The appropriate communication modules are responsible for the exchange messages with the environment [5, 9].

3.3. Agent communication platform

The main task of the platform is to provide an appropriate communication infrastructure and organize agents into a coherently functioning system.

The platform implementation uses the JATLite (Java Agent Template) [10] package of software. JATLite offers a library, written in Java, comprising classes which enable communication at different levels of abstraction, defined as communication layers and protocols (including communication using the KQML protocol/language).

The JATLite software also serves as a simple communication platform. Its services are represented by the specialized agent called AMR (Agent Message Router). The main tasks of the agent are given below:

- Managing the system structure (*i.e.* the name space and communication addresses). Before an individual agent starts functioning in the system, it registers with AMR. Then, using its name and password, the agent can connect and disconnect with the system, and exchange information with other agents operating on different computers until it registers off the platform. AMR stores and manages physical addresses of agents.

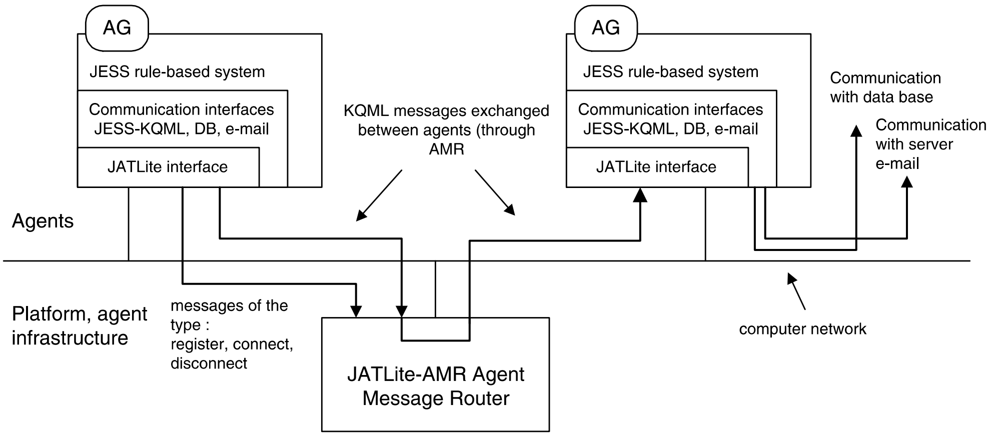
AG
JESS rule-based system
Communication interfaces JESS-KQML, DB, e-mail
JATLite interface
AG
JESS rule-based system
Communication interfaces JESS-KQML, DB, e-mail
JATLite interface
KQML messages exchanged between agents (through AMR
Communication with data base
Communication with server e-mail
Agents
Platform, agent infrastructure
messages of the type : register, connect, disconnect
JATLite-AMR Agent Message Router
computer network

Figure 3. General concept of the agent communication platform

- Acting as an intermediate link between the agents who participate in the system. The agents do not communicate directly with each other, but use AMR to this purpose. Using the JATLite platform, each agent forms a single connection to AMR and uses this connection for exchanging messages with other registered agents. If an agent is disconnected or damaged (stops functioning), the messages sent to it are stored by AMR until being re-connected.

General schematic presentation of the platform functionality as well as types of messages and their means of transportation are shown in Figure 3.

4. Final remarks

The article describes information decision system INFOCAST dedicated to aiding the foundry technologies. The system has been installed on the ACK CYFRONET AGH supercomputer using an ORACLE data base management system and is available through the Internet to users from Poland and abroad. It offers the knowledge and information generated and collected at the Foundry Research Institute in Cracow. The system has proven its usefulness in many research, production planning and marketing projects.

The decentralized version of INFOCAST has been designed to semantically integrate the resources of the Foundry Research Institute with similar resources of other research centers and the knowledge of individual experts and real data collected by industrial enterprises. It is assumed that, based on proper network infrastructure and appropriate computer tools, the system makes it possible even for quite complex analyses to be done automatically without a deeper intervention of the user.

Full capabilities of the system will be gained when direct network connections with industrial enterprises (and their appropriate departments) have been established. Such target situation should enable, on one hand, making local analyses right on the spot, utilizing the general knowledge collected in the INFOCAST system. On the other hand, the communication links should intensify further enrichment of this knowledge through generalization of experience acquired within the industry.

The idea of building a decentralized information system using the agent-based technology is straightforward and effective. Invention and application of a unified architecture of the agents, using rule-based representation of their behavior, has given a similarly positive effect.

References

[1] Kluska-Nawarecka S, Dobrowolski G and Marcjan R 2001 *Acta Metallurgica Slovaca* **7** 441

[2] Kluska-Nawarecka S 1996 *J. Mat. Sci. and Technol.* **4** 11

[3] Ferber J 1999 *Multi-Agent Systems*, Addison-Wesley

[4] Weiss G (Ed.) 1999 *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press

[5] Dobrowolski G 1998 *Management and Control of Production and Logistics*, MCPL'97 IFAC/PERGAMON (Binder Z, Hirsch B, Aguilera L, Eds) **2** 393

[6] Finin T, Fritzon R, McKay D and McEntire R 1994 *Proc. 13th Int. Workshop on Distributed Artificial Intelligence*, Seatle, WA, pp. 126–136

[7] Friedman-Hill E 1999 *JESS – the Java Expert System Shell* http://herzberg.ca.sandia.gov/jess/

[8] Negrete Martinez J and Gonzalez Perez P 1998 *Expert Systems with Applications* **14** 102

[9] Dobrowolski G and Nawarecki E 2001 *Management and Control of Production and Logistics 2000*, PERGAMON (Binder Z, Ed.), pp. 445–450

[10] Jeon H, Petrie C and Cutkosky M R 2000 *IEEE Internet Computing* **4** 89

Diagnosing Skin Melanoma: Current versus Future Directions

Zdzisław S. Hippe[1], Stanisław Bajcar[2], Piotr Blajdo[1], Jan P. Grzymala-Busse[3], Jerzy W. Grzymala-Busse[3], Maksymilian Knap[1], Wiesław Paja[1] and Mariusz Wrzesień[1]

[1] *Department of Expert Systems and Artificial Intelligence, University of Information Technology and Management, Sucharskiego 2, 35-225 Rzeszow, Poland, zhippe@wenus.wsiz.rzeszow.pl*

