ICE

Information, Controlling, Entscheidung

ICE – a Web-Based Information System to Support Higher Education Policy Decisions

EUNIS 2003

Beyond the Network

9th International Conference of European University Information Systems

2 – 4 July 2003 – Amsterdam – The Netherlands

ICE – a Web-Based Information System to Support Higher Education Policy Decisions

Peter Müßig-Trapp, Hans Dicken, Helena Kopp*

*HIS Hochschul-Informations-System
Hannover, Germany
muessig@his.de, dicken@his.de, kopp@his.de

Abstract

ICE stands for Information, Controlling, Entscheidung (in English: Information, Controlling, Decision) and is an information system developed to support higher education policy decision-making which has proven itself in practice. The system is currently in use at the German Ministry of Education and Research[1], at science and research ministries in ten German federal states[2] and at other organisations active in the field of higher education policy (eg, the German Science Council[3] and the German Academic Exchange Service[4]).

The debate on protecting personal data which arose in the 1980s led to increasing sensitivity in the population regarding data protection questions and to stricter data protection rules. Both resulted in fewer possibilities of statistical data analysis being allowed: In many fields, statistical analyses can only make use of so-called aggregated datasets. With its ICE information system, the HIS now provides a solution which is capable of extracting a maximum of information from fundamentally limited aggregated datasets. At the same time - subject to their appropriate availability - the system also allows the analysis of individual case data records.

The following outlines the main system features. ICE

- is a web application, is Java based, platform independent (for example, backend runnable under Windows, Linux, SUN OS), and database independent (eg, Oracle, Informix, MySQL),
- offers very high data import and data analysis flexibility,
- uses XML technology and the Apache Cocoon based ICE Publishing Framework, which means that it provides a wide range of output formats (XML, HTML, XHTML, Excel, Gnumeric, PDF, etc), simple data exchange with third party programs, and is capable of handling future technologies.
- has further developments planned in the direction of data expansion, internationalisation, and towards mobile computing (functional expansion of the existing cellular version into a handheld version, ICE-Mobil).

[1] Bundesministerium für Bildung und Forschung (BMBF)
[2] Bundesländer
[3] Wissenschaftsrat
[4] Deutscher Akademischer Austauschdienst

Keywords: Information system for statistical data, policy advice, higher education planning

1. History

At the start of the 1990s, ICE evolved as a commissioned project on behalf of the German education and science ministry[5], which later merged with the research and technology ministry[6] to form the BMBF. At the time, the ministry used the Macintosh operating system, which explains why the ICE system was originally developed on the proprietary HyperCard basis. So the application was only available under Mac-OS. The mid 1990s saw desktop environments migrate towards Microsoft Windows. This called for a new development. This need was used as an occasion to completely reengineer the system architecture and the technical framework. An Intranet system was developed based a modern multi-tier-architecture, with a database at the backend and Java as its central development platform. This made the system accessible with an Internet browser, regardless of platform. A Content Management System was added as an extra component to visualise the structures and procedures within the participating ministry departments and served to support general document exchange.

The year 1997 saw the system adopted by the German Science Council albeit with a specifically defined data basis: System architecture and software components were very largely adopted direct; only the data basis (the backend module) was changed. Analogue extensions were undertaken in subsequent years for the science and education sections of the relevant ministries of the German federal states (protected Internet access) and for the DAAD.

Changing and expanding user requirements resulted in the system being continually carried forward. For example, the presentation and export options were greatly enhanced and extended for the BMBF. XML-based modules were developed to allow the ministry to automatically generate the layout for its annual book of tables "Basic and Structural Data"[7] direct from the ICE database. The needs of the federal states called for user authentication and a system of rights to provide finely-structured access control to the modules in the system and especially to the user-generated analyses.

[5] Bundeministerium für Bildung und Wissenschaft (BMBW)
[6] Bundesministerium für Forschung und Technologie (BMFT)
[7] Grund- und Strukturdaten

ICE can also be understood as a response to the restrictions introduced on the analysis of statistical information at the end of the 1980s. While access to official micro data had been managed quite liberally for science, education and administration up until the end of the 1970s, extremely restrictive access rules were introduced against the background of the data protection debate which set in at the beginning of the 1980s. At the time, an occasionally controversial political and legislative debate on the planned census and on the protection of personal data raged in Germany. In the wake of great public sensitivity, the existing legislative provisions were interpreted restrictively, while further hurdles were established by introducing additional rules. This is why administrators analysing statistical data cannot, as a rule, access individual case data; they can only access so-called aggregated datasets.

Besides the individual university administrations, only the offices of statistics are allowed to analyse official individual case data in the higher education sector. However, these offices regularly publish their standard analyses (aggregated sets). Composition and structure of these aggregated datasets were defined in time-consuming negotiations with advisory bodies and, as a rule, have now been set for many years. The offices of statistics are able to meet the information needs that can be satisfied by these standard aggregated datasets relatively quickly; however, any structured analysis requests which differ from these can - depending on the workload and technical specifications involved - only be made available, if at all, in the form of special analyses (as a rule, assembler programming) after quite a time delay.

The range of analysis options allowed by standard aggregated datasets are comparatively limited and, generally, allow only precisely *those* analyses to be made which had been the intention of the previously-defined respective aggregated dataset. It is absolutely impossible to completely predict the information needs of politics for even just a few months, let alone over years. This is why it was obvious that long-term planned standard aggregated datasets and the answers they contained to questions that had been anticipated years before would only be able to partly satisfy the higher education policy questions which arise now. This is why there was and still always is a wish for making individual case data records available for statistical analyses.

Recently, a development has been observed in Germany which may be viewed - depending on viewpoint and subject area - either as a relaxation of the rigorous approach to data protection questions or as a crisis in the data protection field. A wide range of differing causes underlie this development, of which overregulation by rules, decrees and laws (data protection has become a domain of the lawyers) is certainly an important one. The increasing willingness of far-sighted players in the field of data protection (eg, the Lower Saxony Data Protection Officer) to allow the use of statistical analyses from individual case data records for academic and administrative purposes, as long as the individual's right to informational self-determination has not been impacted, is a consequence of this development.

In the face of this background, a major new development was started on behalf of the Lower Saxony Ministry of Science and Education[8]; this project is currently in its implementation phase: An ICE with deeply-structured data for individual federal states which not only allows the regularly used standard aggregated datasets to be processed but now also allows very broad aggregated records and also individual case data records to be processed.

Therefore, ICE provides the possibility to process standard aggregated data as well as individual case data. A combined analysis of both kinds of data is also comfortably possible. This is important, because huge portions of higher education statistical data still will only be available as aggregated data in the future.

2. Goals

ICE was and continues to be developed with the following five central goals in mind: Firstly, to make available a system with which the maximum of possible analysis options can be achieved from essentially restricted aggregated data records. Secondly, and at the same time, to develop a data warehouse, that is a system with which the widest possible range of different data stocks (with various depths of structure, quality levels, presentation options, etc) can be integrated and flexibly analysed. Thirdly, to make the system platform independent in order to avoid - as has already happened in the past - a change of a customer's operating system or central database resulting in large sections of the system having to be redeveloped. Fourthly, to make the system easy to operate with the minimum possible need for computing skills or detailed data knowledge. As far as possible the necessary know-how should be contained in the system itself, so that the user - as far as possible free from technical and formal considerations - can concentrate on formulating data content requests. This means that the target group for the system not only extends to experts well-versed in computing, statistics and data content, such as specialists from the ministries of science and education, but also to executive-level decision-makers. Fifthly and finally - and this goal was only added recently - to provide a system which also allows the quick analysis and interpretation of comprehensive individual case data as well as combinations of individual case data and aggregated data records. The background to this is formed by the cautiously growing willingness on the part of German data protection officers to allow the analysis and interpretation of anonymous individual record data.

These principle goals have not yet been fully achieved in all areas. However, the system is being continuously developed. Focuses and priorities are set on the basis of the needs of the customers who are financing the development. The current state of development and so the performance spectrum of the system will be outlined below.

---

[8] Niedersächsisches Ministerium für Wissenschaft und Kultur

3. System features

- *Web application.* Access to an ICE installation is enabled via a network using a Java capable web browser (such as the open source browser Mozilla, Netscape Navigator or Microsoft Internet Explorer). In principle, this means that the system can be accessed from any computer on the Internet or Intranet which is registered with the ICE server. Access to the system (or to parts of the system) can be restricted to authenticated users as required. Where necessary, the system (or parts of the system, eg, collections of standard tables) can be set up to allow access from the Internet.

- *Platform independence.* On the server-side, the system can be installed under Microsoft Windows as well as under Unix (Solaris) and Linux. The client end (user) only needs a Java capable web browser. Browsers are available free of charge for all the commonly-used platforms (such as MS Windows, MacOS, Linux). The system is also independent in terms of the relational database management system that is chosen: The system has also been installed under Oracle and Informix. We are currently testing the use of open source databases (MySQL, PostgreSQL).

- *Flexible data import.* Data with any structure and depth of structure can be imported. The system can also be expanded to include new topics. Since very recently, it has been possible not only to analyse and interpret aggregated data but also comprehensive individual case record data with good performance. Similarly, the combined analysis of aggregated and individual case record data is possible.

- *Flexible data analysis.* The very flexible import of data stocks is mirrored by an equally flexible range of analysis and interpretation modules. Using the so-called "flexible table generator", data stocks available in the system can be used to output any extract in tables. Analyses using information from various data stocks can also be easily requested: It is no problem combining information from several data stocks in a single results table.

