

This document includes some recent decisions of the EPO in 2014 with regards to software related inventions and shows relevant extracts from the respective decisions.

T 0426/09 (Alert notification/SAP) of 9.9.2014

Alert Notification Management

**Inventive step - (main request Inventive step - no)
Remittal for additional search (auxiliary request)**

Application number: 04101335.0
IPC class: G06F 17/60
Applicant name: SAP SE

Board: 3.5.01

<http://www.epo.org/law-practice/case-law-appeals/pdf/t090426eu1.pdf>

European patent application number 04 101 335.0 (publication number 1 475 734) relates to a method of informing parties of alert situations in business or industrial applications, alert situations like the termination of a contract, a delivery problem for an order, a decline in revenue, or some technical problems in a system.

Claim one reads as:

"A method of transmitting alert notification messages, the alert notification messages corresponding to alerts, the alerts being indicative of exceptional situations in at least one industrial application, comprising the following steps:

collecting (S1) the alerts from at least one industrial application (#5-1, #5-2, ..., #5-n),

generating (S2) at least one alert notification message, and

sending out (S3) the at least one alert notification message to at least one recipient (#1, #2, ..., #n) through at least one transmission channel (MC),

whereby alert collecting, alert notification message generating and sending out is controlled (S4) according to an alert notification profile (1-1),

wherein the alert notification profile (1-1) is administrated in a centralised engine, and is configured to comprise information relating to at least one of:

a group of recipients;

a time schedule for transmission;

a channel of transmission (MC);

a method to create text out of the alert."

Auxiliary request I adds the following feature to the end of claim 1:

"and wherein a plurality of alerts are aggregated in the centralised engine; and whereby the alerts are stored in a database, whereby the alerts are represented as objects, each object corresponding to a one of a number of configurable alert types, wherein for each alert type a scheme of database tables and corresponding code segments to access the database tables are generated."

The search division issued a no-search declaration under Rule 45 EPC 1973, indicating that it was not possible to carry out a meaningful search since no technical problem could be established which had required an inventive step to overcome. In the further course of the grant procedure, the examining division refused the application for lack of inventive step, objecting essentially that the invention was merely a common computer implementation of a business method.

The appellant criticised that neither the search division nor the examining division had made a search for relevant prior art. A notoriously known computer system was certainly not the closest prior art for a method claiming a new and unique combination of processing steps. Without prior art, however, it would be impossible to discuss inventive step.

As an initial point, the appellant has argued that a discussion about inventive step is impossible as long as no state of the art has been cited. However, one piece of prior art has been cited, viz. a conventional data processing system (decision, point 4.2 d)), for which the examining division - correctly - assumed that no proof was required. Certainly an assessment can be based on this prior art. **Whether or not there is closer prior art is for this particular issue irrelevant.**

Regarding the main request, the Board considers the decision under appeal to be correct. **Collecting information** - even information about industrial (including technical) applications - and notifying a group of interested recipients of an alert situation via a suitable channel of communication using a notification profile, i.e. for example a forwarding or dispatching list of recipients, **is essentially a non-technical communication process that is per se not able to provide a contribution to inventive step**. The **implementation** of such an alert notification process **as a software application in a notorious network system** comprising a central network server does **not involve any technical aspect** that may **qualify as inventive**.

The appellant has argued that "each and every feature of claim 1 has technical character". The Board does not deny that the method may work automatically. The point is however that it represents, **in the Board's view, a straight-forward implementation, using commonplace technical features, of a non-technical communication process. "Generating" a message, for example, is technical for the sole reason that it is done by a computer**. But the aim of the message is the usual one of conveying information, which is non-technical. Also "collecting" alerts is nothing more than an automatized version of a human collecting information, and the generation of messages nothing more than its expression.

For these reasons, the decision under appeal is confirmed in respect to the main request.

Considering auxiliary request I, it is noted that claim 1 has been **enriched by technical features which are neither common implementation steps nor the mere logical consequence of the alert notification or a similar communication process**. The alerts are stored in a database as objects, each object corresponding to one of a number of configurable alert types, and a scheme of database tables and corresponding code segments to access the database tables are generated for each alert type. The difference between this request and the main request is thus the explicitly claimed database and the relatively detailed way data is stored in it. The appellant has argued that these features **permit alerts of unknown type to be stored, providing a "self-expandable" database**. In the Board's view **these arguments cannot be convincingly dealt with without knowledge of the way databases are normally built. Since a prior art search has not yet been performed it is thus necessary to remit the case to the examining division for carrying out an additional search and for restarting the substantive examination on the basis of the present auxiliary request I and the results of the additional search.**

T 0963/09 (Selective auditing/ORACLE) of 5.6.2014 **Selectively auditing accesses to rows within a relational database at a database server**

Inventive step - main request (yes)

Application number: 01908864.0
IPC class: G06F 17/30
Applicant name: Oracle International Corporation

Cited decisions: G 0010/93

Board: 3.5.07

<http://www.epo.org/law-practice/case-law-appeals/pdf/t090963eu1.pdf>

The invention

The invention aims to enable selective auditing of accesses to tables of a relational database system, said system comprising a database server and a client. This is achieved by including in each table an "auditing flag" indicating whether auditing is enabled for that table. Upon receipt of a database query from the client, the database server checks the auditing flags of the tables accessed by the query to see for which tables the access should be audited. If auditing is enabled, the database server modifies the query by inserting into the query "monitoring logic" which causes audit records to be created "for rows in relational tables that are accessed by the query and that satisfy an auditing condition". The modified query is then processed and the query result is returned to the client. The selectivity of the auditing resides in the use of the auditing condition.

Examples of recent 2014 Board of Appeals decisions related to Software Innovations

Two examples given in the application of auditing in accordance with the invention are:

- the creation of an audit record for any row of a database table satisfying either the auditing condition "salary > 1,000,000" or the auditing condition "title = 'CEO'" (page 5, line 31, to page 6, line 5); and
- the creation of an audit record for any row of a database table that is returned by a query and which satisfies the auditing condition "DEPT = 'SALES'" (page 7, lines 22 to 27, and page 8, lines 28 to 30).

According to the background section of the application, conventional database systems typically provide a general auditing facility that records an audit trail containing general information about the user and the query issued. These auditing facilities record information only as to which tables are accessed, not whether certain records inside a given table are accessed. This table-level auditing tends to generate a large number of false audit records, because many accesses to a given table do not touch sensitive data.

The background section further points out that in certain distributed database architectures, auditing is implemented in applications located on application servers rather than at the database server, which may render it almost impossible to ensure that each application is configured to perform the auditing properly.

Claim 1 of the main request reads as follows:

"A method for selectively auditing accesses to a relational database system (109), comprising:

receiving a query (123) from a client (102) at a database server (110) that processes queries for the relational database system, wherein the relational database system comprises a plurality of relational tables (113), and each of the relational tables includes an auditing flag (206) to indicate whether auditing is enabled for the relational table;

determining whether auditing is enabled by checking all of the tables referenced by the query to see if an auditing flag is set for the tables, and if so, modifying the query prior to processing the query by inserting monitoring logic into the query for causing an audit record to be created and recorded for rows that satisfy an auditing condition;

processing the query at the database server to produce a query result, wherein processing the query causes an audit record to be created only for rows in the relational tables that satisfy the query conditions and are accessed by the query and that satisfy the auditing condition;

recording the audit record in an audit record store (118); and

returning the query result (124) to the client."

Prior art introduced by the Board

The Board considers document D3 to represent a suitable starting point for the invention. It discloses a database system comprising a database server and clients. The database server

audits select statements on tables for which auditing has been enabled. The example on page 31-8 discloses a plurality of relational tables ("emp" and "dept"). The subject-matter of claim 1 differs from the auditing method of document D3 in that:

- each of the relational tables includes an auditing flag to indicate whether auditing is enabled for the table;
- upon receipt of a query, it is determined whether auditing is enabled by checking all of the tables referenced by the query to see if an auditing flag is set for the tables;
- if auditing is enabled, the query is modified prior to being processed by inserting monitoring logic into the query for causing an audit record to be created and recorded for rows that satisfy an auditing condition;
- processing the query causes an audit record to be created only for rows in the relational tables that satisfy the query conditions and are accessed by the query and that satisfy the auditing condition.

In the communication accompanying the summons, the Board observed that **auditing of database accesses, while in itself a technical operation, in the context of the present invention appeared not to serve any specific technical purpose going beyond the act of auditing.** Similarly, the **motivation for making auditing selective**, i.e. limiting auditing of database accesses to accesses of rows satisfying a particular auditing condition, **appeared to be non-technical.**

The Board need not decide whether it maintains this view, as it finds, as explained below, that the claimed implementation of selective auditing involves an inventive step.

According to claim 1, selective auditing is implemented essentially by the insertion into the query of monitoring logic that, upon processing of the query, causes an audit record to be created only for rows satisfying the auditing condition. In the communication accompanying the summons, the Board noted in respect of the corresponding feature of claim 1 of the then main request that auditing of a select statement caused the addition of audit records to an audit table, i.e. a further database action. It expressed the view that it was an obvious possibility to perform such database actions using a suitable database query either as part of the user's query or as a separate query. This effectively resulted in the auditing being performed by "inserting monitoring logic into the query". Reference was made to page 6, lines 16 to 26, of the description.

At the oral proceedings, the appellant explained that the claimed invention allowed row-based selective auditing to be performed based on an auditing condition that referred to fields that were not included in the query result returned to the client.

Taking the first example above, if the query takes the form of a select statement that selects the "salary" column and, for example, a column specifying the name of the employee, but does not select the "title" column, then auditing based on the condition "title = 'CEO'" is not possible by first executing the query and then filtering the query result to determine the rows that additionally satisfy the auditing condition.

In view of this explanation the **Board accepts that the claimed solution to the problem of implementing selective auditing cannot be regarded, without documentary evidence, as a mere obvious possibility**. Since, moreover, none of the documents on file disclose the insertion of monitoring logic into a query, the Board concludes that the subject-matter of claim 1 and that of corresponding independent claims 6 and 7 **involves an inventive step**.

T 0157/10 (Automatic language selection/BLACKBERRY) of
16.7.2014

Automatic language selection for improving text accuracy

Claims - clarity and support (yes Claims - after amendment)

Extension of subject-matter (no)

Inventive step (yes)

Application number: 05256395.4
IPC class: G06F 3/023
Applicant name: BlackBerry Limited

Board: 3.5.05

<http://www.epo.org/law-practice/case-law-appeals/pdf/t100157eu1.pdf>

Claim 1 of the appellants main and sole request reads as follows:

"A method for selecting (126, 132, 136, 160, 202, 216) a language from a number of languages used by a text disambiguation function executed by an electronic device (4) during a text entry session, wherein recipients of said text entry each have a primary language tag and a secondary language tag associated therewith, the method comprising:

determining a context for said text entry session based on the primary and secondary language tags; and

responsive to said determining, selecting (126, 132, 136, 160, 202, 216) automatically a language used by said text disambiguation function during said text entry session; and:

said selecting (126, 132, 136, 160, 202, 216) a language comprises:

choosing, from the primary and secondary language tags of the recipients, the language associated with the language tag common to the greatest number of recipients."