[2] *Regional Dermatology Center, Warzywna 3, 35-310 Rzeszow, Poland,*

[3] *Department of Electrical Engineering and Computer Science, University of Kansas Lawrence, KS 66045, USA*

(Received 18 January 2003)

Abstract: A new database containing 410 cases of *nevi pigmentosi*, in four categories: *benign nevus*, *blue nevus*, *suspicious nevus* and *melanoma malignant*, carefully verified by histopathology, is described. The database is entirely different from the base presented previously, and can be readily used for research based on the so-called constructive induction in machine learning. To achieve this, the database features a different set of thirteen descriptive attributes, with a fourteenth additional attribute computed by applying values of the remaining thirteen attributes. In addition, a new program environment for the validation of computer-assisted diagnosis of melanoma, is briefly discussed. Finally, results are presented on determining optimal coefficients for the well-known ABCD formula, useful for melanoma diagnosis.

Keywords: melanoma, TDS, machine learning in diagnosis of

1. Introduction

In recent papers [1, 2], we have presented the results of experiments on new samples relating to changes in skin melanoma, using machine learning with the idea of generating a model of learning to help identify and classify cases of skin melanoma. Skin melanoma may be a symptom of serious skin diseases, or even cancer, which has a high mortality rate. The numbers of victims of this type are rising because of the high levels of ultraviolet radiation entering the atmosphere and the increasingly thin ozone layer [3]. Anonymous data sets pertaining to cases of skin cancer have been collected by the Regional Dermatology Center in Rzeszow, Poland [4]. This data set of cases has been analyzed and expert systems based upon it have been implemented at the Department of Expert Systems and Artificial Intelligence, University of Information Technology and Management in Rzeszow, Poland. The first version of the data set has been analyzed in a paper presented at the INFOBAZY'99 conference [1, 2]. The actual production version contains (i) new internal structures and (ii) an increased number of registered cases (from 250 to 410). Regarding (i), the data set information is stored in 13 attributes that are regularly used in dermatology for typical analysis of skin-based melanoma. In the context of these attributes, we calculate the **TDS** (Total Dermatoscopy Score) indicator [5]. The underlying idea of our experiments was to prepare our sample in both Polish and English and use specialized software algorithms in the verification of its accuracy, as well as generate learning models to help diagnose diseases. The data sets were acquired in studies taking place simultaneously in Rzeszow, Poland (University of Information Technology and Management) and the United States (University of Kansas, Lawrence, Kansas). In the following sections, the data sets have its statistical analysis and its machine learning results discussed. An earlier version of this paper was presented at the 3rd National Conference INFOBAZY'2002, Gdansk, Poland, June 24–26, 2002 [6].

2. Statistical analysis of the data sets

The attributes used in deducing diagnoses of melanoma have been broken down into 5 categories: ⟨Asymmetry⟩, ⟨Border⟩, ⟨Color⟩, ⟨Diversity⟩ and ⟨TDS⟩. The ⟨Asymmetry⟩ parameter can have the following values: symmetrical, single-axis asymmetry and dual-axis asymmetry. ⟨Border⟩ is a numerical attribute with discrete values between 0 and 8. The next two categories, ⟨Color⟩ and ⟨Diversity⟩, have symbolic values. ⟨Color⟩ can have six allowed values: black, blue, light brown, dark brown, red, and white. Likewise, ⟨Structure⟩ has five possible values: pigment dots, pigment globules, pigment network, structureless areas and branched

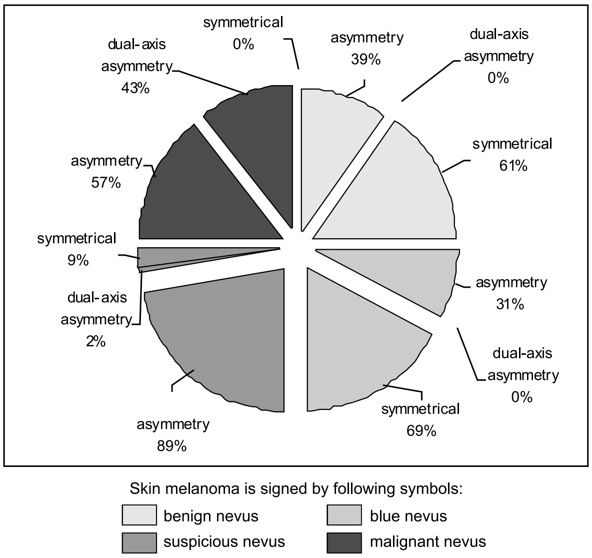

Figure 1. Appearance of the ⟨**Assymetry**⟩ attribute in each decision class

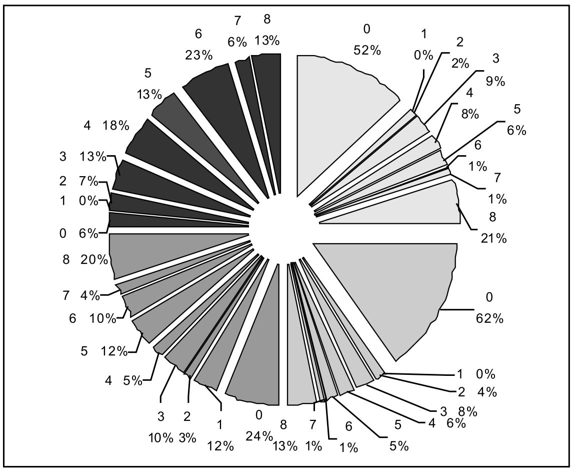

Figure 2. Appearance of the ⟨**Border**⟩ attribute in each decision class (symbols as per Figure 1)

streaks. In all these cases, the attributes pertaining to pigments and their diversity are Boolean and state the presence (1) or lack (0) of an attribute. Thus, every entry from the data set of anonymous patients is characterized by 13 attributes. For computing the fourteenth attribute, called **TDS**, the other 13 attributes are used, so that the **TDS** attribute is obtained by constructive induction [7]. The **TDS** indicator is computed using the following formula:

$$\mathbf{TDS} = 1.3\cdot\langle\text{Asymmetry}\rangle + 0.1\cdot\langle\text{Border}\rangle + \\ + 0.5\cdot\langle\text{Color}\rangle + 0.5\cdot\langle\text{Diversity}\rangle, \quad (1)$$

where the values for ⟨Asymmetry⟩ are as follows: symmetrical equals 0, single-axis asymmetry equals 1, and dual-axis asymmetry equals 2. ⟨Color⟩ represents the sum of represented pigment colors, whereas ⟨Diversity⟩ is the sum of the five represented diversity attributes. The accuracy of the calculated **TDS** plays a key role in generating a machine learning model using the concept induction system, and its correctness has been verified using an Excel spreadsheet calculation [8], which in turn allowed checking the individual work of specialist doctors. In this way,