- *Flexible data export.* Results tables produced with the flexible table generator can be stored in HTML and MS Excel format. This makes it possible to process the tables using third party programs, to pass on statistical information to other interested persons, eg, by e-mail, and to create information and data collections on the web. The data export options were extended substantially in 2002. The ICE Publishing Framework provides a tool which makes additional output formats available. On the one hand, this contains an XML interface which can be used for data exchange and as a universal interface to third party programs (eg, other databases, spreadsheets, graphics programmes, geographical information systems, and so on). On the other hand, this format is suitable both for web-based presentations as well as for high-quality print-outs. The user can influence format and appearance of the PDF output in many different ways.

- *Data harmonisation with an integrated key*: All the data contained in the system are encoded with a uniform ICE key. The project team centrally updates and hosts the key. This ensures that - as far as meaningful for the content - various stocks, also possibly from various sources, can be analysed together. Equivalency rules are defined where necessary to make possible comparison of variously encoded data, which do have like content however, (example: subject groups in the staff statistics ←→ subject groups in the student statistics). The system can also recognise key internal hierarchies and places knowledge at the disposal of the user for carrying out sorting functions, for example. (The system "knows" that the University of Hannover, for example, belongs to the state of Lower Saxony and to the higher education institution type "university").

- *ICE standard tables (with integrated automatic updating).* All results table produced using flexible table generation can be stored as so-called ICE standard tables in the standard table collections. These table collections can also be made accessible to third parties on the Intranet or Internet and can be searched both by a hierarchical directory structure as well as by a keyword search. The integrated automatic update is a particularly useful feature of the ICE standard tables: A once generated table which has been stored as a standard table can automatically be updated at the touch of a button with data imported into the system at a later point. The user can choose from various update options (eg, time series addition, time series shift, substitution of the whole table with the latest available data). Using the ICE Publishing Framework it is possible to request these standard tables in various formats. At present, the following formats are available: HTML, XML, MS-Excel, Gnumeric, PDF. The XML-based technology (Apache Cocoon) we used means that the creation of corresponding XSLT style sheets allows other output formats to be made available with little effort. Just as unproblematic are the user-defined modifications to the output format (colours, document size, fonts, etc) which can be produced. This is relatively easy because all output layouts are produced on demand and on-the-fly and the relevant information is extracted from a database; it is not necessary to fall back on prefabricated files.

- *Reliable and reasonably-priced data updates.* The import of new data stocks and updates of existing stocks is done by the HIS-ICE group. Central system maintenance, data processing and data administration assure a high degree of data quality and reduce costs by producing synergies (eg, keys developed for one customer can be modified for another customer).

- *Hotline.* The ICE group provides advice and support on all questions relating to the information system (both by telephone and by e-mail). This service includes technical questions (network problems, security settings) and operating questions on the software (browser, ICE application) just as it does subject-related/statistical questions (regarding data and their analysis, keys, and other aspects).

- *Development carried forward all the time.* The development of the ICE is being carried forward all the time. Upgrades and improvements produced for a particular customer will be made available to other customers a short while later.

4. Data

In general, the system can process practically all kinds of data; the only condition lies in the development of an appropriate key for the classification and integration of that data. Data encoding with a uniform ICE key means that different data stocks, possibly also from various sources with differing levels of data quality, can be analysed together. Each data s-tock is registered with the help of the ICE key as far as data source and data quality are concerned and can be requested for analysis and interpretation. At present, the system largely contains data from the Federal Office of Statistics[9]. However, data from other suppliers - eg, German Science Council, BMBF, KMK[10], BLK[11], Bundesanstalt für Arbeit[12], HIS, EUROSTAT, OECD, UNESCO and a number of others - can be integrated into the system and analysed. Appropriate data were, for example, integrated into the system for the implementation of the BMBF's Basic and Structural Data publication. The "data quality" feature provides information on the credibility of data (final or preliminary data from the official statistics, random samples, forecasts, etc). This facilitates an extremely high degree of flexibility as regards the provision and processing of statistical data from various sources/systems. It satisfies both the need for reliable and generally comparable data from the official statistics as well as for the very latest corresponding data from other sources.

The system can import data of any structure and depth of structure. The system can also be extended flexibly as far as topics are concerned. The content focus lies specially on higher education statistical data: University staff, student and examination/degree data, holders of higher education entrance qualifications, university funding statistics. At present, the system is being expanded to cover a wider range of educational statistics. Besides higher education statistics, the system is also capable of analysing and interpreting data on kindergartens, schools and vocational training, continuing education, resources for education, science and research, educational assistance, etc (as implemented in the ICE used by the BMBF).

To ensure that the user does not have to concentrate too much examining the various statistics and data in detail, intelligence has been implemented into the system itself at many points. For example, if the user wishes to generate a table which contains both student numbers as well as the case numbers on university staff, each arranged by subject groups, then the user should know that the subject group classification in the official statistics differs for both cases. Equivalency rules are maintained in the system for such comparison of variously keyed but - in terms of content - comparable data. They make it possible - completely transparently for the user - to assign the corresponding data from differing statistics and so facilitate the integrated analysis of different data.

Besides equivalent relationships between ICE keys, the system also has so-called implication rules which reflect a large part of the key contexts. For example, implications determine to which federal state or which type of higher education institution an individual university belongs. This knowledge is placed at the user's disposal in certain cases, eg, for carrying out sorting functions.

5. The system from the user's perspective

The ICE user accesses the various functions of the ICE system from its start page. Two methods are available for searching for data in the ICE. On the one hand, the available, updated and maintained *ICE standard table collections* can be used. ICE standard tables offer frequently needed statistical data in easy to use and quickly accessible formats. On the other hand, data can be quickly requested from the database with the help of the central module known as *table generation.* In addition, the start page contains tools to manage the tables, to update the standard tables as well as lists of the available data stocks. All these functions are protected against unauthorised access by an authentication system. The following will briefly outline the table generation process and its management.

Table generation

In principle, it is possible to differentiate between two kinds of table generation: *Single data stock* and *multi data stock table generation.* Single data stock table generation (Table Generation I) can only be used to produce tables whose data stem from *one* predefined virtual single data stock within the ICE. A single data stock is a given basic stock within the system of matching data (eg, a time series on students and study entrants, categorised by several features). The particular advantage of single data stock table generation lies in the fact that only those data can be combined with each other which also "fit together". So, it is not possible with this tool, for example, to combine staff data (staff, staffing positions) with data on teaching demand (students, study entrants). So this rules out the possibility of making subject-related mistakes, because such data are by definition not contained in a single data stock. This safety leads to less flexibility than the multi data stock table generation (Table Generation II) allows, because the latter allows several data stocks to be combined with each other in a table presentation. Single data stock table generation represents the standard form of system usage, be-

9 Statistisches Bundesamt

10 KMK = Kultusministerkonferenz (Standing Conference of the Ministers of Education and Cultural Affairs of the Länder of the Federal Republic of Germany)

11 BLK = Bund-Länder Kommission für Bildungsplanung und Forschungsförderung (Federal-State Commission for Educational Planning and Research Promotion)

12 Federal Labour Office

cause it requires no in-depth specialist knowledge on the subject area to be reflected by the data.

The *Table Generation* window has a menu bar (with the following index tabs: "Keywords", "ICE Data Stock", "Data Stock Selection", "Table Definition" and "ICE Key"). These index tabs are sequentially processed from left to right up to the table definition tab to produce a table. Essentially, a table is produced in three steps. Users

1. select suitable keywords from a list of keywords to circumscribe the desired topic area,
2. select a data stock from a number of suggestions, and
3. define the table layout and table content.

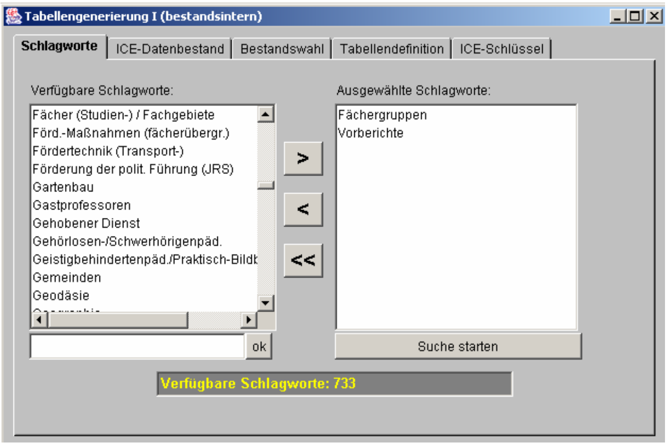


Fig. 1: Window for selecting keywords

Keywords reflect the whole stock of features and categories in the ICE key. There are keywords which reflect a feature within the ICE key (eg, *sex*) and keywords which relate to a form of the ICE key (eg, the form *male* of the feature *sex*). Once the keywords have been selected, the system lists all the data stocks which contain information on the given keywords under the menu item *ICE Data Stock* including a short description. In the following window ("Data Stock Selection") the selected data stock is once again described in detail along with the available forms. It is easily possible to change the data stock selection.

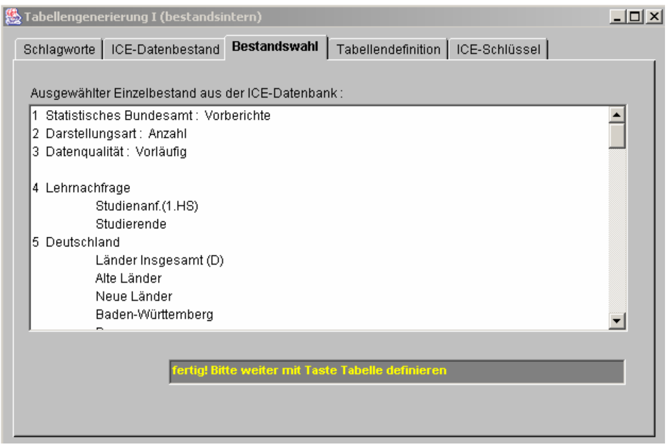


Fig. 2: Window for selecting a data stock with detailed data stock description

To *define the table* the user determines in the next window (index tab "Table Definition") which features along with which values are to be assigned to columns and which to rows.