The procedure for selecting a language for the disambiguation routine takes account of the primary and secondary language preferences of the intended recipients.

The appellant made submissions in its written statement to the effect that D1 only disclosed a single language identifier associated with the user information. In particular, the appellant submitted that D1 did not disclose or suggest multiple recipients of text entries each having a primary and a secondary language preference and referred to [0024] of D1 according to which the user information only has a single language identifier. In cases involving multiple recipient languages, D1 taught manual rather than automatic language selection and a skilled person would therefore not have even considered the existence of a secondary language of the message recipient. Even if they had done, D1 did not disclose or suggest the very specific method of choosing from the primary and secondary languages of the recipients the language common to the greatest number of recipients. On this basis, the appellant submitted that the skilled person would not have arrived at the claimed invention starting from D1.

The present claim 1 incorporates amendments in response to the observations made by the board in its communication and, in particular, includes a specification to the effect that each of the intended recipients of a text entry has an associated primary language tag and a secondary language tag.

The board judges that in the context of a method executed by an electronic device for selecting a language to be used by a text disambiguation function, the contact information fields or "language tags" associated with a recipient can be considered to represent technical features of the invention insofar as they constitute data stored in electronic format on the device and used by the device to automate the process of language selection.

Inventive step

The claimed method is based on the provision of a "primary language tag" and a "secondary language tag" associated with each of the recipients of a text entry whereby the primary and secondary language tags are taken into consideration when selecting a language to be used by the disambiguation routine.

The underlying technical problem may be formulated as how to automate the selection of a language from a number of languages available for a disambiguation routine when preparing text entries for a plurality of recipients who do not necessarily share a common language preferences.

The board concurs with the appellant's submissions to the effect that neither D1 nor any of the other available prior art documents disclose or suggest the association of multiple language identifiers with each of a plurality of recipients and the consideration of multiple language identifiers when selecting a language to be used by a disambiguation routine. D1 in particular only discloses a single language identifier associated with a user (cf. D1: [0024], emphasis added: "... the language for the word recognition, utilised during message editing, is then selected based on the language identifier associated with the user information.")

Whereas the provision of additional language identifiers or "tags" for each recipient may, in itself, represent a relatively straightforward measure from a technical point of view, the board judges that, in the given context, there is no identifiable basis for concluding that, in the light of the available prior art, it would have been obvious for the skilled person to consider extending the teaching of D1 in this manner. In this regard, the

board notes that D1 makes no mention of the problem of automating language selection when preparing text entries for a plurality of recipients who do not necessarily share a common language preference and there is **likewise no disclosure or suggestion of said problem or any hint towards the claimed solution in any of the other cited prior art documents.**

Hence, although it can be argued that the skilled person **could have** provided an additional "secondary language tag" **without undue difficulty, there is no apparent reason as to why he would have been motivated to do so** having regard to the available prior art which neither discloses nor suggests taking multiple language identifiers into consideration when selecting a language to be used by a disambiguation routine.

Hence, the board takes the view that the subject-matter of claim 1 of the main request cannot be derived in an obvious manner starting from D1. In view of the foregoing, the board concludes that the subject-matter of claim 1 of the main request involves an inventive step.

T 0936/11 (Image view server/E-PLUS CAPITAL) of 30.9.2014 Network image view server using efficient client-server, tiling and caching architecture

Claims - clarity after amendment (yes)

Inventive step - (yes)

Application number: 98906484.5
IPC class: G06F 17/30
Applicant name: E-Plus Capital Inc.

Board: 3.5.07

<http://www.epo.org/law-practice/case-law-appeals/pdf/t110936eu1.pdf>

The invention

The invention relates to viewing large digital document images using a client-server architecture and in particular using a standard web browser as image viewer. Instead of copying a whole image file from the server to the client, the invention proposes dividing the image into tiles at the server and transmitting only those tiles that make up the portion of the image requested by the client. This is implemented by letting the web server, in reply to a request for a portion of an image, transmit to the web browser an HTML output file that describes "the order, position and hyperlink for each view tile to be displayed" (page 10, line 29, to page 11, line 1, of the application as published).

According to the description on page 2, prior-art methods for viewing images stored on a server used proprietary workstation application software to access an image file server and transferred a copy of the whole image file from the image file server to the client workstation. These methods had disadvantages such as inefficient use of the network, high software cost

per workstation, high computational demand on the workstation and limited availability of the proprietary software for different workstation types. The invention made efficient use of the network, resulted in greater speed of image display and minimised the computing resources required by a client. The advantages of the invention were further increased by the use of caching mechanisms.

Claim 1 of the main request reads as follows:

"Computer network server (100) adapted to store digital document image files,

- said computer network server (100) being programmed with Web server software
- to receive requests from a client Web browser (10, 20) in URL code, wherein the URL code identifies the image file to be viewed, the scale of the view and the region of the image to view,
- to compose the requested view, and
- to transmit HTML code for the resultant view to the client Web browser (10, 20) to display;
- said computer network server (100) comprising a foreground view composer (50), which is adapted
- to interpret the view request received from the client Web browser (10, 20),
- to determine a grid of view tiles of the image according to the requested scale of the view,
- to determine an array of view tiles which are needed for the requested view, wherein the array of view tiles is selected from the grid of view tiles,
- to create the view tiles (160) needed for the requested view, wherein the foreground view composer (50) takes the needed view tiles from a cache (60) or creates the needed view tiles from the image file (90), and
- to create an HTML output file, wherein the HTML output file references the view tiles needed to display the completed view and describes the order, position and hyperlink for each view tile to be displayed."

Clarity objections raised in the decision under appeal are no longer applicable.

In particular, the feature "to compute a grid of view tiles of the image" now reads "to determine a grid of view tiles of the image according to the requested scale of the view". This clearly expresses that this step merely determines a tiling of the image into a grid of view tiles on the basis of the requested scale of the view.

The feature "to compute an array of view tiles, which are needed for the requested view" has similarly been amended to read "to determine an array of view tiles which are needed for the requested view, wherein the array of view tiles is selected from the grid of view tiles", which now also clearly expresses the relation between the "array" and "grid" of view tiles.

The feature "to compose the array of view tiles needed for the requested view" has been replaced by "to create the view tiles needed for the requested view, wherein the foreground view composer takes the needed view tiles from a cache or creates the needed view tiles from the image file". In addition, the reference to "the view composition" has been removed from the feature "to create an HTML output file to describe the view composition to the client". It follows from the claim as currently worded that the requested view is "composed" by the array of view tiles.

Document D1 as closest prior art

Document D1 discloses an image server for serving watermarked images to client web browsers (see abstract). The server is programmed with web server software (page 1126, left-hand column, last paragraph). The server receives requests from a client web browser in the form of a URL encoding an image name, output style such as thumbnail or full-size, and optionally a graphic format (page 1127, left-hand column, last paragraph, to right-hand column, first paragraph). Upon receipt of a request, the server loads the file into memory, processes it (sections 5.5, 5.7 and 5.8 discuss watermarking, graphic format conversion, scaling to thumbnail format), and delivers the resulting image to the browser (page 1128, left-hand column, second paragraph).

One of the output styles supported by the image server of document D1 is the "info" output style discussed on page 1133, left-hand column, first paragraph. When an image is requested with output style "info", HTML code is returned to the browser defining a full HTML page consisting of the title of the image, an inlined thumbnail of the image which is a link to the full-sized image, and copyright and author/title information.

The subject-matter of claim 1 essentially differs from the image server of document D1 in that:

- the URL specifies a view of the image file in terms of a scale and a region;
- based on the view, the web server determines an array of view tiles that are needed for the requested view and creates said needed view tiles from the image file and/or retrieves said needed view tiles from a cache; and
- the web server creates an HTML output file including appropriate formatting and references to the needed view tiles.

These **distinguishing features achieve an efficient loading of portions of images, which is especially important for viewing of large images**, i.e. images that cannot be displayed in full. Efficiency is not only **achieved by downloading only the portion of the full image data corresponding to the selected "view", but also by partitioning the image into tiles**. Caching "view tiles" both at the server (as expressed by the claim) and at the client (in accordance with standard functionality of web browsers) results in a **reduction of bandwidth usage** when the same client browser and/or different client browsers request overlapping "views" of the same image file.

The **objective technical problem may therefore be formulated as how to adapt the image server of document D1 to allow efficient viewing of large images**.

According to the Examining Division, document D3 disclosed that the image was tiled, and that loading the image was faster. The skilled person knew that tiling and transmitting all the files slowed down the download. In order to nevertheless achieve faster loading, it must have been a subset of the image that was downloaded to the client. This aspect was therefore implicitly disclosed.

The Board notes that **what a document implicitly discloses, i.e. what can be directly and unambiguously derived by the skilled person using only his common general knowledge, is not the same as what can be speculatively deduced as a possible explanation for an apparent technical inaccuracy identified in the document.** The Board considers that document **D3 does not disclose transmitting only a subset of the set of tiles that make up the entire image.**

The Board further notes that transmitting only a subset of the set of tiles is also not, as might be argued, an inevitable consequence of the fact that a sufficiently large image cannot be fully displayed in a limited display area. The normal way of dealing with this problem, especially in the context of web browsers, is to display the entire image in a scrollable area.

6.5.6 The Board is therefore of the view that claim 1 is **not rendered obvious by a combination of documents D1 and D3.**

Regarding the combination with further prior art documents please refer to the full decision.

T 1203/11 (Automated medical testing/HOFFMANN-LA ROCHE) of 1.7.2014

A system and a method for managing information relating to sample test requests within a laboratory environment

Clarity - (yes, after amendment)

Inventive step - (yes, after amendment)

Applicant name: F.Hoffmann-La Roche AG Roche Diagnostics GmbH

Application number: 06011422.0

IPC class: G06F 19/00

Board: 3.5.05

<http://www.epo.org/law-practice/case-law-appeals/pdf/t111203eu1.pdf>

The present invention concerns an automated data management system for testing medical samples in a laboratory environment based on a recursive workflow. The medical data management system is made up of a "host", a "decision unit", at least one "pre-analytical unit", at least one "analytical unit", and optionally at least one "post-analytical unit" (see e.g.

Fig. 1 of the present application). According to the application, the problem to be solved by the present invention is to provide a medical testing system for realising an advanced sample workflow with reduced system complexity and improved laboratory quality enabling a satisfying work environment (cf. page 1, line 31 to page 2, line 5 and page 18, lines 29-31 of the application as filed).