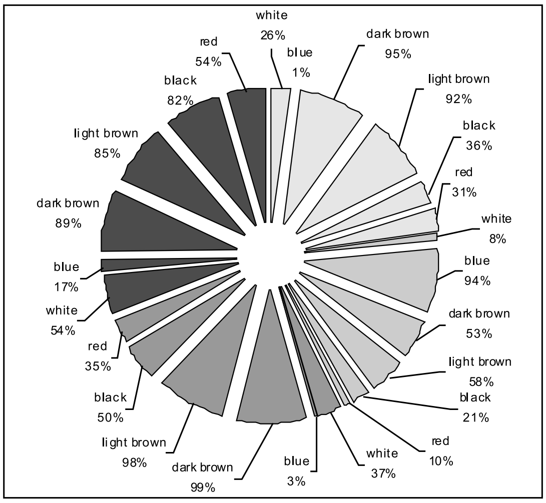

Figure 3. Appearance of the ⟨**Color**⟩ attribute in each decision class (symbols as per Figure 1)

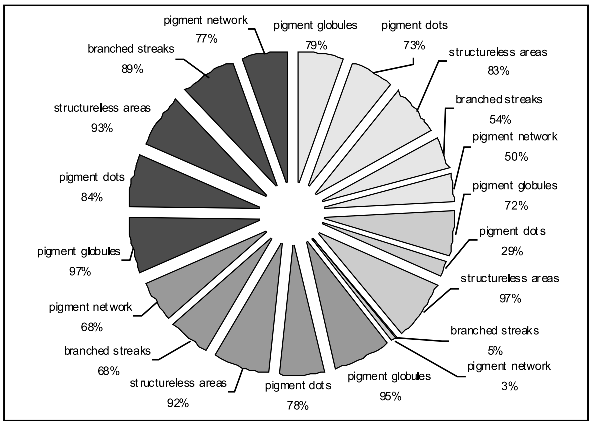

Figure 4. Appearance of the ⟨**Diversity**⟩ attribute in each decision class (symbols as per Figure 1)

we created a data set without any errors. Statistical analysis using the aforemtioned tools can be seen in Figures 1–4.

3. Using machine learning programs for modelling

The working data set was used for testing the machine learning model to help identify and diagnose changes in skin melanocytes. This field used the following program modules: ***Rule*SEEKER** (used to create rules), ***Tree*SEEKER** (generates quasi-optimal decision trees), ***Affinity*SEEKER** (seeks the similarities for diagnosing a patient with known results in the database), ***Plane*SEEKER** (searches for optimal decision planes based upon known attributes) and ***Score*SEEKER** (rates machine learning models generated by the former program modules). Due to restrictions on this paper's size, it is unfortunately not possible to elaborate on the details of all of the program modules. Instead, we shall concentrate on ***Score*SEEKER**, which is the most pertinent to the discussions of this conference, and which works (on raw data sets) on N-series data sets

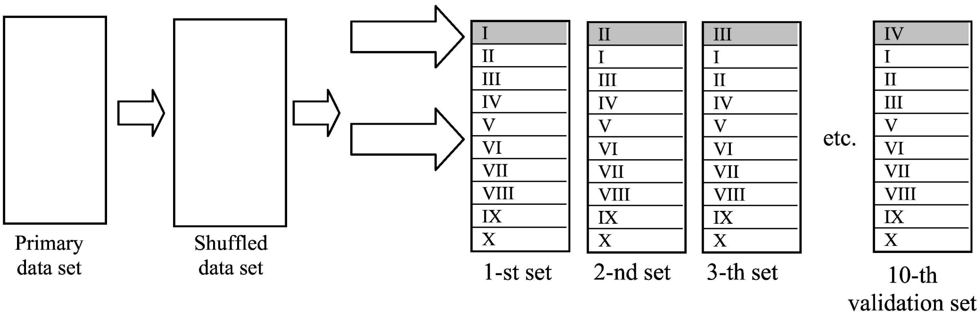

Figure 5. Illustration of the operation of the computer program system ***Score*SEEKER**. In the first step, records in the database are shuffled, and then 10 pairs of databases are created. In each of these, pairs 9/10 are used to generate a learning model and 1/10 – to test it

Table 1. Optimal coefficients for **TDS**

Attribute	Data set with 276 cases	Data set with 410 cases	
	Agglomerative discretization	Agglomerative discretization	Divisive discretization
Asymmetry	0.6	0.9	1.3
Border	0.1	0.14	0.11
Color black	0.5	0.5	0.5
Color blue	0.4	0.5	0.4
Color dark brown	0.5	0.3	0.4
Color light brown	0.5	0.4	0.5
Color red	0.4	0.5	0.5
Color white	0.4	0.5	0.4
Diversity pigment dots	0.5	0.5	0.4
Diversity pigment globules	0.6	0.5	0.5
Diversity pigment network	0.5	0.5	0.5
Diversity structureless areas	0.5	0.5	0.5
Diversity branched streaks	0.5	0.5	0.4

Table 2. Error rates in %

Data set with	Data set with 276 cases	Data set with 410 cases	
	Agglomerative discretization	Agglomerative discretization	Divisive discretization
original **TDS**	10.21	4.38	3.50
optimal **TDS**	6.04	4.51	3.63
no **TDS**	13.73	13.82	13.49

Table 3. Standard deviations in %

Data set with	Data set with 276 cases	Data set with 410 cases	
	Agglomerative discretization	Agglomerative discretization	Divisive discretization
original **TDS**	0.99	0.74	0.48
optimal **TDS**	0.84	0.78	0.49
no **TDS**	1.31	1.08	1.06

with a specific data structure (Figure 5) allowing moving current methods [9] of grading machine learning models.