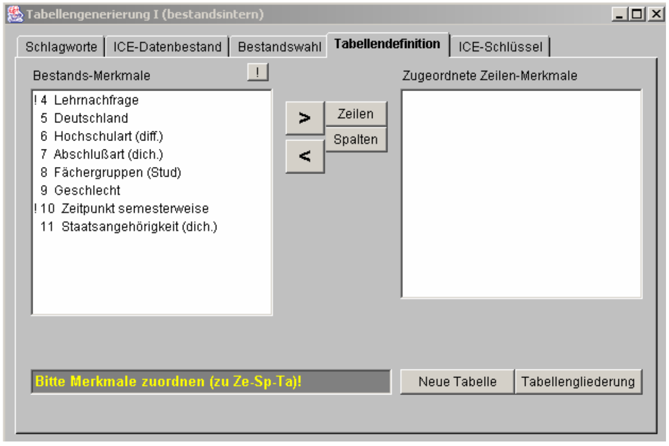


Fig. 3: Window for defining the table

It is possible but not necessary to use all the features offered. If a feature is not selected (eg, sex) then the tables will contain values for the respective "Total" ("male" *and* "female"). Some features have no meaningful "Total" (eg, timepoints "1998", "1999", "2000"; …). In such cases, the user must make a selection. The order of the selected features corresponds with that in the later table. It can subsequently be modified by shifting, and it is also possible to remove features and, when so desired, to reselect them in a different order. In the case of extensive features, the system offers a sorting function; this allows university towns, for example, to be sorted by federal states and/or types of higher education institutions. The user can check the selected values at any time via the table structure function.

With the help of the *Table Generation II* function it is possible to produce tables whose data stem from various single data stocks. For users, this ICE function is like generating a series of tables under version I. A table produced with the help of a multi data stock generation is made up of a series of table sections which have each been individually defined. The structure of such a table corresponds with that shown in Figure 4 below.

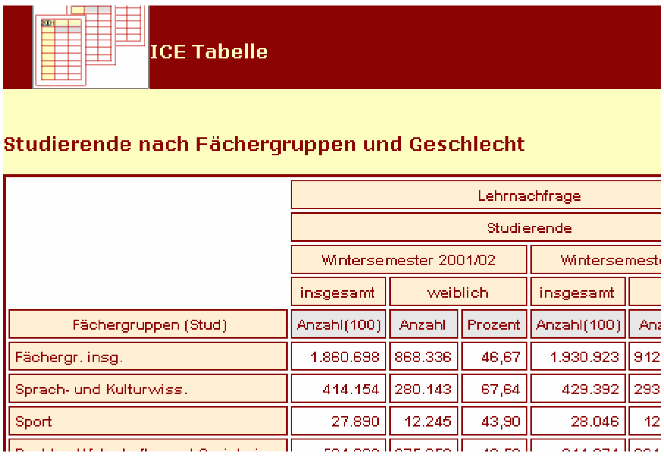
Generation of this table begins with "Table Section 1/1". This step automatically labels Row 1 and Column 1 accordingly. The second step then adds "Table Section 1/2" (with given Row label 1) or "Table Section 2/1" (with given Column label 1). All other table sections are then added in the same way (Table Sections 2/2 and 2/3 follow automatically and do not therefore need to be defined). Any number of further extensions of the table are allowed, which means that columns and/or rows can be added and extended at any time. Since several table sections are merged to form the overall table with in some cases identical labels, it is particularly important when using this version of the ICE to think through the table structure especially thoroughly beforehand.

	Column Heading1	Column Heading 2	Column Heading 3
Row Heading 1	Table Section 1/1	Table Section ½	Table Section 1/3
Row Heading 2	Table Section 2/1	Table Section 2/2	Table Section 2/3

Fig.4: Table structure after multi data stock table generation with several table sections

The table can be generated once the table structure has been described, ie, the desired data are requested from the database and the table assembled in accordance with the table definition. Beforehand, a window indicates the number of requested values and enables the user to correct the table definition if the data should turn out to be too extensive.

The multi data stock generation of tables allows subsequent *data calculations to be carried out*, eg, percentage calculation, indexing, quota formation and difference formation. Finally, the user can output the request result in various formats (eg, Excel, HTML), or also store it on the sever as a so-called standard table.


ICE Tabelle

Studierende nach Fächergruppen und Geschlecht

	Lehrnachfrage				
	Studierende				
	Wintersemester 2001/02			Wintersemest	
	insgesamt	weiblich		insgesamt	
Fächergruppen (Stud)	Anzahl(100)	Anzahl	Prozent	Anzahl(100)	Ana
Fächergr. insg.	1.860.698	868.336	46,67	1.930.923	912
Sprach- und Kulturwiss.	414.154	280.143	67,64	429.392	293
Sport	27.890	12.245	43,90	28.046	12

Fig. 5: ICE table with typical calculation

ICE standard tables

Each table produced with the Table Generation II method can be stored as a standard table. This means that each user can compile personal standard table collections. To store the request results as a standard table the user only has to click the appropriate button in the generation window.

Besides this, there are also the *standard table collections* which ICE makes available, maintains and administers, and which can be viewed under Directory on the start page. The structure of the topic areas allows quick access to the desired tables. The output of these tables is possible in various formats: *HTML, XML, Excel or PDF* and opens up a wide range of further processing and flexible data export options. Moreover, the central ICE collections as well as the collections compiled by users can be searched using a *keyword search* option.

ICE standard tables have a special format which allows them to be *updated* at the "touch of a button". This activates a special tool which automatically updates and extends the table. At present, three update types have been implemented which are described in further detail in Section 8 of this paper.

Standard table administration

This tool can be used to compile, rename or delete collections. The option of allocating reading and writing rights makes it possible to completely block collections for third parties or to allow them to read or to collaboratively process collections. Standard table administration thus makes it possible (if not blocked by the "rights allocation") to access all available standard table collections. This means that ICE standard tables can be used as a basis for individual table collections and can be further developed.

A number of functions are also available for individual tables. They can be deleted or copied into other collections, the table title can be changed or headers and footers added. Metadata from individual tables, such as topic area, data source or "rights allocation" can be read or changed using this tool, such as the type of updating.

6. System architecture

ICE was originally developed on HyperCard basis which meant that it was only accessible for computers running the MacOS operating system. Files were stored in the file system as were the metadata which described the corresponding file contents. Data were imported into the system using special programs which converted these from the supplied formats into the uniform ICE format and correspondingly adapted and complemented the metadata. The import routines were largely written in the programming language Fortran and implemented using Unix-Shell-Scripts.

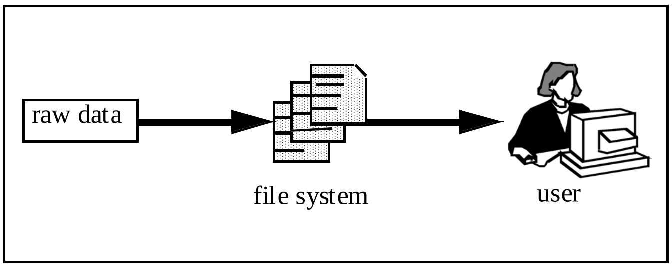


Fig. 6: HyperCard-based architecture

Following the restructuring of ICE into a web-based architecture, the import mechanisms were largely adopted as a first step, because the specifically-developed routines contain numerous correctness and plausibility tests on the raw data. The new architecture now takes the data which have been checked for plausibility and transferred into the file system and imports them with the help of new developments into a relational database, where further tests (in most cases technical) are carried out on the data.

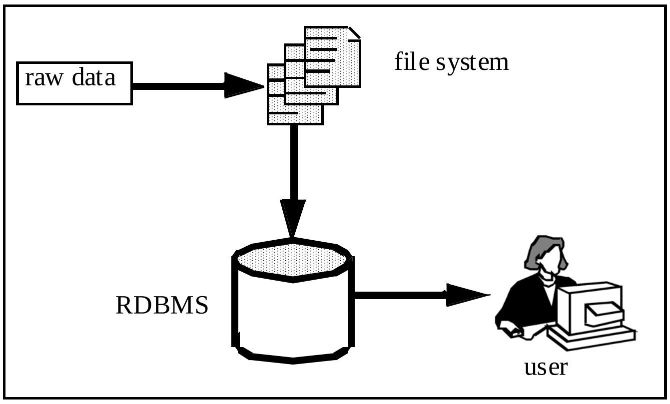


Fig. 7: Use of relational databases

The system was designed as a Multi-Tier-System, whereby an application server located between the database backend and the application level (Java Applets) converted abstract requests received by the system into requests which the database could understand, transferred these to the database, processed the results and transferred them back to the user. Apache was used as a HTTP server, communication between the applets and the application server was originally handled by low level socket connections which proved to be comparatively simple and stable.

Recently, this was changed over to HTTP in connection with Java Servlets. The Tomcat Server developed by the Apache Group has proven reliable as a Servlet Engine. The changeover to HTTP communications proved advantageous, especially in the context of the more recent XML developments for the conversion of request results into various file formats (eg, PDF) with the help of publishing frameworks (see Section 7). The changeover was also needed after the state ministries received a system in the form of ICE which had to be accessible on the Internet, ie, beyond the bounds of agency or company-wide Intranets. The firewalls for the ministry networks, which were generally restrictively configured for security reasons, did not allow communication via sockets. This would have required additional ports to be opened which, as a rule, would not have been permitted or, if so, then only after a great deal of persuasion.