The recursive workflow underlying the present invention (cf. page 4, line 29 to page 6, line 8 and page 10, line 15 to page 12, line 6 of the application as filed), in particular claims 1 and 8, includes the following steps:

1) issuing, by a host, a sample order (including a sample identifier and a list of sample test requests) for a sample to be tested to a decision unit;

2) transporting the sample to a pre-analytical unit;

3) identifying, by the pre-analytical unit, the transported sample and providing the sample identifier to the decision unit;

4) providing, by the decision unit, sample information (i.e. pending sample test requests, sample target information) based on the sample order associated with the obtained sample identifier to the pre-analytical unit;

5) sorting, aliquoting, and archiving, by the

pre-analytical unit, the sample based on the obtained sample information and providing

pre-analytical sorting/aliquoting information (i.e. rack identifier and rack position associated with the analytical unit where the pre-processed sample is to be tested) to the decision unit;

6) transporting the sorted/aliquoted sample to the respective analytical unit according to the

pre-analytical sorting/aliquoting information;

7) providing, by the decision unit, an extended sample order (including sample testing information) to the respective analytical unit;

8) performing, by the analytical unit, the tests requested according to the sample order and providing the test results to the decision unit;

9) updating, by the decision unit, the sample order based on the test results and optionally adding new test requests;

10) performing steps 2) to 9) until all the test requests have been processed, wherein

10a) the next pre-analytical step is decided by combining current analytical data with other sample-related information;

10b) the sample is transported to the respective analytical unit based on pre-analytical information and on test results from test requests which have already been performed;

11) providing, by the decision unit, a report on the test results to the host.

2.2.2 The board concurs with the finding of the decision under appeal that D2 represents the closest prior art, since it is related to the same purpose as the present invention, namely to automated remote testing of medical samples in a laboratory environment upon request of a host ("remote client 100").

The board considers that D2 discloses the following limiting features of claim 1:

A system for realising a workflow for performing a number of tests to be made on a sample ("specimen 102") within a laboratory environment ("automation lab"; see Figs. 4 and 5), the system comprising:

- a pre-analytical unit ("task sequence controller TSC 136") being configured to receive a sample and sort/aliquot the sample on request according to respective test requests included within a respective sample order comprising sample test requests (see e.g. page 12, lines 22-24: "... TSCs 136 are capable of dynamic retasking, which, for example allows adding and subtracting assays ..."; page 20, lines 6-13: "... automated test instrument suite commands ... are provided 314 to the ... task sequence controller 136 ..." in conjunction with Fig. 11, step 314);
- analytical units ("automated instruments 106"; "infectron 135A"; "detectron 135B"; "SLM 134"), each being configured to run tests on the appropriately sorted sample (see e.g. page 10, lines 31-33; Figs. 7 and 9);
- a decision unit ("process controller 128" including "process control tools PCT 124") being configured to
- enable a host ("remote client 100") to access the system and to submit the sample order ("test command message") for the sample (see e.g. page 10, lines 19-24 in conjunction with Fig. 11, step 308 and Fig. 12, step 330);
- act as coordinator in communication between the pre-analytical unit and the analytical units (see e.g. page 12, lines 26-33);
- download the sample order comprising the sample test requests from the host (see page 10, lines 26-27 in conjunction with Fig. 11, step 308);
- distribute the sample to an appropriate analytical unit (see in particular page 10, lines 24-31; Fig. 4, "package 104" and "storage 104");
- update the sample order by adding new tests or replacing/modifying/extending requested tests (see e.g. page 13, lines 18-25: "An operation PCT 124B ... offers selections of standardized tests. This PCT 124B also allows researchers to design new experiments, and offers the test designer specified degrees of freedom ..."; page 19, lines 5-8: "... process controller 128 also provides high-level tools to remote clients 100 that allow programming of SLM controllers on the fly, enabling one instrument to perform any number of unique experiments ...");

- collate gained test results with the sample and give a respective report ("output 112") to the host (see e.g. page 10, lines 31-33 in conjunction with Fig. 11, step 316).

The examining division found that the "task sequence controller 136" of D2 corresponded to the decision unit claimed (cf. appealed decision, page 4, penultimate paragraph). The board agrees however with the appellant that the task sequence controller of D2 fails to provide the functionalities of a decision unit as claimed. Rather, the board takes the view that the task sequence controller 136 of D2 corresponds to the pre-analytical unit of claim 1.

Hence, the difference between the subject-matter of claim 1 and the disclosure of D2 is considered to be that (emphasis added by the board)

- a) the sample order comprises a sample ID;
- b) the pre-analytical unit is configured to scan the sample once upon reception and again when the sample is transported back to the pre-analytical unit after the sample order has been updated by the decision unit;
- c) the pre-analytical unit uploads pre-analytical sorting/aliquoting information while the analytical units upload the respective test results to the decision unit;
- d) distribution of the sample to the appropriate analytical unit is performed by the decision unit according to distribution criteria based on pre-analytical information from the pre-analytical unit and on test results from tests of the number of tests which have already been performed by the analytical unit;
- e) the decision unit acts as an intermediary in communication between the pre-analytical unit and the analytical units and coordinates processing of the number of tests until all tests have been done;
- f) updating the sample order is performed by the decision unit with respect to the uploaded test results and the updated sample order is to be processed again until all test requests have been done;
- g) the distribution criteria are configured by the decision unit itself;
- h) the decision unit is configured to combine current analytical data with other sample-related information to decide a next pre-analytical step.

Consequently, the subject-matter of present claim 1 is found to be novel over D2

Regarding the assessment of **inventive step**, the board first notes that, in accordance with the conclusion drawn in the impugned decision (cf. page 5, penultimate paragraph, first and second sentences), the **recursive workflow as defined in point 2.2.12.2.1 represents a workflow which may typically be devised and specified by a medical expert who has arguably no technical knowledge and skills. In view of the generality of its definition, the board also considers that the respective system units mentioned in the workflow are not restricted only to technical devices but may well be represented by human beings (for**

example, the "host" could be a physician, the "decision unit" a laboratory administrator, and the "pre-analytic unit" and "analytic unit" a first and a second laboratory assistant respectively). Thus, the (allegedly improved) workflow itself cannot contribute to an inventive step. The question which therefore arises next is whether the technical implementation of this recursive workflow according to claim 1 may justify an inventive step or whether it constitutes merely an obvious automation of the underlying medical workflow on a common distributed computer system using standard data processing techniques. To answer that question, it has to be ascertained whether distinguishing features a)a) to h)h) give rise to a non-obvious synergistic technical effect. In this context, it is first apparent that the implementation of the medical workflow in question according to claim 1 relates to a mixture of automated electronic processing (partly of cognitive data such as "pre-analytical information"; see features c)c) and d)d) of claim 1) and physical/human-based distribution (transportation) of samples (see features b)b) and d)d) of claim 1).

Although document D2 does not disclose explicitly that the identification codes are scanned by the task sequence controller, the board takes the view that using a scannable sample ID constitutes one of several equally likely implementation alternatives for electronic identification purposes from which the skilled person would choose, depending on practical constraints such as implementation complexity or technological preferences (see e.g. D1, Fig. 1, according to which a bar code scanner/reader for sample identification is used). Thus, the board holds that features a)a) and b)b) constitute obvious measures for implementing steps 1)1) and 3)3) of the recursive workflow.

Features relating to the implementation of steps 5)5) , 6)6) , 8)8) , and 10b) of the recursive workflow have allegedly the technical effect that the decision unit is capable of forwarding a pre-analysed sample to the appropriate analytical unit for being subsequently tested according to the respective sample order (cf. page 16, lines 13-17 in conjunction with page 23, lines 11-14 and 27-36 of the application as filed). From D2 the skilled person in the field of computer-based medical systems would readily deduce that, in order to deliver the sample to be tested to the appropriate SLM, the task sequence controller, which is obviously the only unit that knows which SLM is supposed to perform the requested tests (see D2, page 11, lines 18-19), has to provide the target location of the sample in question (i.e. the respective "storage 104") to the process controller. The board therefore concludes that the implementation measures according to features c)c) and d)d) are a direct and straightforward consequence of steps 5)5) , 6)6) , 8)8) , and 10b) of the recursive workflow.

As to distinguishing features e)e) to h)h) , the board finds, however, that they cannot be considered as a direct and straightforward technical consequence of any step of the recursive workflow, for the following reasons. Features e)e) to h)h) imply that the corresponding test requests are iteratively processed without any intervention by the host computer. The board therefore accepts that those features provide the overall technical effect that the respective sample tests according to a sample order are performed substantially in real-time and in a consistent manner (see also page 18, lines 14-17 of the application as filed), with the bonus effect that no bandwidth is wasted with regard to the connection used between the host and the decision unit for sending an initial sample order and receiving a test report by the host. In other words, after the host has triggered the execution of sample tests by sending a sample order, the remaining units of the system under consideration, i.e. the decision unit, the pre-analytical unit, and the analytical units, automatically and autonomously

perform all the test requests required by the respective sample order. Accordingly, the **objective problem** associated with features e) to h) may be formulated as "**how to ensure that the system of D2 operates substantially in real-time and saves network bandwidth at the same time**".

However, the skilled person in the field of computer-based medical systems, starting out from D2, would immediately recognise that it is the remote user (i.e. "remote client 100") of the underlying system who is exclusively allowed to specify and update the respective test procedures, and in particular to define new tests (see e.g. D1, page 10, lines 17-19 and 26-27 and page 19, lines 5-8). Thus, the remote client is apparently the pivotal point of the entire medical testing system of D2. This is also embodied by the flow diagram of Figs. 11 and 12 of D2, according to which the remote client 100 is supposed to send, for each test request, a separate "access request message" (replied to by an "access enabling message") and an individual "test command message" (responded to by "testing data results") to the process controller 128 via the "communication link 126" such as the Internet (see e.g. page 20, lines 16-26 in conjunction with Fig. 11, steps 300 and 308; Fig. 12, steps 322 and 330). The skilled reader would readily understand from this that it is an **indispensable cornerstone of the system of D2 that the remote user triggers each and every test request**. Confronted with the above objective problem, the skilled person would notice that, in the system of D2, at least the request/response scheme (i.e. the exchange of access request/enabling messages) related to the access of the remote client to the testing system via the process controller 128 causes some unnecessary overhead with regard to the overall transmission speed and bandwidth efficiency as to communication link 126 (see Fig. 5). As a consequence, **the person skilled in the art would attempt to avoid any such overhead either by dispensing with any prior access procedure or by applying such an access scheme only in the initialisation phase of the overall test procedure. However, this kind of implementation would plainly lead away from the solution according to present claim 1.**

Moreover, the board finds that the other cited document on file, i.e. document D1, would not render the subject-matter of claim 1 obvious, regardless of whether taken alone or in combination with the disclosure of D2. This is due to the fact that D1, though also addressing the issue of testing medical samples via bar-code identifiers and of performing user-defined tests, is completely silent as to iteratively implementing a comprehensive medical workflow in both a substantially real-time and bandwidth-efficient manner.