4. Optimization of TDS

Both data sets, the old one with 250 training cases and additional 26 testing cases, and the new one with 410 cases, have been examined at the University of Kansas in Lawrence, Kansas. One of aims of the research conducted there was optimization of the ABCD formula to compute **TDS** or, more precisely, optimization of the 13 coefficients of Equation (1). The criterion of optimization was the minimization of the error rate in diagnosis of melanoma. The main problem is discrete optimization, which is known to be difficult and time consuming. Results of experiments on both data sets, described in [10] and [11], are presented in Tables 1, 2, and 3.

The error rate presented in Table 1 was computed using randomized ten-fold cross validation, in which, for every set of coefficients, experiments were repeated 30 times using different reshuffling of the original data set for each process of ten-fold cross validation. The optimal coefficients, presented in Table 3, were searched for using fixed (*i.e.* not randomized) ten-fold cross validation. However, the error rate of melanoma diagnosis for the final choice of optimal coefficients was verified with randomized ten-fold cross validation.

5. Conclusions

With 276 cases in the data set, there are significant differences between the original and the optimal choice of coefficients for **TDS**, at a 95% confidence level. Furthermore, with the same 95% confidence, diagnosis with **TDS** determined with any choice of coefficients (original or optimal), with 276 or 410 cases, is better than diagnosis in which **TDS** has not been used. The difference between error rates for diagnoses of melanoma using different discretization methods is not significant (with 95% confidence).

With 276 cases, the error rates for diagnosis using **TDS** are significantly higher than the rates for using **TDS** with 410 cases. This is due to better rule sets induced from the more representative data set of 410 cases. Also, with 276 cases, the error rate using optimal coefficients for **TDS** is significantly lower than the error rate using original coefficients for computing **TDS**. On the other hand, with 410 cases, the difference between the error rate for original and optimal coefficients for computing **TDS** are insignificant, always with 95% confidence.

In future experiments, with further increase in the number of registered cases, a hierarchy of importance of values may be created, based upon the described attributes. This would predictably allow more objective diagnosis and classification of cases.

Acknowledgements

This research has been partially supported by the State Committee for Research (KBN) of the Republic of Poland under grant no. 7 T11E 030 21.

References

[1] Hippe Z S 1999 *Proc. "INFOBAZY'99 – Databases for Science"*, TASK Computer Center in Gdansk, Poland, pp. 120–124

[2] Hippe Z S 1999 *TASK Quart.* **3** (4) 483

[3] Riegel D S, Friedman R J, Kopf A, Weltman R, Prideau P G, Safai B, Lebwohl M G, Elizeri Y, Torre D P and Bonford T T 1986 *J. Am. Acad. Dermatol.* **14** 857

[4] Bajcar S and Hippe Z S 1999 *Proc. Telemedicine Conference*, Lodz, Poland, pp. 175–178

[5] Braun-Falco O, Stolz W, Bilek P, Merkle T and Landthaler M 1990 *Hautartzt* **40** 131

[6] Hippe Z S, Grzymala-Busse J W, Bajcar S, Blajdo P, Knap M, Paja W and Wrzesień M 2002 *Proc. "INFOBAZY'2002 – Databases for Science"*, TASK Computer Center in Gdansk, Poland, pp. 51–55 (in Polish)

[7] Michalski R S, Bratko I and Kubat M 1998 *Machine Learning and Data Mining. Methods and Applications*, John Wiley and Sons LTD., Chichester

[8] Hippe Z S and *et al.* 2001 *Research on Methods of Computer-aided Diagnosis of Melanocytic Lesions and Melanoma*, Research Report, Grant KBN no. 7 T11E 030 21, Poland (in Polish)

[9] Weiss S and Kulikowski C A 1991 *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 17–49

[10] Grzymala-Busse J P, Grzymala-Busse J W and Hippe Z S 2001 *Proc. 25th Anniversary Annual International Computer Software and Applications Conference COMPSAC 2001*, Chicago, IL, pp. 615–620

[11] Grzymala-Busse J W, Hippe Z 2002 *Rough Sets and Current Trends in Computing* (Alpigini J J, Peters J F, Skowron A and Zhong N, Eds.), Springer-Verlag, Heidelberg, pp. 538–545

Detecting Approximately Duplicate Bibliographic Records with Text Algorithms: Experience of Creating a Union Catalogue of Libraries at the Warsaw University of Technology

Grzegorz Płoszajski

Warsaw University of Technology, Main Library, Faculty of Electronics and Information Technology, Politechniki 1, 00-661 Warsaw, Poland, g.ploszajski@ia.pw.edu.pl

(Received 5 November 2002; revised manuscript received 24 March 2003)

Abstract: The paper describes a fault-tolerant method of selecting duplicate bibliographic records in catalogues. The method is based on the use of text algorithms; decisions are suggested to librarians who make the final decision. The method was applied to four library catalogues at the Warsaw University of Technology which were compared with the catalogue of the main library. Process of joining catalogues was conducted differently for non-duplicate records and for duplicate ones. Thanks to this method, a significant portion of records in the catalogues of the joining libraries had been found to be duplicate before the catalogues were added. The algorithms proved helpful in assuring high quality of information.

Keywords: duplicate record resolution, n-grams, text algorithms

1. Introduction

Information in library catalogues should be unique: all identical books should be referred to by one and only one bibliographic record. A union catalogue collects information from catalogues of a group of libraries. To avoid duplicate bibliographic records in a union catalogue such records should be found before or during the process of the joining catalogues of the participating libraries [1, 2].

Selecting duplicate bibliographic records which are identical is an easy task. There is, however, a less straightforward problem of finding duplicate records which actually refer to the same books, but are not identical. The differences between such approximately duplicate records occur due to errors in typing as well as due to habits and experience of individual librarians. In most cases, these differences are small and records are "similar" in some way.

The task described herein was to compare catalogues and find duplicate bibliographic records, either identical or having minor differences. It is important especially in the case of union catalogues [1, 2], but also in the case of multi-database searching [3], where elimination of duplicate records significantly improves data quality. This task can be facilitated by means and procedures known as "duplicate record resolution".