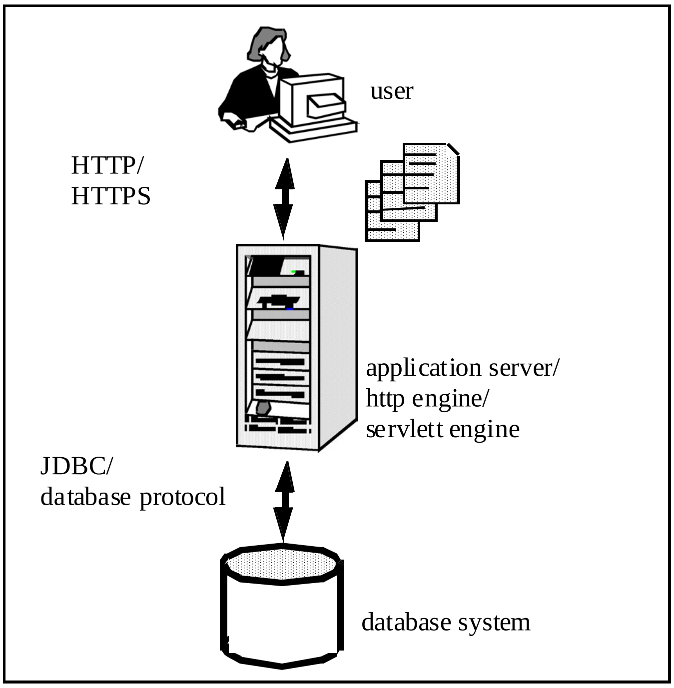


Fig. 8: Multi-tier-architecture of ICE

The decision in favour of Java as the development platform was made as early as in 1996. One goal in connection with the multi-tier decision consisted of shifting as many application tasks as possible from the client to the application server in order to, in particular, achieve better system scalability. The programming language Java from Sun Microsystems had hardly been tested in practice at the time; looking back, however, we can say that the decision to go for this environment was right, and not only in terms of platform independence. Indeed, Java has meanwhile developed into one of the most important programming environments.

However, the use of applet technology failed to relieve the workload of the client resources ("thin client" principle) to the desired extent, even though, instead of the more modern swing technology we chose an even leaner implementation with the help of the AWT (Abstract Window Toolkit), also because no more advanced technology was available at the time of the implementation of the client components. The presentation of request results in table format was realised with the help of the JKit/Grid class library from Objectshare. At the time of the relevant developments, this proved powerful enough to also depict interleaved tables, for example.

More recent considerations go in the direction of using two different requirement levels:

- Thin Clients, especially for ICE Internet applications: In such environments it is only possible to expect users to meet low hardware requirements, because these are hard to "check", since the users are more or less unknown. In such cases, Java Applets will be replaced by HTML/XML solutions, used on the server-side by Servlets, Java Server Pages (JSP) or Extensible Server Pages (XSP). This means that no Java code is run on the user computer and, consequently, there is no need for pre-installations on user computers.

- Modern Applet technologies using swing components, especially for Intranet applications. The circle of users is manageable in such environments and this is why certain hardware requirements and pre-installations can be expected of users (eg, a modern Java run-time environment). This makes it possible to run very powerful libraries for the presentation of the applications on the clients.

Communication between the database level and the application level is managed by JDBC-API (Java Database Connectivity). This makes it possible to incorporate SQL commands into the Java program components which are understandable for relational databases and to transmit these to the database; it also makes it possible to receive and process request results. A major advantage of this solution was seen in the fact that (insofar as standards are adhered to in the development) the database systems of various providers can be exchanged for each other practically without problem. Indeed, ICE is meanwhile used in production operations both with Oracle databases as well as with Informix and MySQL. The interface between the SQL sections and the DBMS systems is made by so-called JDBC drivers which are available for all common relational database systems. Besides the direct translation of SQL into database language, it is also possible to address proprietary database constructs, such as network components from Oracle or procedures stored in the database. However, we have only made use of these options when it was absolutely certain that porting to other database management system could be completely ruled out.

New questions arose in connection with the making available of ICE for users in various state agencies via the Internet. Such issues had not arisen to this extent in controlled Intranet environments. This applies, in particular, to data security/data protection and system availability. According to German law (from a data protection perspective), data within ICE do not require any particular protection, for they are not personal data. Nevertheless, economic aspects, aspects of state competition and other aspects would suggest the use of technical arrangements to prevent uncontrolled access to the data. This is why a group and role-based authentication and rights management system was designed and implemented which allows the finely-structured control both of access to the system and its various modules as well, and in particular, access to the analyses and interpretations generated by users (ICE standard tables).

For web-based applications, the system was made highly available by introducing multiple replication and further measures, and its performance was raised by using a simply extendable server cluster. Replication takes place both at database level as well as at application server level. A scheduler serves to distribute the requests, depending on the current load of the individual server knots on the computer with the least load. For reasons of cost, this solution made use of the open source database MySQL. Together with self-developed applications and further open components (Apache, Tomcat, etc.) this meant that is was possible to achieve quite an economic solution.

7. ICE Publishing Framework

The ICE Publishing Framework is a system component designed for the dynamic generation of (table) outputs in various formats. It is run *after* users have defined the logical structure of their desired analysis and have extracted the data from the database. At present, users can choose to output the tables in one of the following formats: HTML, XML, MS-Excel, Gnumeric, PDF. Since the PDF format is especially suitable for the production of print templates, several standard output formats are offered (eg, DIN A4 landscape, DIN A4 portrait). The architecture of the Publishing Framework, described in more detail below, means that further output formats (eg, addition of a logo, change of font, table types, paper formats, etc.) can be made available as necessary with very little effort. In addition, a layout tool was developed to let the user influence the PDF appearance of each table in a wide range of different ways. Among the aspects which users can influence are the font, font size, line spacing, column width, page breaks, etc.

The ICE Publishing Framework is based on the XML application platform Cocoon developed by the Apache Group. Cocoon is free software, which means: The source code is available (open source) and may be changed and extended. Licence fees neither need to be paid today nor in the future. Cocoon was chosen because it incorporates three key principles of modern software architecture:

1. *The strict separation of data, logic and layout*: Such separation makes work sharing easier in the development, maintenance and extension of the software. This separation differentiates Cocoon from other technologies such as Active Server Pages (ASP), as well as Java Server Pages (JSP).
2. *Consistent use of XML:* XML is an internationally-recognised standard used and supported both in the open source sector as well as in commercial fields, and has meanwhile asserted itself across a broad front. The main areas of application for the mega language are integration tasks as well as data exchange between applications and companies. The standardisation process for the key components has been completed (XML, XSLT and XML schema have successfully completed the W3C standardisation process and have the status of a "recommendation").
3. *Component integration*: All necessary XML components for the creation of a Publishing Framework are already contained in Cocoon, since it is possible to make use of the large collection of Apache projects. In ICE it is possible therefore to concentrate on the development of the application as such. For example, Cocoon uses Apache Xerces as an XML-Parser, Apache Xalan as an XSLT-Processor and Apache FOP for PDF generation. The architecture is based on the Apache project Avalon, a Java approach to component-oriented software development.