Therefore, even if the teachings of D2 and D1 were combined, the skilled person would not arrive at the claimed solution.

T 0835/10 (Array transformation/MENTOR GRAPHICS) of 3.9.2014 ARRAY TRANSFORMATION IN A BEHAVIORAL SYNTHESIS TOOL

Claims - clarity (no)

Inventive step - (no)

Application number: 03721348.5

IPC class: G06G 7/62, G06F 17/50
Applicant name: Mentor Graphics Corporation
Board: 3.5.06
Catchwords: **De-automation is not per se inventive (see 5.1.8, 5.1.9 and 5.1.6).**
Cited decisions: T 0634/01, T 1937/09, T 0437/98, T 1123/04

<http://www.epo.org/law-practice/case-law-appeals/pdf/t100835eu1.pdf>

The application relates to a method of using a behavioural synthesis tool to design integrated circuits. The tool reads a (behavioural) source code description (in a programming language like C or Pascal, or in a behavioural hardware description language like VHDL or Verilog; see original description page 5, lines 12-17) of a circuit designed by a user (figure 2 (30)).

This description contains arrays, i.e. memories modeled in the source code language (page 1, lines 26-33), and a default set of memory allocation constraints (page 6, lines 2-3; step of "storing" in claim 1 of all requests). A constraint can designate the array length and width (page 17, lines 15-20), the memory type (on-/off-chip, single-/multi-port, synchronous/asynchronous; see original claim 21), the packing mode or the format (page 7, lines 31-33). Note that in original claims 31-33, "format" designates little/big endian or interlacing, whereas on page 9, lines 16-18 and page 10, lines 3-4, "format" designates the size, i.e. array length and word width.

The tool generates from the read source code description an "intermediate data structure" in a kind of internal format, called "synthesis intermediate format" ("SIF", page 5, 17-20; page 6, lines 5-13; page 3, lines 9-12; figure 2 (32)), and stores this data structure.

Now, the tool calculates and displays a report of area and speed of the current array allocation in the memories of the circuit (figure 2 (36)).

The user can then modify the constraints, whereupon the tool re-allocates the arrays (figure 2 (34)). There are two way of modifying: either directly by editing the constraints in the source code description (figure 16(c); claim 4 of the main request; original claim 6; page 3, lines 2-4; sentence bridging pages 7 and 8) or indirectly by using a graphical user interface (GUI) like in figure 13 (see also page 10, lines 1-5).

After the constraints have been modified, the tool displays an updated report of area and speed and waits for some unspecified kind of user satisfaction input (page 8, lines 23-24; page 10, lines 4-5). After the user has expressed his satisfaction, the tool transforms/updates the intermediate data structure (in the SIF format) to represent the modified constraints (the step of "transforming/updating" in claim 1 of all requests; see also page 8, line 24 to page 9, line 9).

All this is done before the register-transfer level description (RTL code; figure 2 (40)) of the circuit is generated.

According to original description page 2, lines 10-19, it is the advantage of the invention to avoid the prior art method of re-editing the source code and re-reading it into the tool in case

the user is not satisfied with the array allocation. Another passage (page 9, lines 16-30) further specifies that when re-editing the source code description, it is the re-calculation of the array addresses for every code segment which accesses an array element (figure 16(b): A[2], A[9]) which "requires substantial editing time" and is "error-prone".

Claim 1 of the main request reads as follows:

"1. A method of reorganizing an array layout in a behavioral synthesis tool used to design an integrated circuit, comprising:

reading a source code description associated with the integrated circuit into the behavioral synthesis tool, the source code description having multiple arrays;

storing the source code description in an intermediate data structure within the behavioral synthesis tool, the intermediate data structure including one or more memory allocation constraints associated with one of the arrays in the source code description, the one or more memory allocation constraints indicating that the one of the arrays is allocated to one or more memories of the integrated circuit according to a first layout format;

displaying to a user a report of circuit area and speed with the one of the arrays allocated according to the first layout format;

allowing a user to modify at least one of the memory allocation constraints via a graphic user interface, wherein the graphic user interface displays the name of the one of the arrays and allows the user to modify any one or more of the memory allocation constraints associated with the one of the arrays; and

responsive to the user modifications, transforming the array layout of the one of the arrays from the first layout format to a second layout format,

wherein the act of transforming comprises updating the intermediate data structure to include the modified memory allocation constraints, and

wherein the acts of displaying, allowing, and transforming are performed while the source code description is stored within the behavioral synthesis tool and prior to the source code description being synthesized into a register-transfer level description of the integrated circuit."

Clarity of claim 1

The expression "source code description" in claim 1 of all requests is unclear. According to the description (page 5, lines 12-17), such a description may be written in hardware description languages like Verilog or VHDL. It is generally known that the latter also allow to formulate circuit descriptions at the RTL level. Thus a source code description could be at the RTL level. However, it follows from description page 5, lines 10-14 that a behavioural source code description is meant, and not an RTL source code description.

Inventiveness

The board considers D1 to be the closest prior art, as it was in the appealed decision (Reasons, 1.1). This was accepted in the grounds of appeal "for the sake of argument" (A.2, first paragraph).

According to the appealed decision (page 4), claim 1 of the refused main request (corresponding to the current main request) differs from D1 in that a graphical user interface (GUI) is used for modifying the constraints, and a report of the estimated circuit area/speed is displayed. The **objective technical problem is how to modify D1 in order to allow a user to manually influence the design space exploration process.** To solve this problem, a skilled person would provide input means for manually overriding the parameters manipulated by the succeeding optimisation process, e.g. array width, length and grouping (page 5, paragraph 3). He would also provide displaying the results of the manipulation, because interactive design and manual correction is a standard procedure in computer-aided design. GUIs are an obvious implementation choice.

According to the grounds of appeal (page 3, first paragraph), the technical problem formulated in the decision points to the solution, since the problem of allowing a user to manually influence the design space exploration is achieved in the claim by "allowing a user to modify ...". The technical problem without an "ex post facto analysis" as used by the examining division is "the need to provide a method for re-organising an array layout to achieve more efficient memory designs more quickly using customisable design options" (paragraph 4). The automated approach of D1 leads the skilled person in a completely different direction than the application: It teaches that optimisation must involve a thorough evaluation of each design possibility relative to the system cost (last paragraph). Based on D1, a fully automated design optimisation is preferable and desirable (grounds, page 4, paragraph 4).

The board agrees with the view of the examining division. It cannot see that the problem formulated in the decision would point to the solution: The feature of the claim which allows the user to manually influence the design is the provision of a GUI and not the "**allowing a user to modify ... constraints**". The latter **cannot be considered as a technical feature in the proper sense. It is more an aim that is achieved by the GUI and by storing the entered values.**

The technical problem formulated in the grounds is less appropriate than that of the decision: "Achieving more efficient memory designs" (page 3, paragraph 4; emphasis added) is stated to be a technical effect of the claimed method and contained in the problem formulation.

However, **a more efficient design is not a result of the computer-implemented method itself, but (if achieved) of the design decisions taken by a human designer which are merely entered via a GUI and then cognitively evaluated by the designer with the help of a report.** Moreover compared to D1, it is unlikely that the claimed method finally achieves a more efficient memory design, since D1 apparently evaluates exhaustively all the possibilities in an automatic way. So, one can only say that the method aids the designer to manually find a good memory design (therefore the expression "computer-aided design"), but **such a good memory design cannot be seen as a technical effect of the method itself.**

The re-organisation of the array layout is done automatically in D1 (see page 106, left column, third sentence: "Other basic moves are the widening and the narrowing of the array." in combination with page 105, left column, last complete sentence: "All three procedures are based on ... and the same set of design state manipulations, called the move set."), in contrast to the application where the user has to enter the values of the array layout.

However, **the board cannot see why a manual computation and entering of optimisation values would be quicker than an automatic computation of these values.** The human designer also would have to spend considerable time to compute appropriate values by hand.

The board is of the opinion that the method of **D1** is characterised by **maximally automating the memory allocation design process, whereas the application aims to provide the designer with a half-automated tool for the same purpose, leaving to the designer the determination of important parameters for allocating arrays in memories.**

However, **as little as merely automating human behaviour is usually considered to be inventive (see for example section 6. from decision T634/01 of a different board, or section 5.4 from T1937/09 of this board in a different composition), the inverse - i.e. "de-automating" or undoing (in a computer-implemented method) the automation performed by a prior art software - cannot in general be considered to be inventive.**

In particular in the present application, the board cannot see an inventive activity in leaving the optimisation task mainly to the designer and providing him with the necessary aid to perform that task (reporting the evaluation parameters for the current design and providing him with a GUI for modifying the design). Therefore, claim 1 of this request is not inventive in the sense of Article 56 EPC.

T 1259/08 (Diversion agent/PENDRAGON WIRELESS) of 6.5.2014 **DIVERSION AGENT USES CINEMATOGRAPHIC TECHNIQUES TO MASK LATENCY**

Novelty - diverting a user's attention from a delay (no Novelty - subjective feature)

Application number: 98957085.8
IPC class: G06F 17/00
Applicant name: Pendragon Wireless LLC
Opponent name: -
Board: 3.5.01

<http://www.epo.org/law-practice/case-law-appeals/pdf/t081259eu1.pdf>

The invention

The invention concerns a networked, user-interactive software application, such as a virtual environment or a multiplayer video game (page 1, lines 2 to 4 of the published application).

Such software applications are generally sensitive to delay caused by network latency. In a networked video game, where multiple users are interacting in real time, the network delay may spoil the interactivity of the game (page 1, lines 8 to 12).

The software agent of the invention seeks to mask the delay by creating a distracting effect that diverts the user from the parts of the application affected by the delay (page 2, lines 3 to 6, 13 and 14). The effect is created using a "cinematographic technique" to manipulate an image displayed to the user, sounds supplied to the user, or tactile feedback to the user (page 2, lines 10 to 14). The cinematographic technique may be, for example: zooming in or zooming out; a dummy object blocking the view; an "interlude"; or switching to another scene (page 2, lines 17 to 24).

Figure 3 shows an example of a networked video game according to the invention, involving two avatars engaged in hand-to-hand combat. Each avatar is controlled by respective users at networked machines (Figure 1: 102 and 104). When either machine detects an unacceptable network delay, it switches to a close-up (zoomed in) representation of the avatars showing their facial expressions but not the blows that are affected by the delay (page 6, lines 21 to 31). This results in the claimed effect of "diverting the user from parts of the application affected by the delay".