2. The method of comparing bibliographic records: the concept

The comparison of bibliographic records is based on the comparison of corresponding fields and subfields. In the present project the scope of comparison of records was limited to the following elements of bibliographic information:

- ISBN – field 020 (10 characters without hyphens),
- code of language – field 041,
- author – field 100,
- title, subtitle, coauthors, part – field 245: subfields a, b, c, n,
- edition – field 250,
- place and year of publishing – field 260: subfields a, c,
- series – field 440.

Publishers (field 260b) were not included because, according to librarians, there were too many differences and errors; entries in this subfield were compared and corrected separately.

The comparison of bibliographic records could not have been limited to ISBN (International Standard Book Number) because many records had no ISBN at all. Moreover, the possibility of errors in ISBN was assumed. Nevertheless, ISBN was treated as an important point of comparison and had a large weighting coefficient.

A bibliographic record may have multiple fields and subfields. In such a case, the process of records comparison is more complex. It was assumed in our exercise that, in the case of multiple subfields 260a, only two (the first and the second) are considered. In the case of subfields 245c, considered are: the second author, the third author and the editor. Due to these assumptions, the above mentioned set of fields and subfields can be treated as an entity in a relation, *i.e.* as columns of a table in relational database. One row in such a table corresponds to one bibliographic record in a catalogue. One table was created for each library.

A pair of records in two tables was defined as "similar" (*i.e.* the corresponding bibliographic records were assumed to be approximately duplicate), if:

1. some fields are identical, or
2. some fields have minor differences (*e.g.* lack one or a few characters, have mistyped characters, have a changed sequence of characters in a word), or
3. there are no fields having big differences, or
4. in case of big differences between titles (245a) and/or subtitles (245b) and/or series (440), the condition of minor differences was first checked for a concatenation of "title & subtitle" and series and secondly for a concatenation of "title & subtitle & series".

The fourth condition was added to treat a common type of record discrepancy resultant from cataloguing by individual librarians.

It is worth noting that even in cases when all fields of two records in the compared tables are identical, the corresponding bibliographic records are often only "similar", but can in fact differ, and thus need to be compared.

Information contained in the above mentioned fields was preprocessed and stored in additional fields. In the case of subfields 245n and 260c and field 250, only numbers were selected for comparison and were written as numerals (if field 250 was empty, the number was assumed to be 1). Text information from these subfields (*e.g.* "reprint") was not taken into consideration.

In the case of field 100 and subfield 245c, the names of authors and/or editors were abbreviated (except for the last name) and written in additional fields. At the same time, the original form was checked as to whether the names are written in an abbreviated form or in full. If, in two corresponding fields, the authors were written in a non-abbreviated form, then this form was used for comparison, otherwise the abbreviated form was used.

The comparison of records was made by a group of algorithms applied to particular fields. Differences between corresponding fields were expressed numerically, and their weighted sum was treated as a measure of difference (discrepancy) between records; the bigger difference the greater the number. The sum of these numbers gave a value of "dissimilarity" or "distance". Pairs of records with low "distance" were treated as "similar", *i.e.* as approximate duplicates.

If the difference between given fields was great enough, such a pair of records was treated as "dissimilar" (there was no need to compare other fields).

The numerical fields were compared by means of simple algorithms:

1. subfields 245n were checked only for identity (0 if identical; 1 if not),
2. the number of the edition and the year of publication were cross-checked:
   - the absolute value of the difference between edition numbers was multiplied by a small weighting coefficient in the case of identical years, and by a much greater coefficient in the case of different years,
   - the absolute value of the difference between years was multiplied by a small weighting coefficient in the case of identical edition numbers, and by a much greater one in the case of different edition numbers,
   - in the case of a smaller edition number and a greater year (or vice versa) the librarians are alerted to a possible error.

In the case of all other fields, text algorithms were applied to measure "distance" between texts.

Text algorithm is a general name for an algorithm dealing with text data. A typical example of such an algorithm is a spellchecker – a program used in many word processors. Spellcheckers usually compare an edited text with a dictionary. This is not useful in the case of bibliographic records which often contain words from a number of languages, possibly in bibliographic transliteration.

There are text algorithms oriented towards searching for instances of a given string in a text (pattern matching), or the longest common substring [4, 5]. Another group of algorithms is based on the use of the so-called n-grams [6], *i.e.* sequences of n characters. To detect whether a given n-gram appears in a given string of characters (and, if so, how many times), a pattern matching algorithm can be used.

3. Algorithms

A number of text algorithms have been tested elsewhere with respect to the effectiveness of comparing records [6]. In the present project, two types of universal algorithms have been chosen to compare most of the text fields, and a heuristic algorithm has been developed to compare ISBN numbers.

The first type of text algorithm is based on the comparison of the numbers of individual characters in each of the two texts (strings) compared. It has been assumed that the strings are transformed to lower case before such comparison. The set of characters has been limited to 26 Latin letters, 9 Polish diacritics, 10 digits, space, point and hyphen (minus sign). All other characters have been treated as "other". In total, 49 characters have been considered.

For each character, the number of instances in the text is counted, 49 such numbers constitute a "profile". A comparison of two texts is based on a comparison of their profiles. The measure of "distance" between pairs of texts is defined as the sum of absolute values of differences between 49 pairs of corresponding numbers. In the case when one character in one of two texts being compared is missing, the value for such a distance is 1; in the case of a mistyped character in one record, it is 2. However, in the case of exchanged positions of two characters the "distance" is 0. This kind of error can be "noticed" by the next algorithm.

The second type of algorithm is based on the comparison of the numbers of n-grams, *i.e.* sequences of n characters, present in the texts compared. To detect whether a given n-gram is present in a given string of characters (and, if so, how many times), pattern-matching algorithms are applied. There exist so many possible n-grams that the comparison of texts is not made with respect to a fixed set of n-grams (as it has been in the case of the 49 characters), but with a dynamic set created during the process of comparing a given pair of texts. In an m-character text, there is $m-1$ digrams (2-grams), $m-2$ trigrams *etc.* Each of $m-1$ subsequent digrams from one text (usually the shorter one) is searched for in the second text. The number of digrams not found in the other text is a measure of "distance" between the texts. A similar measure is used for trigrams.