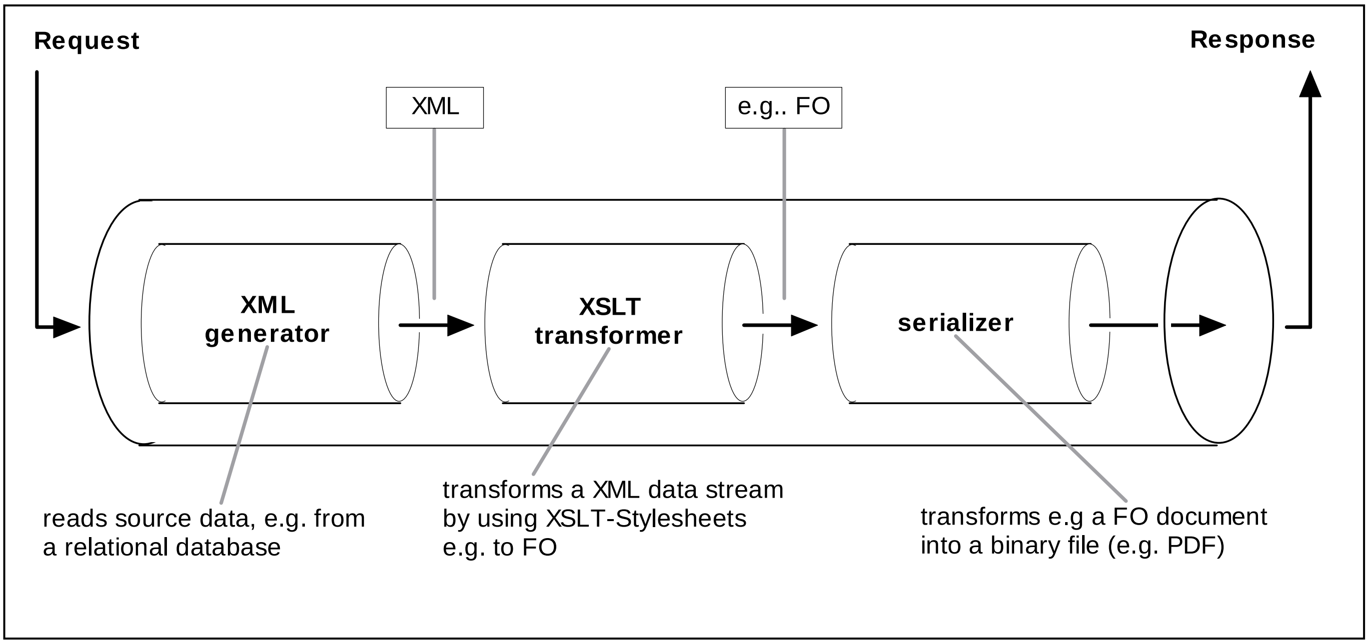


Fig. 9: Cocoon XML-Pipeline

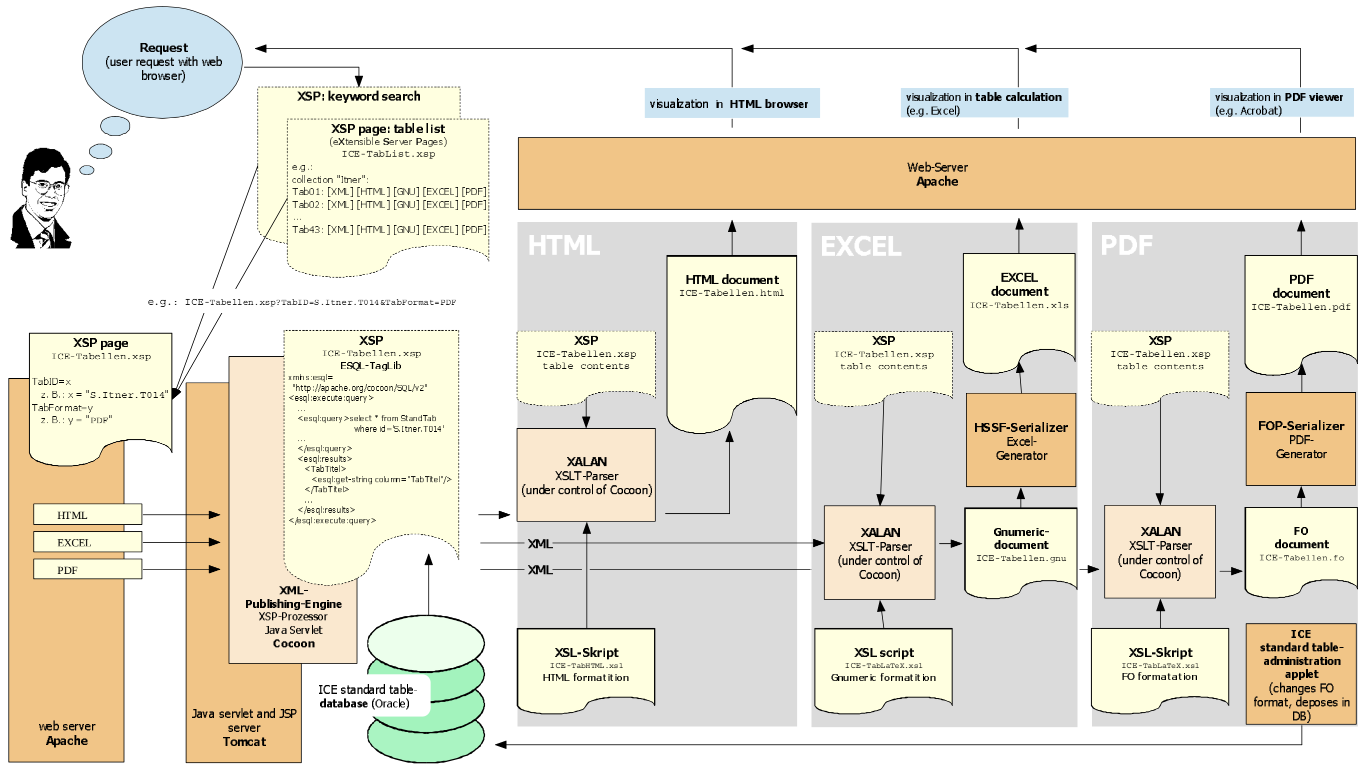


Fig. 10: ICE Publishing Framework

This has the following advantages for ICE:

- The Publishing Framework can easily be adapted to bring it into line with the existing infrastructure. Only very limited changes are necessary to the existing ICE.
- Very flexible, easily expandable system. Besides the presently available formats HTML, XML, MS-Excel, Gnumeric, PDF, it is possible with very little effort to create further formats (eg, WML for display on handhelds or mobiles/cell phones). For example, the generation of vector graphics or maps from an XML base file is also conceivable.
- Use of generally available standard software from the Apache Group (producers of easily the most frequently used web server worldwide).
- Use of international standards

Figure 10 illustrates the production process in the ICE Publishing Framework. The user request via a web browser, such as Netscape Navigator, Microsoft Internet Explorer or Opera browser ("Client"), marks the starting point. On a usual webpage, the user selects the XSP page (eXtensible Server Pages, called "XSP page: table list" in Figure 10) to be depicted from a list by using the table title for a table and the desired format. The one and the same XSP page is always requested via the eXtensible Server Pages. Although, possibly, thousands of standard tables are available in the system, they are requested via a single document ("ICE-Tabellen.xsp"). Upon receipt of the request, this XSP page, the table identification and the desired format are transmitted as parameters. XSP is an advanced development of the Java Server Pages (JSP). An XSP page is an XML document with dynamic content. This dynamism is achieved by means of directives defined in tags. In ICE, the XSP page calls up a program logic which, using the parameters communicated to the page, extracts the data required to present the tables (table title, remarks, footnotes, table values, etc) from the ICE standard tables database. Using the ESQL-TagLib it is easy to send appropriate SQL commands. The ESQL-TagLib is a further component of Cocoon. And so by accessing the database, the system gradually creates a well-formed XML document from an initially empty XSP page.

The generation of native spreadsheet formats represents a particular challenge, especially the only rudimentarily documented Excel format. The realisation made use of the POI Library. POI (Poor Obfuscation Implementation) is meanwhile a firm component of the Apache Jakarta Project. In the form of the HSSF Library (Horrible Spreadsheet Format - programmers seem to feel a certain degree of dislike towards the Excel format which they reimplemented) the POI provides an implementation of the Excel file format which makes the writing of simple tables easy. POI is fully integrated into Cocoon via the HSSFSerializer. It makes it possible to output XML documents which correspond as Excel files with the Gnumeric Spreadsheet Format in Cocoon. This is why, initially, the generation of Gnumeric-XML-files first needed to be implemented. And so ICE offers a further output format which, against the background of the emerging broad rejection of proprietary formats, will probably play an even greater role in the future as an open standard. At present, this format can be read above all by the ever more popular office packages OpenOffice.org (an efficient open source package) as well as its commercial counterpart Staroffice from SUN.

8. Self-updating standard tables

Tables produced with ICE tools can be stored locally on the client computer by converting them into HTML or MS Excel format or as so-called standard tables on the server. The advantage of storing a results table as a standard table lies in the fact that is can be updated or extended and overwritten with the latest data at a later date. The changes that are to be made when more current timepoints are available are defined when the standard table is compiled. The type of update of each individual standard table depends on the table structure, according to whether the time-based assignment of data is made up of only one timepoint (so-called cross-sectional table) or of a time series sequence. Users can choose from three *update types* at any time:

1. *Timepoint related update:* In this type of update, all the table data are replaced with the latest data available in the system (for example, the table contains the data for 2001 and these are replaced with data for 2002).
2. *Time series addition*: Depending on the table structure, more recent data are added to rows or columns, ie, the table grows with the addition of more recent timepoints.
3. *Time series shift:* Depending on the table structure, more recent data are added to rows or columns while simultaneously the corresponding number of older timepoints are chronologically left out, meaning that the table size remains unchanged.

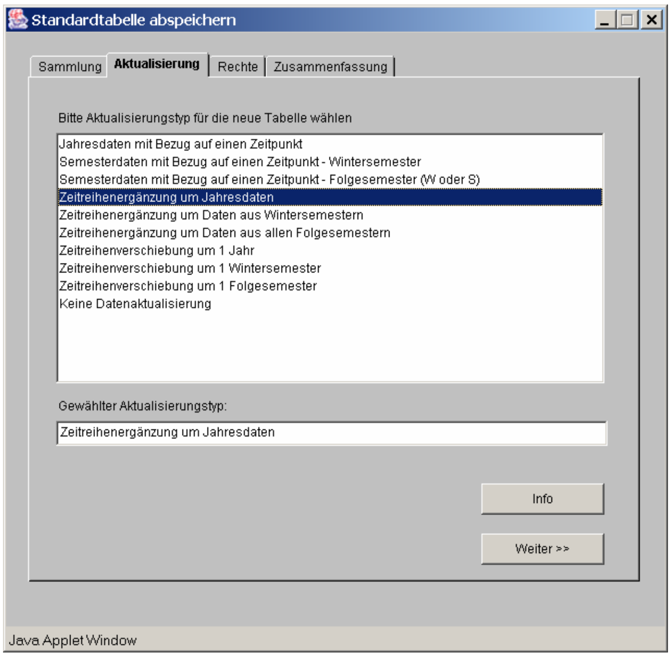


Fig. 11: Selecting an update type when storing a standard table

For each of these three update types it is additionally necessary to define whether the years, dates of the next winter semester or of the following semester (possibly also of the summer semester) are to be used. It is possible at any time to change the update time of a standard table (eg, after reaching a certain number of timepoints, the time series addition option can be replaced with the time series shift option), so as to influence the table appearance over the course of time.