Claim 1 reads as follows:

"A software agent (114, 116, 118) for local use at a user in a user-interactive software application (112) for running in a distributed system (100) with multiple data processing machines (102, 104, 106) connected via a data network (110), the software agent (114, 116, 118) comprising

an input (203) for receiving information about a delay in the network (110), and

an effector (224) for locally at the user masking the delay by selectively creating an effect dependent on the delay,

the effector being arranged for creating the effect and diverting the user from parts of the application (112) affected by the delay using a cinematographic technique to manipulate an image displayed to the user, sounds supplied to the user or tactile feedback to the user."

The prior art

D6 discloses a user-interactive software application for use in a distributed system with multiple data processing machines connected via a data network (section 1. Introduction), e.g. a networked, shared virtual environment ("VE"), or a multi-user, Internet-based action game ("Quake"). The 3D graphics of the virtual world ("the world") are rendered at the local client (section 1, right column, lines 7 to 11). This corresponds to the "software agent" in claim 1. D6 also discloses "ameliorating" the effects of network delay (section 3.2.2). To this end, the client software uses knowledge of network delay to manipulate the 3D image by modifying the rendering of objects (section 3.2.2, second paragraph). The image manipulation is called "temporal warping" (ibid). For instance, if a user A throws a ball to user B, temporal warping

may be used to modify the velocity of the ball to compensate for network delay. It was not disputed that this constituted masking the delay by creating an effect dependent on it and that the effect used a cinematographic technique to manipulate an image, as claimed.

Novelty - claim 1

The appellant argued that in the invention the delay and the adverse effects were accepted as they were and the cinematographic technique was used to divert the user from these adverse effects. In contrast, the aim of the temporal warping in D6 was to compensate for the delay such that the end of the ball's trajectory was reached without delay. Thus, the temporal warping in D6 was not "diverting the user from parts of the application affected by the delay".

However, in the Board's view, **whether or not a cinematographic technique has the effect of diverting the user from parts of the application affected by the delay is a matter of human perception.** The **effect will depend on the user's visual system and state of mind,** and is, therefore, also **subjective.** For example, the warping of the ball in D6 might serve to distract some users from other parts of the application, such as the throwing and catching at each end of the event. Other users might not be so distracted and might still notice the delay in all parts of the application. In the Board's view, **such a difference cannot have a limiting effect on the claim.**

Accordingly, claim 1 is **not novel** (Article 54(1) and (2) EPC).

Inventive step - claim 1

Even if the feature of diverting the user from parts of the application affected by the delay had been considered to distinguish the invention, the Board judges that it would **not have involved an inventive step.**

It is established jurisprudence of the boards of appeal that an allowable invention must be a technical solution to a technical problem. The appellant stated that the technical problem was how to deal with network delay. The technical solution was to divert the user with the effect. However, since as discussed above, this **solution is a matter of human perception, it follows that it would be non-technical.** Furthermore, it also follows that **it would be unpredictable whether such a subjective feature would actually solve the technical problem.** In this respect, the invention is somewhat analogous to showing a video clip to somebody waiting for a lift to arrive, which is also using a cinematographic technique to deal with a delay. Thus, there would be **no technical solution to the problem.**

T 0318/10 (Load Distribution/CLUB IT) of 2.9.2014
SERVER/CLIENT SYSTEM, LOAD DISTRIBUTION DEVICE, LOAD DISTRIBUTION METHOD, AND LOAD DISTRIBUTION PROGRAM

Inventive step - (yes)

"Non-technical mathematical formulation" - (no)

Application number: 03780885.4
IPC class: G06F 9/46, G06F 13/00
Applicant name: Club IT Corporation
Cited decisions: T 0208/84

Board: 3.5.06

<http://www.epo.org/law-practice/case-law-appeals/pdf/t100318eu1.pdf>

The context of the invention

The application relates to a method of load distribution to assign a server from amongst a plurality of servers (figure 1, 101a to 101n) connected through a network (figure 1, 100) to a plurality of clients (figure 1, 102, 103) for the execution of a processing request from one of said clients (page 8, line 31, to page 9, line 5). The load distribution function can be installed on any server in the network (page 8, lines 19 to 30).

The load distribution method is illustrated in figures 5, 14 and 15 and explained on pages 14 to 15 and 29 to 34, where server 101a acts as the load distributor to processing servers 101b to 101n. Accordingly the load distribution device receives a request for the execution of a process from the client through the network (page 13, lines 2 to 4; page 14, lines 14 to 17; 501 in figure 5; page 30, lines 13 to 17; S1405 in figure 14; page 32, lines 29 to 33; (5) and 1502 in figure 15). A determining unit (402 in figure 4) within the load distribution device (page 12, lines 25 to 28) uses the results, calculated by its distance calculating unit (404 in figure 4; page 12, lines 29 to 30) based on the estimated resource consumption of the process, to determine the server to which the execution of the incoming process is to be assigned (page 13, lines 4 to 15; page 14, lines 17 to 25; 502 in figure 5; page 30, lines 13 to 27; S1407 in figure 14; page 33, lines 1 to 22).

Subsequently the load distribution device requests the selected server to process the request (page 14, lines 25 to 29; page 31, lines 10 to 18; S1408 in figure 14; page 33, lines 23 to 31; (8) and 1501a in figure 15). Figures 12 and 13 illustrate an example calculation based on two resources, namely CPU and memory, carried out by the distance calculating unit 404; see page 22, line 25, to page 24, line 31. Resources other than CPU and memory can also be used; see page 29, lines 8 to 10. In figures 12 and 13 the x-axis represents the CPU consumption and the y-axis the memory consumption (page 22, line 31, to page 23, line 4). The amount of resources used by processes, already assigned to and executing on servers 101b (figure 12, 1202) and 101c (figure 12, 1203), are depicted as steps of rectangles (figure 12, 1204, 1205, 1210 and 1211; page 23, lines 9 to 11 and 16 to 17). A straight line connecting the origin of the graph for each server with the intersection of the maximum possible values of CPU and memory consumption on that server yields the "resource consumption optimal line" (figure 12, 1206 and 1212) along which processes are distributed without any waste of either resource. For the selection of the optimal server for the process request 1201, its estimated resource consumption is added to processes already executing on each server (figure 12, dashed rectangles), thus shifting the total resource consumption for servers 101b and 101c to points 1209 and 1215, respectively.

The process is assigned to the server which has the shortest length for the normal (figure 12, 1208, 1214) drawn from the shifted coordinates 1209 and 1215 to the resource consumption optimal line 1206 and 1212 which indicates a more balanced resource consumption; see page 24, lines 19 to 31.

Claim 1 of the main request reads as follows:

"A server/client system in which a plurality of servers (101a, 101b to 101n) and a plurality of clients (102) are connected through a network (100), and the servers (101b to 101n) execute a process based on a process request from the clients (102) and transmit a process result to the clients (102), wherein at least one of the servers (101a) includes a process information receiving unit (401) configured to receive information on the process from the clients (102) through the network (100); a determining unit (402) configured to determine a server (101b to 101n) to execute the process from among the servers (101b to 101n) based on the information on the process; and a server information transmitting unit (403) configured to transmit information on determined server (101b to 101n) to the clients (102), and each of the clients (102) includes a server information receiving unit (412) configured to receive the information on the server (101b to 101n); and a process request transmitting unit (413) configured to transmit the process request to the determined servers (101b to 101n); characterised in that the determining unit (402) includes a first calculating unit (404) configured to calculate, for each of the servers, a first distance from an estimation point indicating an estimated consumption to an ideal consumption line, the estimated consumption obtained by adding an amount of resource to be consumed by execution of the process to a point indicating an amount of resource that has been consumed by each of the servers (101b to 101n), the ideal consumption line being a straight line that connects an origin and a point indicating a maximum resource capacity of each of the servers (101b to 101n) expressed in a space having parameters of resources as axes; and the determining unit (402) is configured to determine the server (101b to 101n) with the shortest first distance."

Inventive step, Article 56 EPC 1973

D1 was regarded as the closest prior art. The appellant has not disputed this. D1 discloses a broker mechanism for allocating servers to clients to deliver services (see D1, figure 2: B is the broker; Si are the servers each providing a set of services Ai; Ci are the clients) based on a "network policy" and resource capacities of servers; see D1, abstract.

The following features of present claim 1 were identified in the appealed decision as differences over the disclosure of D1:

the determining unit includes

a first calculating unit configured to calculate, for each of the servers, a first distance from an estimation point indicating an estimated consumption to an ideal consumption line, the estimated consumption obtained by adding an amount of resource to be consumed by execution of the process to a point indicating an amount of resource that has been consumed by each of the servers, the ideal consumption line being a straight line that connects an origin and a point indicating a maximum resource capacity of each of the servers expressed in a space having parameters of resources as axes; and

the determining unit is configured to determine the server with the shortest first distance.

the examining division considered these differences to have the technical effect of allocating servers more efficiently. The problem to be solved by the invention was identified as how to allocate servers in a more efficient manner.

D1 mentions a "network policy" for allocating servers to clients, but does not disclose any specific allocation policy. According to the appealed decision, the skilled person knows that optimisation algorithms are commonly used for resource allocation and load balancing, thus the skilled person, starting from D1, would routinely look for a policy to efficiently allocate servers and would choose any optimisation algorithm providing this effect. As D2 discloses the idea of balanced resource utilisation, the skilled person would find document D2 and adapt its teaching, in particular the so-called "Backfill Balanced algorithm" (see below), to the context of D1.

D2 discloses improved heuristics for selecting backfilling jobs in First-Come-First-Serve (FCFS) scheduling algorithms. Backfilling allows, in the event that the job at the head of a FCFS job queue cannot be scheduled immediately, smaller jobs waiting further down in the queue to be scheduled immediately (see §2.1, second paragraph). A "greedy" approach, termed First-Fit (FF) backfilling, might create imbalances in resource usage (see §2.1, first paragraph and §2.2, last paragraph), so D2 proposes (see §3) two different heuristics, Backfill Lowest (BL) and Backfill Balanced (BB). BL looks only at the single lowest used resource (see §3, first and second paragraphs), whereas BB uses a balance measure to score each backfill job candidate and then selects the job which achieves the best resource utilization (see §3, third paragraph).

The board is not convinced that the skilled person would consult a document on optimal backfilling in FCFS scheduling when looking for a "network policy" for allocating servers to clients in the context of D1. Even if, for the sake of argument, the skilled person were to do so, in the board's view the combination of these two documents would more likely lead the skilled person to a load distribution policy based on FCFS-scheduling which could additionally implement backfilling using either the BL or the BB algorithm disclosed in D2, rather than that set out in claim 1.

However, assuming that the skilled person were to combine D1 and D2 in the manner argued in the decision, the board agrees with the appealed decision that the subject-matter of claim 1 would still differ from the combined disclosure of D1 and D2. **In particular, the calculation of a distance between the estimated consumption point and the ideal consumption line, as defined in claim 1,** is neither disclosed in D1 nor in D2.