The process of comparison of all text fields (except for 020 and 041) is organized as follows:

1. First, the length of the texts is compared. If the difference between in length is greater than a chosen threshold value $t1$, the difference between the texts is stated as "great". Then there is no need then to go to steps two and three of this algorithm.
2. For each of the 49 characters, the number of instances in both texts is calculated and the absolute values of differences are totalled. If the sum total is greater than a chosen threshold value $t2$, the difference between the texts is stated as "great", and there is no need to go to step three of the algorithm.
3. As an introductory step, one space is added at the beginning and one at the end of each of the two texts. Subsequent digrams are taken from the shorter text (text one); for each such digram, its occurrence in the other text is checked by a pattern-matching algorithm. The number of digrams not found in the second text is totalled; if the total is greater than a chosen threshold value $t3$, the difference between the texts is stated as "great". If the difference is not "great", a numerical measure of the distance between the texts is calculated by adding the sum total from step two and the number of digrams not found in the other text in the present step of the algorithm.

Threshold values for all the three steps depend on text length and the type of information. For author names, the threshold values are small, while they are greater for the titles. The following formulas have been used, where li stands for the length of i-text (it was assumed that $l1 \leq l2$) and E stands for the function equal to the integer part of its argument:

for the names of authors:

$$t1 = 5,$$
$$t2 = 1+(l2-l1)+E(l1/10),$$
$$t3 = 3+E(l1/10);$$

for titles, subtitles and series:

$$t1 = 5+E(l1/15),$$
$$t2 = 3+(l2-l1)+E(l1/15),$$
$$t3 = 6+E(l1/15).$$

Publishers (260b) were compared by means of an algorithm similar to the one used for comparing titles, having some additional rules to deal with differently abbreviated terms.

A heuristic algorithm has been developed to compare ISBN numbers.

1. All corresponding digits (characters) are compared sequentially, *i.e.* the first with the first, the second with the second *etc.*
2. If there are no more than two differences, the measure of the distance is calculated as follows:
   - in the case of a changed sequence of two following digit, a relatively small value of distance is given (*e.g.* 0.3);

- in the case of the following pairs of digits: 3–8, 1–7 and 6–9 on corresponding positions, the difference is small, while in all other cases it is standard (*e.g.* 0.3 and 1);
- these differences are multiplied by a large weighting coefficient when the difference occurrs in the initial position of the ISBN number and by a medium coefficient when in the second position.

3. If there are more than two differences, two cases are checked:
   - part of the digits is "moved" in cyclic way (the "distance" is proportional to the length of the cycle);
   - one digit is missing, others are moved forward and some other digit or character is added at the end (the "distance" is greater than in the case of a cycle).
4. In other cases, the difference between ISBN numbers is treated as "great".

ISBN numbers are compared only when two records have this field non-empty.

The code of language is checked for identity in the case of a single code (three characters). In the case of multilingual codes, the codes are transformed into separate single codes and compared as sets of codes (the sequence of codes is not important, only the presence of a given language).

4. Statistics

The catalogue of the Main Library had 49 000 of records. The catalogues of the other libraries had 14 000 of records in total, and were added sequentially. Each of them was compared to the Main Library (ML) catalogue, and the second of them was also compared to the first, the third – to first and second *etc.* For each library, the following was specified:

- the number of bibliographic records in its catalogue,
- the number of duplicated records found, and
- within this number, the number of duplicated records with non-zero measure of distance (MD).

Lib. 1. 7 334 rec. – 2 472 dupl. rec. to ML (727 with MD > 0).

Lib. 2. 1 669 rec. – 470 dupl. rec. to ML (164 with MD > 0) and 33 to Lib. 1.

Lib. 3. 3 062 rec. – 788 dupl. rec. to ML (217 with MD > 0), 63 to Lib 1. and 7 to Lib 2.

Lib. 4. 1 969 rec. – 477 dupl. rec. to ML i Lib. 1-3 (254 with MD > 0).

The catalogues of the four joining libraries contained 14 034 records. 4 309 pairs of records, *i.e.* approximately 31%, was found to be duplicate, among them 2 848 pairs, *i.e.* approximately 20%, were nearly identical (MD = 0), and so easy to find, while 11%, *i.e.* 1 450 pairs of duplicate records with a non-zero distance, were found mainly due to the application of the text algorithms. About 30 cases of duplicate records with mistyped ISBN numbers were found.

n-grams proved to be helpful especially in the case of short texts, *e.g.* authors. After a number of tests, the comparison was based only on digrams. In the case of short texts, threshold values for trigrams had to be too great in comparison with the length of texts. In the case of long texts, the comparison of trigrams added little value to the information obtained from the comparison of digrams. Generally, in the case of long texts, the comparison of profiles was precise enough, while being much faster then the comparison of *n*-grams.

The comparison of bibliographic records helped to improve data quality and identify some types of differences and librarian errors.

Generally, the presented approach to joining catalogues into the existing union catalogue was approved at the Main Library of the Warsaw University of Technology as helpful and efficient. This approach can also be applied to detect approximately duplicate information in other catalogues and databases.

References

[1] Cousins S A 1998 *J. Information Science* **24** (4) 231

[2] Preece B 2001 *The Journal of Academic Librarianship* **27** (6) 470

[3] Jolibois S, Mouze-Amady M, Chouaniere D, Gradjean F, Nauer E and Ducloy J 2000 *Work & Stress* **14** (4) 283 (on-line DOI: 10.1080/02678370110040056)

[4] Atallah M J, Chyzak F and Dumas P 2001 *Algorithmica* **29** 468 (on-line DOI: 10.1007/s004530010062)

[5] Baeza-Yates R and Navarro G 1999 *Algorithmica* **23** 127

[6] Tian Z, Lu H, Ji W, Zhou A and Tian Z 2002 *Int. J. Digital Libraries* **3** (4) 325 (on-line DOI: 10.1007/s007990100044)

Implementation of the Regional Meteorological Database

Marek Wojtylak and Krystian Rorbek

Institute of Meteorology and Water Management,
Regional Department of Katowice, Bratkow 10,
40-045 Katowice, Poland, admzmbs@imgw.katowice.pl,
Krystian.Rorbek@imgw.pl

(Received 4 November 2002; revised manuscript received 31 January 2003)

Abstract: The special character of meteorological data, especially various measurement times and standard statistics, makes a programmer solve non-standard problems. The Regional Meteorological Database (RMD) was created in the early nineties of the last century. The RMD uses a data module stored in files, a control module (for data reading and recording) based on indexing by the hashing functions, and a module which makes it possible to display data.