Technical background:

A results table which is subsequently to be stored on the server as a standard table is produced from a so-called virtual data stock which the user accesses by entering keywords. Preselection in the form of virtual data stocks was chosen because the data contained in the database can, in principle, be combined in any which way. When generating the standard table, the system remembers the virtual data stock from which the data were extracted.

As far as the timepoint is concerned, each virtual data stock is registered and requestable for analysis and interpretation with the help of an ICE key. When importing new data into the system, the time information for the virtual data stock also changes. When the update of standard tables is triggered by pressing the appropriate button, the first step involves comparison of the latest timepoint in the results table with the original data stock. If the system finds that the virtual data stock meanwhile has more up-to-date time information, then the structure of the standard table is analysed and changed in accordance with the chosen update type. In the case of time series additions, the table structure is amended at the relevant places by adding the new timepoints. In the case of time series shifts and the updating of cross-sectional tables (as a special case of time series shift), the table structure remains unchanged. At one end, new timepoints are added while at the other end older timepoints are removed to make space. The data for the new timepoints are subsequently extracted from the database, while any old data which the updated table will continue to show are simply adopted. This avoids having to fetch table sections which had already been found at an earlier stage.

The advantage of automatically updating standard tables not only lies in the fact the table structure is automatically brought into line with the new timepoints. In addition, when a table is updated the new table sections are subjected to the calculations which had been defined for the old table. This modification of the updated table (percentage calculation, indexing, difference formation) also occurs automatically, therefore.

Further planned update types:

The currently-defined update types offer users the opportunity to maintain standard analyses over time and, as necessary, to change them easily. In the future, we intend to add further update types which make the updating of analysis tables even more flexible or will facilitate the updating of special table structures.

- *Time series with various qualities of data*: This update type will allow a table which is made up of several virtual data stocks and which uses data of varying degrees of data quality to be intelligently updated. Final data are already available for the older data in the time series, while for more recent timepoints only provisional data are available and for the most recent timepoint only flash reports. When updating such a table, the system checks whether the time series can be supplemented with new timepoints and, at the same time, whether data of poorer quality can be replaced with data of better quality.
- *User-defined time series amendments:* In the case of tables which have several timepoints, we intend to give the user more influence over the table timepoints so that it is possible to choose from all the available timepoints in the virtual data stock when updating a table.

9. Outlook

- Data

The utility of the system depends essentially on the data which it makes available. It would seem to be a law of nature of statistical data analysis that it is precisely those data or differentiations which are only incompletely available which are just now needed to answer current questions. This is why it is our constant endeavour to increase the data stocks and to achieve greater depth of structure. If, in the future, it becomes possible to use individual case data records more than has been allowed in the past, then this problem will probably also solve itself from that side.

- Internationalisation

HIS is currently thinking about extending the system by adding the option of outputting one and the same data stocks in various languages. Specifically, this involves enabling the publication of the German Ministry of Education and Research's Basic and Structural Data on the education system in several languages as a web application.

A second consideration goes in the direction of using the system for European comparison. To do this, we would need the appropriate data, of course, and the above-mentioned multilingualism would have to be implemented.

- ICE-Mobil

In the future, we want ICE not only to be available to users from a stationary workplace computer (or from an Internet café), but also to be available as a mobile system.

Access to elementary data via cell phone or WAP interface has already been implemented. A further interface is planned (Thin Client) with which access would also be possible via handheld computers (Palm, Pocket-PC, Sharp-Zaurus).

Sources

Dicken, Hans: JDBC – Internet-Datenbankenanbindung mit Java; Internat. Thomson Publ., Bonn u. a. 1997

Dicken, Hans; Hipper, Gunther; Müßig-Trapp, Peter: Datenbanken unter Linux; Thomson Publ., Bonn u. a. 2000

Birk, Lothar; Dicken, Hans; Kopp, Helena: Der goldene Schnitt – Statistische Datenauswertung mit Java. In: Java Spektrum 4/2000

Web resources:

Grund- und Strukturdaten des BMBF (Basic and Structural Data)
http://www.bmbf.de/pub/GuS2002_ges_dt.pdf

ICE der Wissenschaftsressorts der Länderministerien (ICE system for the science and education departments in the state ministries)
http://iceland.his.de

Wissenschaft Weltoffen website operated by the DAAD und HIS (ICE Data Basis)
http://www.wissenschaft-weltoffen.org

7. ICE Publishing Framework

Das ICE Publishing Framework ist die Systemkomponente zur dynamischen Erzeugung von (Tabellen-)Ausgaben in unterschiedlichsten Formaten. Sie kommt zum Einsatz, *nachdem* der Benutzer die logische Struktur seiner Auswertungswünsche festgelegt hat und die Daten aus der Datenbank geholt wurden. Derzeit können die Tabellen wahlweise in einem der folgenden Formate dargestellt werden: HTML, XML, MS-Excel, Gnumeric, PDF. Da das PDF-Format sich in besonderer Weise für die Erstellung von Druckvorlagen eignet, werden mehrere Standardausgabeformate angeboten (z. B. Din-A4 quer, Din-A4 hoch). Auf Grund der unten näher dargestellten Architektur des Publishing Frameworks können bei Bedarf mit sehr geringem Aufwand weitere Ausgabeformate (z. B. Hinzufügung eines Logos, Änderung bei den verwendeten Zeichensätzen, Tabellenstilen, Papierformaten usw.) verfügbar gemacht werden. Zum anderen wurde ein Layout-Tool entwickelt, mit dem der Benutzer in vielfältiger Weise Einfluss auf die PDF-Darstellung jeder einzelnen Tabelle nehmen kann. Beeinflussbar sind vom Benutzer unter anderem der verwendete Zeichensatz, die Zeichensatzgröße, die Zeilenabstände, die Spaltenbreiten, Seitenumbrüche usw.

Das ICE Publishing Framework basiert auf der von der Apache Group entwickelten XML Applikationsplattform Cocoon. Cocoon ist freie Software, das bedeutet: der Quelltext steht zur Verfügung (Open Source) und darf geändert und erweitert werden. Lizenzgebühren sind weder heute noch in Zukunft zu entrichten. Die Wahl fiel auf Cocoon, weil es drei wesentliche Prinzipien moderner Softwarearchitektur realisiert:

1. *Strikte Trennung von Daten, Logik und Layout*: Eine solche Trennung erleichtert die Arbeitsteilung in der Entwicklung und die Wartung und Erweiterung der Software. Diese Trennung unterscheidet Cocoon von anderen Technologien wie Active Server Pages (ASP) aber auch Java Server Pages (JSP).

2. *Durchgängige Verwendung von XML:* Bei XML handelt es sich um einen international anerkannten Standard, der sowohl im Open-Source-Bereich als auch im kommerziellen Umfeld anerkannt und unterstützt wird und sich inzwischen auf breiter Front durchgesetzt hat. Haupteinsatzgebiet der Metasprache sind Integrationsaufgaben sowie Datenaustausch zwischen Anwendungen und Unternehmen. Der Standardisierungsprozess der wesentlichen Bausteine ist abgeschlossen (XML, XSLT und XML-Schema haben den W3C-Standardisierungsprozess erfolgreich beendet und haben den Status einer "Recommendation").

3. *Komponenten-Integration*: In Cocoon sind alle notwendigen XML-Komponenten zum Aufbau eines Publishing Frameworks bereits vorhanden, da sich aus dem großen Fundus der Apache-Projekte bedient werden konnte. Im ICE konnte sich daher auf die die Entwicklung der eigentlichen Anwendung konzentriert werden. In Cocoon werden z. B. Apache Xerces als XML-Parser, Apache Xalan als XSLT-Prozessor und Apache FOP zur PDF-Generierung eingesetzt. Grundlage für die Architektur ist das Apache-Projekt Avalon, ein Java-Ansatz zur komponentenorientierten Softwareentwicklung.

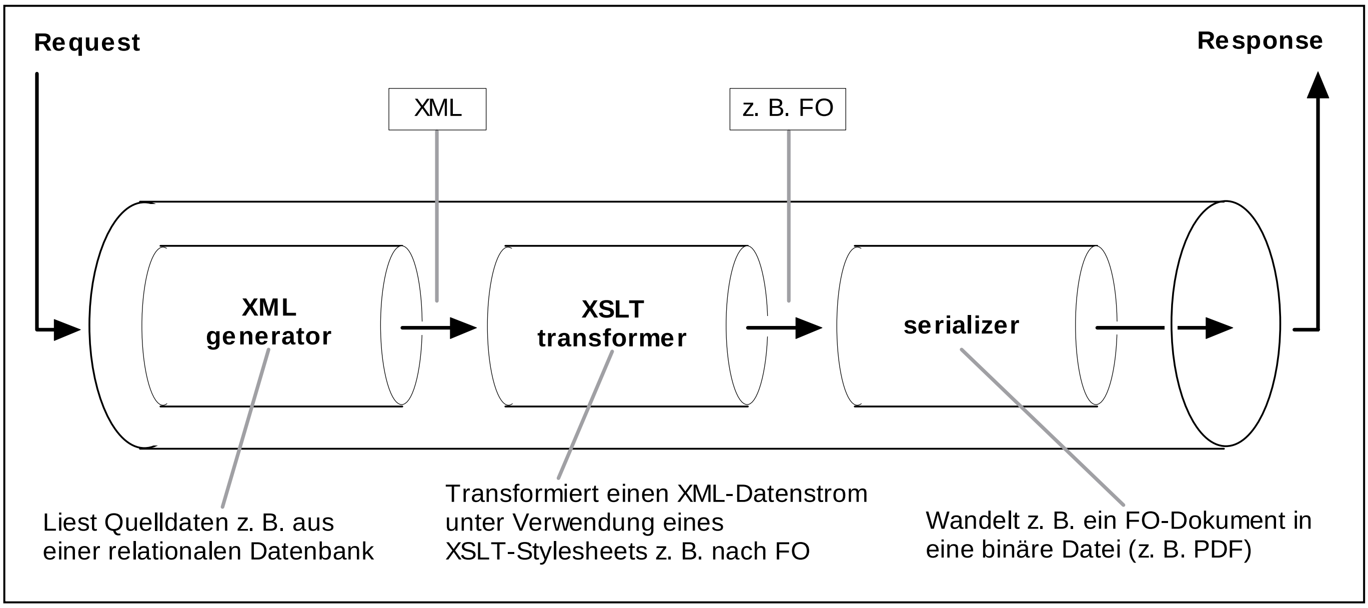


Abb. 9: Cocoon XML-Pipeline

ICE HIS

HIS - ICE

A Web-Based Information System to Support Higher Education Policy Decisions

Peter Müßig-Trapp, Hans Dicken, Helena Kopp
HIS - Higher Education Information System, Hannover, Germany