The decision states that these "remaining differences refer only to non-technical mathematical formulation of the optimisation function". The appellant argued in the statement of grounds of appeal that this was a "clear misinterpretation of Art. 52(2)(c) in combination with Art. 56 EPC". As it is Article 52(2)(a) EPC which refers to mathematical methods, and not Article 52(2)(c), the board understands the appellant's criticism as referring to Article 52(2)(a) EPC. **The board does not agree with the appealed decision that the mathematical formulation of the optimisation function used in a load distribution method in a computer network is non-technical, or, put another way, without technical effects.** As stated in T 208/84 (Reasons, point 5) ("Computer-related invention/VICOM"; see OJ EPO 1987, 14), a basic

difference between a mathematical method and a technical process can be seen in the fact that a mathematical method is carried out on numbers and provides a result also in numerical form, the mathematical method being only an abstract concept prescribing how to operate on the numbers and producing no direct technical result as such. In contrast thereto, if a mathematical method (in the present case the optimisation function) is used in a technical process (in the present case assigning a server from among a plurality of servers to process a client request), that process is carried out on a physical entity by some technical means implementing the method and provides as its result a certain change in that entity. In the present case **the optimisation function as defined by claim 1 results in a change in the resource consumption of the server system which is a technical effect.**

Furthermore neither D2, nor common general knowledge, would lead the skilled person to supplement the teachings of D1 with the distance calculation and process request allocation techniques according to the present invention. Consequently the board finds that the subject-matter of claim 1 **involves an inventive step** in the sense of Article 56 EPC 1973 in view of the combination of D1 and D2. The subject-matter of independent claims 3 and 5, which set out a corresponding load distribution method and load distribution program, respectively, is thus also considered to involve an inventive step.

T 0710/09 (Image layout/SEIKO EPSON) of 28.8.2014

Image layout information determination

Claims - clarity after amendment (yes)

Novelty - (yes)

Application number: 04007011.2

IPC class: G06F 17/21

Applicant name: Seiko Epson Corporation

Board: 3.5.07

<http://www.epo.org/law-practice/case-law-appeals/pdf/t090710eu1.pdf>

The invention

2.1 The invention relates to the rendering of documents described in a structured tag language such as HTML, XHTML and XML. The background section of the application discusses a system (known from JP 2002/091726) for rendering such documents. This system, which may be a printing system, comprises:

- analysis means for parsing the document and identifying "objects" such as text, images, borders, buttons;

- layout means for generating layout information which comprises for each object the coordinate position at which it should be rendered;

- rendering means for rendering each object on the basis of the generated layout information.

This system is referred to in the present application as "image rendering system". Its rendering means is referred to as "image rendering means".

The application explains on page 2, lines 6-25, that the image rendering means of an image rendering system may be limited in the kinds of objects (or "image rendering components") it can render.

The object of the invention is therefore to provide an image rendering system capable of rendering documents in a structured tag language comprising objects that its image rendering means cannot render.

The proposed solution is to provide the layout means of the image rendering system with an **"inquiry function"** and a **"simulation function"**. The inquiry function inquires with the image rendering means whether it can render a particular object extracted from a document, which essentially depends on whether the image rendering means supports an "image rendering command" corresponding to the type of the object. For an object that the image rendering means cannot render, the simulation function of the layout means simulates image rendering of the object by using objects that can be rendered by the image rendering means. The description on page 12, lines 5-9, gives as an example the simulation of an image rendering command for a text box object using an image rendering command for a border object and an image rendering command for a text object.

Independent claim 1 reads as follows:

"An image rendering system, comprising

an analysis means (21) for analyzing a document described in a structured tag language, the document representing an image to be rendered, and extracting therefrom objects constituting the image;

an image rendering means (23) responsive to image rendering commands that the image rendering means (23) is able to use, to perform image rendering corresponding to the image rendering commands; and

a layout means (22) for determining coordinate positions of the extracted objects constituting the image, and for outputting to the image rendering means (23) layout information including image rendering commands for rendering the extracted objects in their positions,

characterized in that:

the layout means includes

- an inquiry function adapted to place an inquiry with the image rendering means (23) as to whether a respective extracted object can be rendered or not, and

- a simulation function adapted to simulate image rendering, when the respective extracted object cannot be rendered, by using objects that can be rendered by the image rendering

means (23), and to output to the image rendering means (23) layout information including image rendering commands resulting from the simulation; and

the image rendering means (23) is adapted to receive the inquiry from the layout means (22) and to inform the layout means (22) about whether or not an image rendering command corresponding to the respective extracted object is among a set of predetermined image rendering commands that the image rendering means (23) is able to use."

Clarity

According to point 3.1 in combination with point 2.1 of the decision under appeal, the term "image rendering means" had no clear definition and it was not possible for the skilled person to understand which functionalities it implemented. **The term "image rendering means" corresponded to means arbitrarily defined by the applicant.**

The term "image rendering means" is defined in the preamble of present claim 1 as means that in response to certain "image rendering commands" performs corresponding image rendering. In the Board's view, the preamble of present claim 1 defines a conventional system for rendering documents described in a mark-up language ("structured tag language"). Typically, such a document is first parsed using "analysis means" in order to determine constituent objects. The positions of these objects within the rendered document are then determined using "layout means". Finally, the objects are rendered at these positions using "image rendering means".

The Board therefore considers the term "image rendering means" as used in present claim 1 and corresponding claim 3 **to be clear**.

Novelty

According to the decision under appeal, the subject-matter of claim 1 of the then auxiliary request lacked novelty in view of commonly available web browsers such as Internet Explorer. In point 2.3 of the decision, the Examining Division considered in particular that an inquiry function that inquired with the browser's image rendering means whether a predetermined "object of a form" could be rendered or not was anticipated by implicit browser functionality that checked whether a particular image to be rendered could be rendered or not.

Documents D1 and D4 were cited as evidence of how web browsers dealt with images in web pages that cannot be displayed ("broken images"). Document D5 was cited in support of the argument that an image is a "typical form object". The "broken image" pictograms shown in documents D1 and D4 represented the result of simulated rendering.

In point 2.5 of the decision, the Examining Division further argued, without providing written evidence, that the implicit browser functionality that checked for the availability of resources was not limited to checking for resources "on the web", but also checked for "resources on local file systems, GUI objects, font management systems, etc.". Since no standard definitions existed for the claimed layout means and image rendering means, one could "decide" that the latter included this implicit "resource availability checking means".

In the statement of grounds of appeal, the appellant submitted inter alia that the display of a broken image implied that the browser placed an inquiry with an image source, e.g. a web server in the World Wide Web, that was expected to provide the required image file to the browser. This image source was clearly not an image rendering means in the sense of the application. This criticism of the Examining Division's novelty reasoning was also included in the appellant's submission in response to the summons to oral proceedings before the Examining Division.

The Board agrees with the appellant's submission. According to claim 1, the layout means includes functionality ("inquiry function") for placing an inquiry with the image rendering means and the image rendering means includes functionality for responding to this inquiry. The novelty reasoning presented in the decision under appeal equates the "resource availability checking means" both with the functionality included in the layout means (see point 2.3 of the decision) and with the functionality included in the image rendering means (see point 2.5 of the decision). This appears to be how the Examining Division attempted to circumvent the appellant's criticism, but is **clearly not correct**.

The Board sees **further differences** between the claim and the prior art on which the Examining Division relied. According to present independent claim 1 (as well as claim 1 of the auxiliary request on which the decision was based), the image rendering means responds to a received inquiry by informing the layout means "about whether or not an image rendering command corresponding to the respective extracted object is among a set of predetermined image rendering commands that the image rendering means is able to use". This feature makes clear that the response to an inquiry whether or not a particular object can be rendered is dependent on the type of the object: is the object of a type for which a suitable rendering command is available?

The resource availability checking on which the Examining Division relied is not of this kind. According to document D4, page 1, "broken images" are images that "do not load for one reason or another, usually because the graphic file is not where the IMG tag is pointing". If a particular image object cannot be rendered, that is not because objects of the type "image" cannot be rendered, but because that particular image object is not (or no longer) available for download at the URL specified in the HTML document being rendered. The claimed inquiry functionality is therefore not comparable with the "resource availability checking" discussed in the contested decision.

The Board further agrees with the appellant that the **conventional handling of "broken" (i.e. missing) images does not "simulate" rendering of those images. The "simulation" of claim 1 refers to the emulation, by the layout means, of a rendering command that the image rendering means does not support by invoking one or more (typically more elementary) image rendering commands that the image rendering means does support.**

The Board concludes that the subject-matter of claim 1 and that of corresponding independent claims 3 and 4 is new within the meaning of Article 54(1) and (2) EPC with respect to the prior art considered in the decision under appeal.

T 0042/09 (Logical hierarchical data model/BOEING) of 10.3.2014 **Logical hierarchical data model for sharing product information across product families**

Inventive step - main request (no)

Application number: 03076109.2
IPC class: G06F 17/50
Applicant name: The Boeing Company
Cited decisions: T 0049/99, T 0258/03, T 1227/05, T 1841/08, T 1954/08

Board: 3.5.07

<http://www.epo.org/law-practice/case-law-appeals/pdf/t090042eu1.pdf>

The application seeks to address the problem of modelling a "family" of products. Instead of providing a separate model for each variation of a product within a family of products, a single "product data-model" is provided that captures the whole family. This product data-model models a generic product by means of a number of "logical component-usage" nodes. Each "logical component-usage" node essentially represents a logical component function and is connected, by means of a number of "component-usage" nodes, to respective "components" providing such function. The "component-usage" nodes thus represent configuration options for the "logical component-usage" node. By applying certain "applicability attributes", at each "logical component-usage" node a choice is made from the possible "component-usages" and their corresponding "components". See in particular paragraphs [0034]-[0036] of the description.

This product data-model can be understood more easily with the help of Figure 3A. Product data-model 20 models a family of product configurations. The data-model comprises two logical component-usage nodes 183 and 184, each representing a "logical" component having a pump function. The component usage nodes 149 and 1410 represent possible choices of specific pumps for the "logical" component 183, component usage node 149 representing a large pump P1 and component usage node 1410 representing a small pump P6. Similarly, component usage nodes 1411 and 1412 model a choice between pump P1 and pump P6 for logical component 184. A product configuration is now obtained by labelling each component usage node with an "applicability attribute" designating the choice being made.

Claim 1 of the main request reads as follows:

"A method of designing products using CAD and of manufacturing products using CAM, wherein a product data-model is stored in a memory associated with a computer, said product data-model representing product information about at least one component that is a constituent of at least one parent assembly, comprising:

a parent assembly and two or more children, each comprising at least one component, said parent assembly holding information for associating the two children with the parent assembly;

a plurality of component-usages for holding information relating to usages of the component, said component-usages being operatively connected to a component of the parent assembly;

a plurality of logical component-usages for holding information relating to logical usages of the component, said logical component-usages being operatively connected to said component-usages; and

said parent assembly, said component-usages and said logical component-usages being hierarchically interconnected."