Keywords: hashing function, meteorological database, pointer file, portion file

1. Introduction

The special character of meteorological data requires an individual approach to record them and get the best performance. The main obstacle to unite meteorological data has been related to various observation frequencies. For example, observations have been performed every 6 hours by climate stations and every hour by synoptic stations. At the same time, some parameters have been observed once a day, others – every 6 hours, still another – every hour, and some – only during their occurrence.

Another problem has been that the number of meteorological parameters depended on the type of an observation post. For example, some of the rainfall measuring stations observe only one parameter, whereas synoptic stations record over 100 parameters.

Therefore, there has been no possibility to create a satisfactory and user-friendly table structure on the basis of the existing database software. In this situation, a decision has been made to develop a dedicated database structure. Thus, the Regional Meteorological Database (RMD) has been created. The RMD is composed of three modules: a customer module, a control module, and a data module.

The Data Module is composed of three files:

- a *portion file*, where data is stored as a portion of a constant size,
- a *pointer file*, where addresses of the first portion data are stored,
- an *auxiliary file*, where information facilitating data management is stored.

It has been decided that a data set corresponding to one meteorological parameter measured within a month by one post will be used as a logical object. Such an object has been called a MIESIAC, and is created in the Control Module.

A meteorological parameter can be measured with varying frequency and, so, the physical size of a month may vary as well. A month is divided into data portions corresponding to parameters measured with various frequencies, and such data portions are stored in the data file. Such a system makes it possible to store parameters in the database exact to a minute.

2. Implementation of the RMD

A user creates a query by defining such parameters as year, month, a meteorological parameter, and an observation post. Then, an object MIESIAC is generated according to this query and a key is evaluated by means of a hashing function. According to this key, the Pointer File is reviewed. The Pointer File is a constant table with a size corresponding to a prime number. The key designates a position in this table, where an address of the first data portion is recorded. Afterwards, it is checked whether the record of the Pointer File contains data corresponding to the required key or not. If negative, the pointer table is searched by means of the cubic method. After finding the address of the right data portion, this data portion is taken from the Portion File and stored in the table of MIESIAC. The data portion contains its number and an address of the next data portion. If a portion isn't the last one, the next portion is extracted and so on, up to a complete filling.

The Control Module uses standard instructions to open a disk file and store its parts to the memory buffer, so that the software needs no database controllers.

3. Current state

The performance of the RMD depends considerably on the filling degree of the pointer table. The more filled the table, the larger number of collisions occur, which means that the number of searchings within the table increases.

Since the starting point, the table capacity has increased four fold (121 441, 255 023, 400 187, 600 011). The table capacity is presently equal to 600 011 and is filled as much as 63.4%.

The Pointer File presently contains 380 217 indexes, which point at 1 045 153 portions stored in the Data File.

These portions include data related to 85 meteorological parameters measured by 214 observation posts. We have at our disposal complete data sets for most of these posts since 1960. However, some data originate from the fifties of the last century.

4. Maintenance of the RMD

The RMD has now operated continuously for over 10 years. Every month, the database size goes up by 210 indexes, which point at 1 000 data portions.

The maintenance of the RMD relies on its rewriting to obtain as large a number of indexes found without collisions as possible. At the same time, wrong indexes are removed.

In the case of the Data File, the maintenance procedure relies on the arrangement of as large data sequences of one element as possible, and especially on the arrangement in neighbourhood of the data portions used in the composition of the same object MIESIAC. Owing to this, data access time will be minimized.

Structure and Extension Forms of the NPL (News on Forest Literature) Reference Database

Ewa Lewandowska

Forest Research Institute,
Bitwy Warszawskiej 1920 r. 3, 00-973 Warsaw, Poland,
E.Lewandowska@ibles.waw.pl

(Received 30 July 2002; revised manuscript received 28 February 2003)

Abstract: The "News on Forest Literature" database contains general reports from journals and serial publications. The database has up to 75 000 records. The database has been created with MICRO CDS/ISIS software. WWW versions of the database contents are made available with EasyInt and ISISWWW software.

Keywords: forest bibliography, EasyInt, ISIS, ISISWWW

Since 1989 a computerized database for NPL (News on Forest Literature) has been maintained at the Scientific Information Department of the Forest Research Institute in Warsaw. It contains scientific and general reports from journals and serial publications (Polish and foreign) on forestry and related fields. The database is handled by the Department workforce and regularly updated from the documents collected in the Institute's library, which is a crucial element contributing to the value of the database. Over 70 thousand reports have been indexed so far, and this forest database is unique in Poland.

The MICRO CDS/ISIS (free of charge) software has been used to create the database. The CDS/ISIS software is a system for collecting and searching information. It was designed especially for computerized management of structural non-numerical databases, *i.e.* databases in which text is the main component. The NPL reference database contains information on papers from journals and each information unit is composed of elementary data: author, title, publication date *etc.*

A printed (hardcopy) form of the database in also available. However, these are printouts of a part of the database (about 350 records) that are published monthly as a publication of the Forest Research Institute in Warsaw (IBL) entitled "Nowości Piśmiennictwa Leśnego" (News of Forest Literature) – NPL. A WWW version of NPL has been presented on the IBL homepage since 1998 (*http://bazy.ibles.waw.pl/bazy/npl/index.html*). It has been possible thanks to the EasyInt (Easy Presentation on Internet) program. The program has been used to present a part of the textual content of the NPL database on the Internet. EasyInt processes text files prepared earlier by the user, containing suitably sorted database records. Such files are created in the CDS/ISIS package when generating printouts recorded into files. Three printouts are made from the bibliographic content of the NPL database, sorted according to authors, keywords and forest classification. EasyInt makes it possible to present the database in a static way, and this means that the content of the database is made accessible on the Internet in a form which is up-to-date at the moment of making printouts, and any changes introduced later will be seen after generating and processing new printouts.