EUNIS, Amsterdam, July 2 - 4th 2003

ICE HIS

Structure

1. About HIS Hochschul-Information-System
2. ICE Overview and Main System Features
3. Using the System (Live Demonstration)
4. Outlook

Organisation

- History
  - Founded in 1969
  - 1975: Taken over by the federal government
- Funding
  - A Non-profit Organisation
  - Services free of charge
  - Results of studies and analyses are published

Mission and Services

- Software Developement
  Administrative computing applications for higher education
- Planning and Management Assistance
- Pool of Information
  - Empirical Studies
  - Higher Education Funding
  - Support of Desision Making

ICE History

- **~ 1990**: Proprietary Hypercard based system for MacIntosh
- **~ 1995**: Completely reengineering towards a platform independent architecture
  - intranet system
  - Multi-tier-architecture
  - RDBMS at the backend
  - Java as middleware
- **~ 2000**: Extending the System
  - Presentation- and exportmoduls
  - User authentication

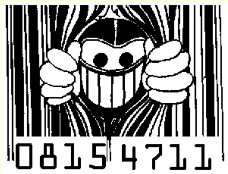
Information, Controlling, Entscheidung / Decision

In use at ...

- German Ministry of Education and Research (BMBF)
- Science and research ministries in ten German federal states
- German Science Council (Wissenschaftsrat)
- German Academic Exchange Service (DAAD)
- HIS

Protection of personal Data

- 1980s: Political and legeslative debate on the protection of personal data in Germany
- Restricted access to individual case data
- Use of **standard aggregated datasets**



Standard Aggregated Datasets vs. Individual Case Data

- Standard Aggregated Datasets allows only previously-defined analyses
- Individual case data records are much more flexible
- New development on behalf of Lower Saxony Ministry of Science and Education
- Now ICE can process both: **standard aggregated data and individual case data**

5 Central Goals

1. Make the best out of **standard agregated data records**
2. **Data Warehouse**: integrated and flexible analysing of different data stocks
3. **Platform independent** system
4. **Easy in use** (no computing skills or detailed data knowledge needed)
5. Combinated analyses of **individual case data and aggregated data records**

Main System Features

- **Web application**: Access via Java capable Webbrowser
- **Platform independence**!
  - **Operation System**:
    Server-side: Linux, Sun OS, MS Windows
    Client-side: All comonly-used platforms
  - **Database**:
    Commercial Databases: Oracle, Informix
    Open Source Databases: MySQL, PostgreSQL

Feature: Flexibility

- **Flexible data import**
  - Data with any structure and depth of structure can be imported
  - New topics can be included
- **Flexible data analysis**
  - Flexible table generator
  - Combining informations from several data stocks
- **Flexible data export**
  - HTML, MS Excel, PDF

Feature: Data Integration

- Data harmonisation with an **integrated key**
- **Centraly maintained** uniform ICE key
- **Equivalency rules**
- System known **key internal hierarchies**

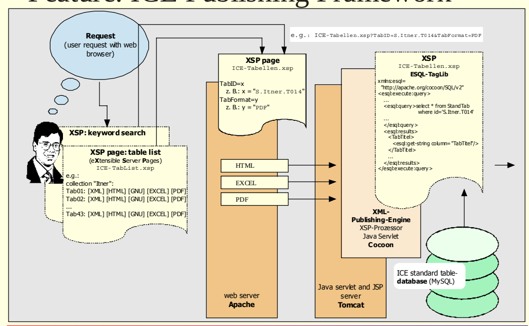
ICE 2. System Features HIS

Feature: ICE standard tables

- Analyses can be stored as ICE standard tables in standard table collections on the server
- **Integrated automatic updating** of ICE standard tables (or whole standard table collections)
- ICE standard table export formats: HTML, XML, MS-Excel, Gnumeric, PDF, ...