Although claim 1 of the main request is directed to a "method of designing products using CAD and of manufacturing products using CAM", it does not define any steps, let alone steps of designing products using CAD and of manufacturing products using CAM. Instead, claim 1 **defines features of a "product data-model" without explaining its relation to a method of designing or manufacturing**.

In this respect, the Board notes that the invention as disclosed in the application also rather appears to be concerned with the **general use of a particular "product data-model" stored in the memory of a computer in unspecified activities related to CAD/CAM**. For example, paragraph [0005] of the description suggests that the product data-model may serve logistical purposes.

The arguments put forward by the appellant further confirm that the present invention is essentially aimed at solving non-technical administrative problems such as checking product configurations for consistency and reducing the number of copies of similar but unique documents.

The features of claim 1 relating to the "product data-model" **define an abstract information model**. Indeed, these features are worded in abstract terms and make **no reference to any concrete physical representation of the product data-model**. A product data-model having these features could take the form of a drawing on paper such as depicted in Figure 3A.

According to decision T 49/99 of 5 March 2002, reasons 7, **information modelling is in principle a non-technical activity, and only a purposive use of information modelling in the context of a solution to a technical problem may contribute to the technical character of an invention**. For the reasons given under point 2.32.3, the Board considers that the claimed connection with CAD/CAM activities cannot qualify as such a purposive technical use. The **product data-model does not enable, improve, or otherwise contribute to the solution of a concrete technical problem**.

The features defining the "product data-model" hence are non-technical. This means that they cannot contribute to an inventive step. Since the mere additional mention of unspecified CAD/CAM activities and the feature specifying that the product data-model is "stored in a memory associated with a computer" cannot support an inventive step either, the invention as defined by claim 1 lacks an inventive step within the meaning of Articles 52(1) and 56 EPC over a notorious general purpose computer.

The same conclusion is reached when starting from document D1 as closest prior art. According to the contested decision, the distinguishing feature solved the problem of how to

avoid having multiple copies of components. The claimed solution was considered to be obvious in light of the common general knowledge that, when multiple copies of identical components are needed, memory can be conserved by creating only a single copy and employing references to that single copy. An example of this common general knowledge was to be found in document D3, Figure 5, in the form of "NextAssemblyUsages".

The appellant further argued in favour of inventive step on the basis of the introduction by the present invention of "the idea of considering the component usage to be a lightweight copy of a component instead of as a relationship". The appellant submitted document D4 to "further introduce" the concept of a lightweight copy.

The Board understands this argument as being intended to support the non-obviousness of the provision of "component usages" as a link between "logical component usages" and "components", instead of providing multiple full copies of a component definition. It could be argued, as the Examining Division did, that this reduces memory usage.

However, any **such reduction of memory usage manifests itself only at the level of the physical layout of the product data-model as a data structure in the memory of a computer.** The **claim defines the product data-model in more abstract terms** and would appear to cover implementations **requiring essentially arbitrary amounts of memory.**

Furthermore, the Board considers that **a reduction of memory usage merely arising out of a change in an abstract data model does not confer technical character on the data model.** See in particular decisions T 258/03, OJ EPO 2004, 575, headnote II, T 1227/05, OJ EPO 2007, 574, reasons 3.2.5, and T 1954/08 of 6 March 2013, reasons 6.2.

For these reasons the Board considers, consistently with the reasoning in point 2.42.4 above, the distinguishing feature to be a non-technical feature not contributing to an inventive step.

T 2072/09 (Firmware upgrade/AXIS) of 29.7.2014 **METHOD AND APPARATUS FOR UPGRADING FIRMWARE IN AN EMBEDDED SYSTEM**

Inventive step - (yes)

Application number: 00946726.7
IPC class: G06F 9/445
Applicant name: Axis AB

Board: 3.5.06

<http://www.epo.org/law-practice/case-law-appeals/pdf/t092072eu1.pdf>

The independent claim 1 of the main request reads as follows:

A method for upgrading firmware of an embedded system (110) over a network connection, comprising the steps of:

sharing a virtual file system (122) of the embedded system (110) with a client computer (10) via the network connection in such a way that the virtual file system (122) can be utilised by the operating system of the client computer (10) as if it was part of the file system of the client computer (10),

receiving (406) data of a firmware upgrade file in a directory for upgrading (126) at the virtual file system (122) of the embedded system (110) via the network connection, said directory for upgrading (126) being associated with a data channel (128) being a control means,

performing control operations by means of the data channel (128) on the data of the firmware upgrade file received in the directory for upgrading (126),

storing (406) the data of the firmware upgrade file in a volatile memory (150) of the embedded system (110) connected to the virtual file system (122), and writing (424) the firmware upgrade file from the volatile memory (150) to a persistent memory (160) of the embedded system (110).

The independent claim 14 of the main request reads as follows:

An embedded system (110) comprising:

a volatile memory (150),

a persistent memory (160),

a protocol stack (180) for communication over a network (30),

means for upgrading firmware received over a network (30), and

means for writing the firmware upgrade file from the volatile memory to the persistent memory, said embedded system is characterised by:

a virtual file system (122) being shareable with an operating system of a client computer (10) via the network (30) and being connected to the volatile memory (150),

a directory for upgrading being included in said virtual file system, and

a data channel associated with said directory for upgrading, the data channel being a control means arranged to handle data of a firmware upgrade file sent to the directory for upgrading, and arranged to perform control operations on the data of the firmware upgrade file received.

Inventive step; Article 56 EPC 1973

The board considers that D3 represents the closest prior art and discloses a method for upgrading firmware of an embedded system over a network connection (see D3, page 1, lines 5 to 6), comprising the steps of:

receiving data of a firmware upgrade file ("patch file") via the network connection (figure 1: communication network 12),

storing the data of the firmware upgrade file in a volatile memory of the embedded system (page 14, lines 1 to 2 and page 23, lines 26 to 28), and writing the firmware upgrade file from the volatile memory to a persistent memory of the embedded system (page 14, lines 3 to 5 and page 22, lines 24 to 28).

As acknowledged in the appealed decision, Reasons 11.2, D3 does not disclose a virtual file system. In fact, the board considers that, whilst it is true that the patch information in D3 is stored in a file (which is called "patch file" throughout D3), D3 does not disclose any kind of file system. The single patch file could very well exist independently of a file system. Therefore, even if the skilled person wanted to solve the problem mentioned in Reasons 11.3 of the appealed decision, i.e. to facilitate data transfer, the absence of a file system in the mobile device of D3 would not allow him or her to solve the problem in a straightforward manner by having the client computer use the file system of the mobile device as if it was part of its own file system.

It may be true, as was part of the argument in the appealed decision, Reasons 11.4, that the combined disclosure of documents D3 and D4 contains all the features of claim 1. **However, given that, as pointed out above, D3 does not use or at least does not disclose the use of a file system, there is no apparent incentive for the skilled person to apply the concept of a shared file system disclosed in D4 to the method of D3.** Furthermore, in D4 it is the embedded system (target computer) which uses an image of a disk drive stored on the host computer as if it were a disk drive on the target computer (see D4, figure 1 and corresponding part of the description). In the present claim 1, apart from the fact that no disk image but a virtual file system is used, the situation is reversed, i.e. it is the client computer which uses the virtual file system of the embedded system as if it were part of its own file system.

It is further argued in the appealed decision, Reasons 11.4, penultimate sentence, that document D2 demonstrates that shared file systems were well known since the 1980's as a standard way of transferring data between computers. The board acknowledges that **this may well be the case. However, it would remain necessary to demonstrate that the skilled not only could but would apply such a technique in the context of D3.** No such demonstration was provided in the appealed decision and the board fails to see a reason why the skilled person would proceed in this manner.

The board considers that the existence of the "patch file" in D3 does not automatically imply the existence of a directory. In fact, since the patch file is the only file used in the mobile device of D3, there is no need for a directory. Given that a skilled person would naturally try to economise memory in a mobile device, as well as avoid the introduction of features that have no benefit but would only complicate the device and increase the risk of failure, he or she would have no incentive to implement any kind of directory in the device of D3.

A fortiori, D3 does not disclose or render obvious the use of a "directory for upgrading".

T 0506/10 (Virus Scanning / McAfee) of 3.7.2014
**METHOD AND SYSTEM FOR DETECTING VIRUSES ON
HANDHELD COMPUTERS**

Inventive step - (no)

Application number: 01914786.7
IPC class: G06F 11/00, G06F 1/00
Applicant name: McAfee, Inc.

Board: 3.5.01

<http://www.epo.org/law-practice/case-law-appeals/pdf/t100506eu1.pdf>

The invention is concerned with virus scanning. That was, at the priority date, the most widely used type of antivirus measure (published application, page 4, lines 14 - 15). A virus scanner worked by examining files on a computer, and looking for particular patterns of data that were known to be associated with viruses. One of the measures that was commonly taken, if such a pattern was found, was to "clean" the data (published application, page 5, lines 2 - 4).

The invention is not about how viruses come to be associated with particular patterns of data, or how the patterns can be efficiently recognised. It deals with a different problem. Hand-held computers were becoming prevalent, and were vulnerable to viruses just as any other computer was. Virus scanning did not work very well on them. They had too little memory and were too slow. That is the problem behind the present invention.

The idea of the invention can be simply stated: have another, more powerful computer do what the hand-held computer cannot. Data are copied from the hand-held computer to another computer, and the other computer performs the scan.

In the invention as defined by claim 1 according to the main request, infected data are erased and clean data are written to the hand-held device.

Claim 1 according to the main request reads as follows:

A method for detecting viruses on a handheld computer comprising:

scanning (188) data from a handheld computer for viruses with a virus detection program;

characterised by the following steps taking place before scanning:

putting said handheld computer into communication with a computer system,

reading (182) data from the handheld computer (20);

storing (184) said data at least temporarily on the computer system (22);

and characterised in that said scanning (188) of said data for viruses with the virus detection program takes place on said computer system;

and further characterised by updating data held on the handheld computer, after said scanning, wherein said updating is based on the results of the scanning by writing (198) cleaned data to the handheld computer and deleting (202) infected data.

The appellant suggests that the prior art set out in the application should be the starting point for the assessment of inventive step, but, in its arguments in support of the auxiliary request, accepts that D1 or D4 could be taken (see point XII.XII. , above). The Board's view is that the invention defined by claim 1 would have been obvious to the skilled person whether she started from D1 or from the prior art as set out in the application. Both are reasonable starting points.