The appearance of the NPL journal on the Internet has aroused interest and approval of users. However, critical remarks have started to appear as well. Users who wanted to information from several years have complained that they had to search through several, even a dozen or so issues of NPL. Moreover, we are aware that not all documented articles have appeared in NPL, due to selection of materials in the process. Therefore, the user has had no possibility to see the entire database on the Internet, even by going through all the issues of NPL. This has become possible only with the application of the ISISWWW software to searching the database on the Internet. It has been updated and improved, and the following facilities have been added: easy searching and using indices choosing the presentation format, selection of right-side masking of searching terms, selecting the way of using operators within one field, improved reviewing through multiple-page search results, asking via links on the basis of information taken from reviewed description, presenting headings, footnotes, and background on generated WWW pages depending on the database being made accessible. To search information in the NPL database, as in other databases, a WWW search page is used, where the user can formulate a request, review indices *etc.*

The "Schwappach's Permanent Plots" Forest Database: an Announcement

Marek Wirowski

Forest Research Institute in Warsaw, Forest Management Planning and Monitoring Department, Bitwy Warszawskiej 1920 r. 3, 00-973 Warsaw, Poland, M.Wirowski@ibles.waw.pl

(Received 7 November 2002; revised manuscript received 15 January 2003)

Abstract: Experiments in the field of dendrology have been made for many years now. Current researchers, working on these plots, are the forth generation of foresters (two German and two Polish). They have created a large and important database for future forest research.

Keywords: forest experimental plots, internet database, characteristics of stand

The forest database "Schwappach's permanent plots" is an example of co-operation between Polish and German foresters in the field of experimental forest research. The database contains information about the location, growth and development of stands in the experiments planned and started within the framework of the programme of Prussian Experimental Stations. The data have been collected since 1890. In the late 1950, the Forest Research Institute in Eberswalde shared the collected resources with the Forest Research Institute in Warsaw. The exchange of data and experimental cooperation has continued since then, with joint research expeditions being organized.

The scope of data collection has not changed significantly, while the technique of measurements and collecting data has changed considerably. The data input sheets and forms have been replaced by digital carriers, and the database is currently available outside the Institute. Many research works, including master's and doctoral dissertations, have originated on the basis of information contained in the database. The database also finds its application in conducting various types of experiments.

The database contains information about 67 forest experimental plots and the records of the dbh growth under bark (diameter at breast height (n) - tree width at a height of 1.30 m) of all trees growing on the plots and the heights of selected trees contributing to calculating the growth curve of a stand.

The access to each plot is provided using GPS. The plots have digitized cartographic materials.

A system has been developed to enter data directly to a palmtop-type computer with the MS Windows PocketPC 2002 system, containing electronic data input sheets based on the measurement data collected in the previous measurement period.

Data stored in the database are available in the internet in the form of graphs and tables. A photographic file has also been made available via the Internet. It contains electronic records of photographs taken during inspections and measurements on the plots.

A Database on the Tourist Sites of Cracow and its Environs

Robert Krupa, Maria Pociecha,
Joanna Szlezynger and Grażyna Kruszelnicka

Department of Statistics and Computer Science, Institute of Tourism, Academy of Physical Education in Cracow, Jana Pawla II 78, 31-571 Cracow, Poland, wspociec@cyf-kr.edu.pl

(Received 15 October 2002; revised manuscript received 14 January 2003)

Abstract: The database of tourist attractions of Cracow and its surrounding area contains categorized data about the tourist infrastructure and attractions of the City and the entire Malopolska region. It includes texts, graphics and numeric data.

Keywords: tourist attractions, tourist trails, architectural monuments, tourist database

Nowadays, there are many information directories and particular offerings for tourists.

These may not only be a source of information for potential customers and tour-operators, but may also be a source of valuable research material for tourism as an important sector of a market economy. This information quickly loses its immediate interest. There are no comprehensive data with information concerning tourist facilities and attractions. Therefore, we have taken the initiative to establish a database system that would include a wide range of data relevant to tourism. These include descriptive information in text files and graphic data, including photographs and drawings.

There are three basic modules in the system:

- acquisition and recording of data,
- data sharing and presentation,
- data sharing as an on-line service.

Cracow is rich in tourist attractions, and it is a very good example of how to create databases of this kind. This was made use of in designing the database on its monuments.

The data on monuments include:

- the name of a tourist sight,
- its localization,
- description and
- classification,
- a presentation of the sight and
- its photograph.

An important problem in the arrangement of the data on monuments was to determine how to classify them. In order that each and every sight could be clearly localized, we introduced our own system of their classification. The majority of the monuments recorded in the system are part of a group of architectural monuments. The classification introduced mostly concerns these sights. This division distinguishes the elements of religious and secular architecture in detail.

At this stage, the data are collected as Paradox tables: the interface for the database operator was compiled by means of Delphi 3 Professional by Borland, whereas the applets were compiled by means of JBuilder 2 of the same company. This choice of tools was due to the fact that, at the stage when this application was being developed, it did not need be connected to any particular type of database, thanks to its access mechanism to databases of various kinds. Secondly, this tool makes it possible to go beyond the restrictions that result from the nature of the database used, because it is, at the same time, a programming language.

This system has also been equipped with some tools to allow to design tourist trails on the basis of the data collected on tourist attractions. It enables users to group the tourist sights according to their own ideas and allows them to create alternative tourist trails. The data collected in the database can be customized. A collection of data which are thematically interrelated is customarily referred to as a trail. This makes it possible to prepare alternative "guide-books" or "info directories".

At the first stage of the implementation of the program, we presented four existing standard trails, and our choice was based on historical tradition. These were:

- the Royal Route,
- the trail leading to Cracow's Colleges and the Jagiellonian University,
- Cracow's Kazimierz - an old Jewish residential district, named after King Casimir the Great,
- Medieval burghers' houses.

Each of the presented trails consists of the same elements. First, users familiarize themselves with a general description of a particular trail. This is a short introduction, which gives the most important facts on the trail in question, its historical significance, as well as other basic information. Interactive maps form another component. Users can see the trail marked with a red line across the region they are interested and identify numbers marked thereon with individual monuments. On the right-hand side, there is a key to each of the maps, which gives the names of sights on a particular trail. The third and the most important part is a collection of photographs and notes concerning particular spots on the tourist trail. It is here that users can find descriptions of particular sights, each being further illustrated with an artistic photograph.