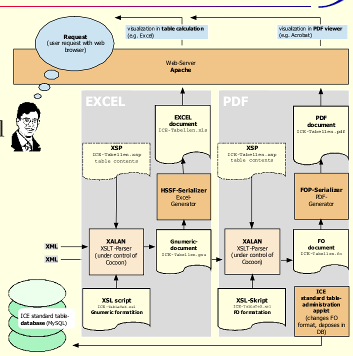
ICE 2. System Features HIS

Feature: ICE Publishing Framework



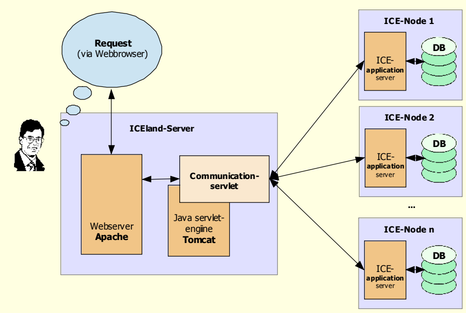
ICE 2. System Features HIS

Publishing Framework:
Generating Excel and PDF



ICE 2. System Features HIS

Feature: ICE Cluster

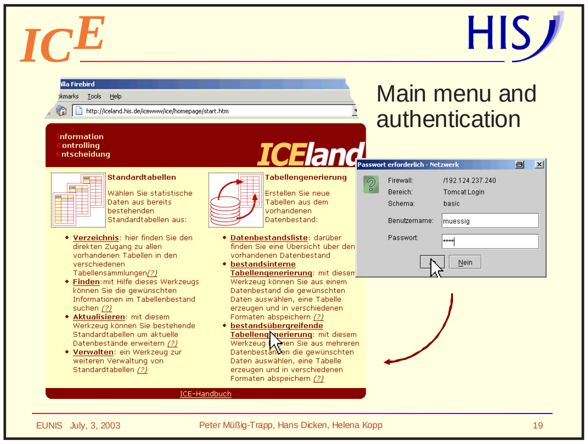


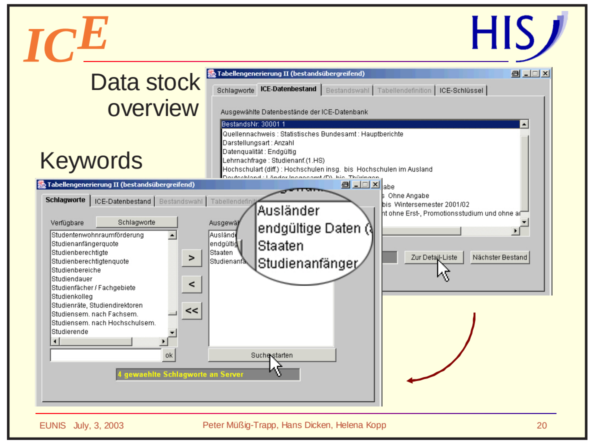
ICE HIS

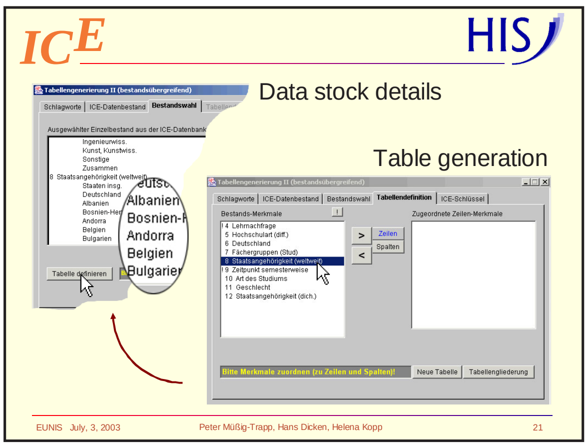
Live Demonstration of
ICE

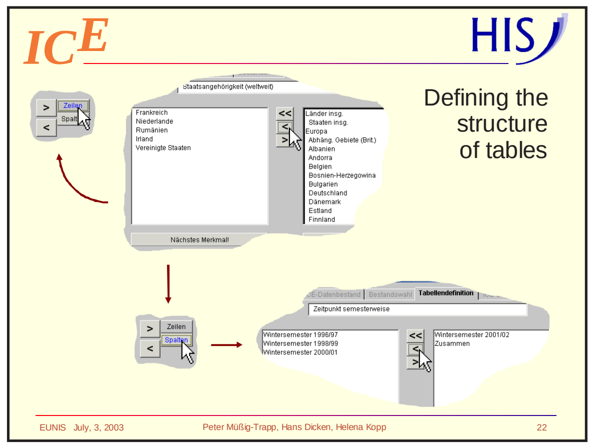
ICE HIS

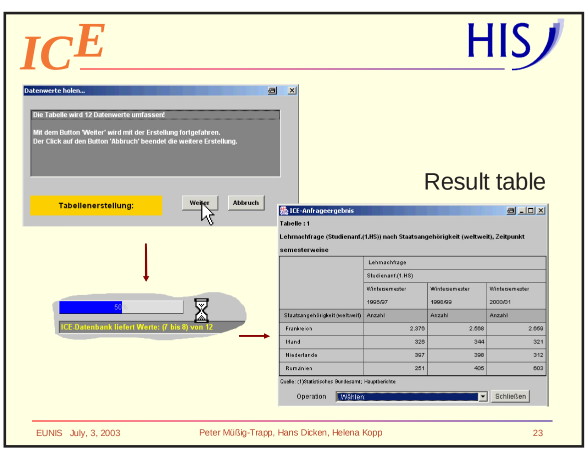
ICE in practical use

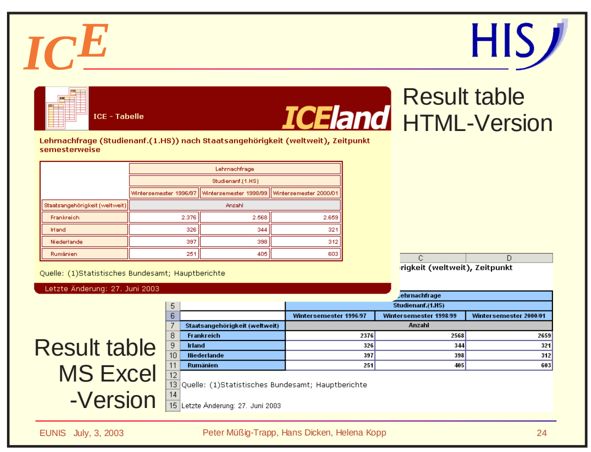

ICE
HIS
Main menu and authentication
ICEland
Standardtabellen
Tabellengenerierung
Passwort erforderlich - Netzwerk
ICE-Handbuch
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
19

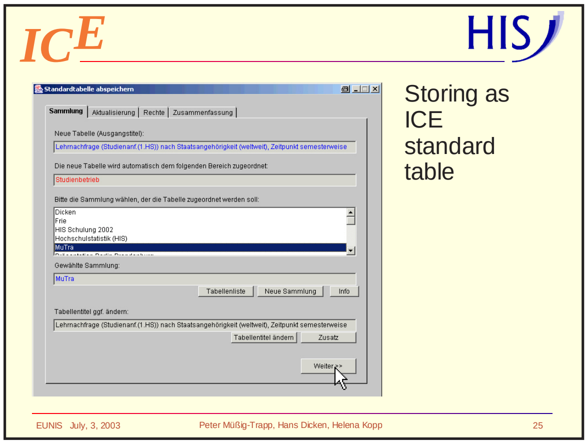

ICE
HIS
Data stock overview
Keywords
Tabellengenerierung II (bestandsübergreifend)
Ausländer
endgültige Daten
Staaten
Studienanfänger
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
20

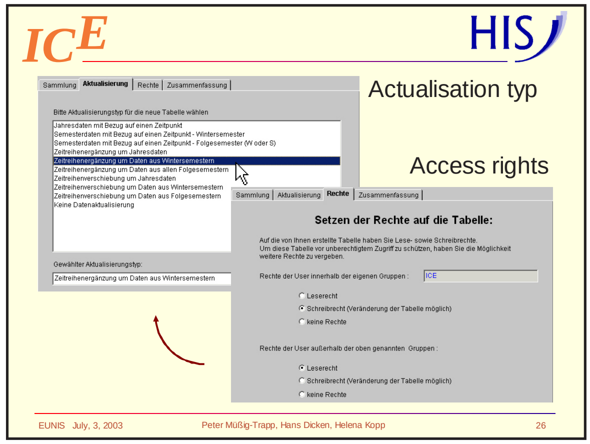

ICE
HIS
Data stock details
Table generation
Albanien
Andorra
Belgien
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
21

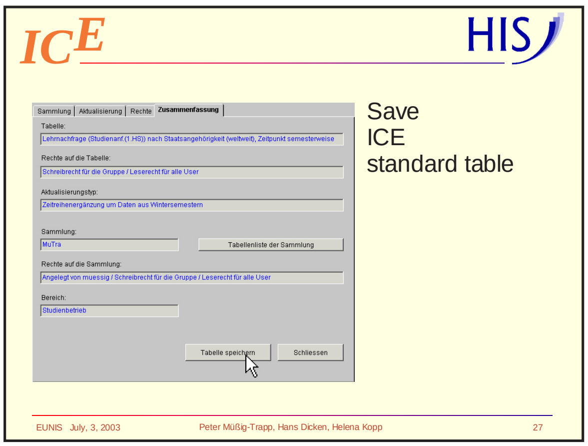

ICE
HIS
Defining the structure of tables
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
22

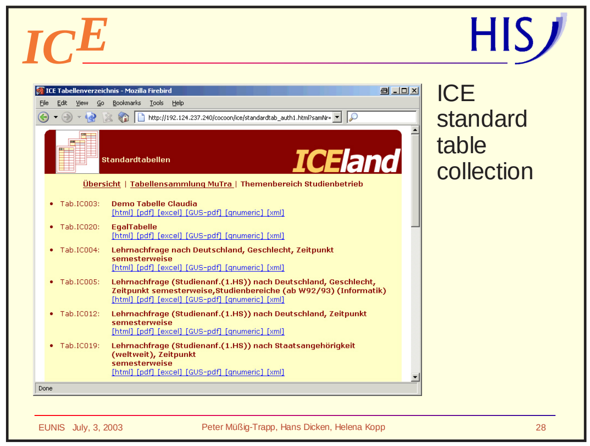

ICE
HIS
Result table
Datenwerte holen...
Tabellenerstellung:
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
23

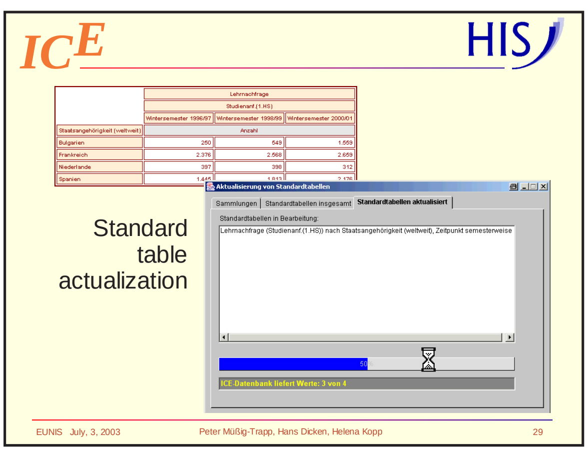

ICE
HIS
Result table HTML-Version
ICEland
Result table MS Excel -Version
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
24

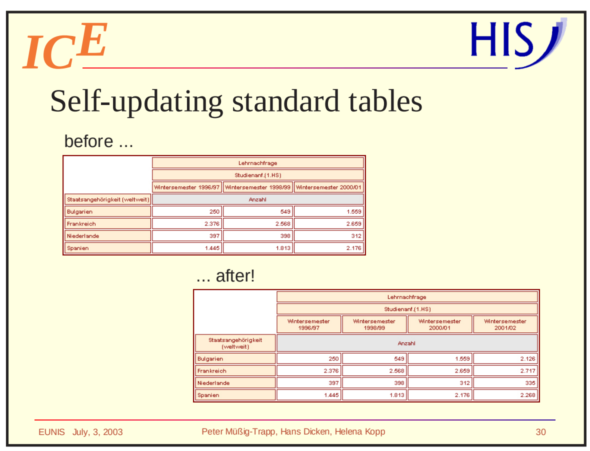

ICE
HIS
Storing as ICE standard table
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
25


ICE
HIS
Actualisation typ
Access rights
Setzen der Rechte auf die Tabelle:
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
26


ICE
HIS
Save ICE standard table
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
27

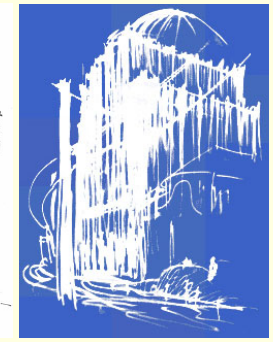

ICE
HIS
ICE standard table collection
ICEland
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
28

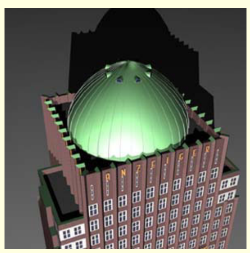

ICE
HIS
Standard table actualization
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
29

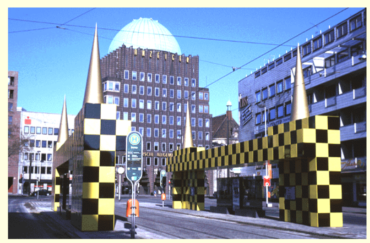

ICE
HIS
Self-updating standard tables
before ...
... after!
EUNIS July, 3, 2003
Peter Müßig-Trapp, Hans Dicken, Helena Kopp
30

Outlook

- **Data**
  - Increase the data stocks
  - Use of individual case data records
- **ICE-Mobil**
  - Access via handheld computers (Palm, Sharp Zaurus, Pocket PC)
- **Internationalization**
  - Multilingual output
  - European comparison







Oldest skyscraper in Hannover

The building is rated as a historic monument by the city of Hannover

Built: 1927 / 1928

Architect:
Fritz Höger

Roof: 50m / 164ft

Floors: 12

Headquater of HIS