D1 discloses a network of workstations and a server. The workstations have limited capacity, and so virus scanning is carried out by the server (D1, page 1, line 27 - page 2, line 1; page 2, lines 10 - 16 and 18 - 22). That involves copying the data to the server. When a virus is found, there is no cleaning; rather, access to the data is blocked (D1, page 4, lines 8 - 11; page 8, line 7). As a result, D1 discloses the whole subject matter of claim 1 except that the workstations of D1 are not hand-held, and that D1 imposes a sort of quarantine on files rather than cleaning them. The appellant has not suggested that there is any unexpected advantage in cleaning data on hand-held computers; nor does the Board see any.

The Board considers that the skilled person would have been motivated to use at least one hand-held device. If such devices were not common in 1995, when D1 was written, they were by the priority date (24 March 2000) of the present application. The skilled person would have been aware of general developments in computers, in particular the fact that many manufacturers were producing hand-held devices. The published application mentions several, on page 6. In such an environment, it would have been natural for her to consider their use.

The Board put the view that cleaning was well known to the appellant, who did not argue against it. Indeed, the application itself states that cleaning was included in "most scanners" (published application, page 5, lines 2 - 4). Such cleaning would, in the Board's judgment, have been an obvious alternative to the quarantine proposed in D1. Thus, the invention would have been obvious for the skilled person starting from D1.

T 2270/10 (Programmieroberfläche/RENNER) of 2.7.2014

Programmieroberfläche zum Programmieren von Computern

Erfinderische Tätigkeit - (nein)

Anmeldenummer: 07012983.8

IPC-Klasse: G06F 9/44

Name des Anmelders: Renner, Peter

Angeführte Entscheidungen: G 0003/08, T 0641/00, T 1171/06, T 1539/09

Kammer: 3.5.06

<http://www.epo.org/law-practice/case-law-appeals/pdf/t102270du1.pdf>

Die Erfindung

Die Anmeldung geht von der Beobachtung aus, dass die konventionelle Programmierung aufgrund ihrer Komplexität nur noch Spezialisten offenstehe und befasst sich mit der Aufgabe, diesen Umstand zu ändern (ursprüngliche Anmeldung, S. 2, 1. Satz; S. 3, 1. Abs.).

Als Lösung schlägt die Anmeldung ein Programmiersystem und das entsprechende Verfahren vor, das den Programmierer "vom Ballast der Sprachkonventionen und dem komplizierten Formalismus ... befreien" soll (S. 3, 1. Satz). Dieses Ziel wird erfindungsgemäß durch eine grafische Benutzeroberfläche erzielt, die Programmkomponenten zur Auswahl aus Listen (Kontextmenüs) bereitstellt und so hilft, das mühsame und fehleranfällige Eintippen von Text zu vermeiden (S. 3, letzter Abs.).

Gleichzeitig sind Programmanweisungen vorgesehen, die grundsätzlich aus Operationen und Objekten bestehen (S. 3, vorletzter Abs.), so dass der Programmierer sich auf den zu steuernden Prozess - das "Prozessuale" - konzentrieren kann (S. 4, 1. Abs.). Auf diese Weise werde das Programmieren intuitiver und der Programmcode selbst leichter zu lesen und zu warten (S. 4, 1. Abs.).

Die Anmeldung stellt fest (S. 5, 2. Abs.), dass "Objekte ... häufig in Baumstrukturen eingebunden [sind], so dass diese mit einem Pfad gekoppelt sein können". Damit laute die typische Grundform einer Anweisung wie folgt: ">Operation: Pfad\Objekt<". Die Teile "Operation" und "Pfad\Objekt" nehmen hierin beschreibungsgemäß die "Funktion von Schlüsselworten" ein, die im "Bedienprozess" bei der Erstellung einer konkreten Anweisung durch "eine spezielle Operation und Objekte ersetzt werden können" (S. 5, letzter Abs.). So erstellte Anweisungen können in dem Abschnitt, in dem eine Operation oder ein Objekt nach Auswahl aus einer Liste eingefügt wurde, eine markierte Stelle aufweisen, mittels derer die relevante Auswahlliste wieder aufgerufen werden kann (S. 28, 2. Abs. und ursprünglicher Anspruch 18). Es wird beschrieben, dass eine solche Markierung vom Softwareentwickler angeklickt werden könne, um so etwa ein gewähltes Objekt durch ein anderes zu ersetzen.

Es sind unterschiedliche Kategorien von Objekten vorgesehen, die als "Hauptobjekte, Objekte und Subobjekte" bezeichnet werden (S. 8, 2.-4. Abs.). Die Beschreibung erläutert, dass "Hauptobjekte" Objekte enthalten und somit zum Beispiel die "Funktion von Containern" annehmen können, während "Objekte" keine weiteren Objekte enthalten. Sie erläutert weiter, dass "Subobjekte" einen "Bezug zu einem Objekt oder zu einer Operation haben" und so "die korrespondierende Operation oder das Objekt ergänzen". "Welche Hauptobjekte, Objekte und Subobjekte einem Softwareentwickler zur Verfügung" stünden, hänge "von der Mächtigkeit der Sprache ab, die dem Softwareentwickler zur Verfügung steht" und "davon, welche Objekte er installiert hat" (S. 9, Zn. 1.-4.).

Anspruch 1 des Hauptantrags lautet wie folgt:

"Computergestütztes Verfahren zur imperativen Programmierung von Computern mit mehreren Programmanweisungen umfassendem Programmcode, wobei Schlüsselworte und Auswahllisten vorgesehen sind, mit einer grafischen Benutzeroberfläche wobei sich Programmanweisungen aus Operationen und Objekten zusammensetzen, wobei Objekte adressierbar sind sowie Operationen und Objekte aus Auswahllisten ausgewählt werden,

dadurch gekennzeichnet, dass

einem individuellen Namen eines Objektes, das durch das Ersetzen eines Schlüsselwortes in eine Anweisung eingefügt wird, ein Pfad zugeordnet ist, welcher zusammen mit dem individuellen Namen in die Anweisung eingefügt wird, wenn dieses Objekt in einer hierarchischen Struktur eingebunden ist."

Technische Aufgabe und technischer Beitrag

Die Anmeldung gibt als **Ziele** der vorliegenden Anmeldung an, die **Programmierung von imperativen Programmen für Nicht-Spezialisten zu vereinfachen, sowie den so erzeugten Programmcode leichter lesbar und wartbar zu machen**. Die Kammer ist der Ansicht, dass **keines dieser Ziele als eine technische Aufgabe gelten** kann, wie sie nach ständiger Rechtsprechung der Beschwerdekammern als Voraussetzung für eine erfinderische Tätigkeit im Sinne des Artikel 56 EPÜ 1973 unerlässlich ist. Die Kammer folgt in dieser Entscheidung ihren früheren Entscheidungen T 1539/09 und T 1171/06.

In T 1539/09 (1. Orientierungssatz) hat die Kammer festgestellt, dass die **Tätigkeit des Programmierens als ein mentaler Vorgang anzusehen ist, soweit sie nicht im Rahmen einer konkreten Anwendung oder Umgebung in kausaler Weise der Erzielung einer technischen Wirkung dient**.

Eine solche Wirkung ist im vorliegenden Fall für die Kammer nicht ersichtlich. Insbesondere ist das beanspruchte Programmiersystem intentionsgemäß universell, nicht auf ein konkretes Anwendungsgebiet beschränkt und schließt ausdrücklich technische wie nicht-technische Anwendungsgebiete ein (vgl. S. 4, Zeilen 1-3). Ob und inwieweit die Erfindung daher die Programmentwicklung erleichtert, scheint der Kammer somit für die Bewertung der erfinderischen Tätigkeit nicht relevant zu sein.

Ob und inwieweit eine Programmiersprache und ein Programmiersystem das Programmieren erleichtert, hängt zum einen von vielen (nicht beschriebenen oder beanspruchten) Umständen ab und stellt zum anderen wenigstens zu einem erheblichen Teil ein **subjektives Kriterium** dar, das sich **nur schwer, falls überhaupt, sinnvoll quantifizieren und verlässlich feststellen lässt**. Die Anmeldung stellt keine belastbare Grundlage dar, auf der die behauptete Vereinfachung der Programmerstellung zu überprüfen wäre.

In der Anmeldung wird insbesondere dargestellt, dass die Erfindung eine neue Art des Programmierens - "[p]rozessuales Programmieren" - unterstütze, gestützt u. a. auf den "Grundgedanken", Programmanweisungen aus Operationen und Objekten" zu bilden und so "[d]ie Denkweise beim Programmieren ... in eine neue Richtung" zu lenken. Die Kammer verweist in dieser Hinsicht darauf, dass **Programmiersprachen bekanntermaßen auf unterschiedlichen sogenannten "Paradigmen" beruhen**, die jeweils eine unterschiedliche Sicht auf die zu entwickelnden Programme ermöglichen oder sogar erzwingen: Beispielsweise stehen

im Zentrum "objekt-orientierter" Programme sogenannte "Objekte", die jeweils die "Methoden" zu ihrer Bearbeitung gleich selbst bereitstellen, und Lösungen in objekt-orientierten Sprachen müssen als Kooperation solcher, miteinander kommunizierender Objekte konzipiert werden; im Zentrum "funktionaler" Programme stehen hingegen mathematische Ausdrücke, und Lösungen in solchen Sprachen müssen als die Auswertung bzw. Vereinfachung solcher Ausdrücke konzipiert werden.

Die Bereitstellung der entsprechenden Programmiersprache und ihrer Ausdrucksmittel wie Objekt/Methode/Nachricht, Funktion/Auswertung oder eben Operation/Objekt wie im vorliegenden Fall **trägt** nach Ansicht der Kammer - und gemäß T 1539/09 (2. Orientierungssatz) - **nicht zur Lösung eines technischen Problems bei**.

In T 1171/06 (Orientierungssatz) stellte die Kammer weiter fest, dass **einem in der Softwareentwicklung verwendeten Modell kein technischer Effekt dadurch zukomme, dass es der Dokumentation oder Kommunikation dient, selbst wenn sein Gegenstand ein technisches System sei**.

In diesem Sinne hält die Kammer auch die behauptete **Erleichterung der Lesbarkeit und Wartbarkeit der erstellten Programme für die Bewertung der erfinderischen Tätigkeit nicht für relevant**.

Auch hier sei aber zudem angemerkt, dass die Frage, ob ein Programm **leicht lesbar und wartbar** ist, eine **größtenteils subjektive** ist, und es schwierig bis unmöglich zu sein scheint, einen solchen Schwierigkeitsgrad sinnvoll und verlässlich zu beziffern.

Die Kammer schließt nicht aus, dass konkrete Details einer graphischen Benutzeroberfläche die Bedienung eines Computers als ein technisches Gerät und/oder bei der Anwendung auf eine technische Aufgabe erleichtern, und auf diese Weise einen Beitrag zur Lösung einer technischen Aufgabe dienen können. Allerdings legt wenigstens Anspruch 1 aller vorliegenden Anträge - wie oben ausgeführt - allenfalls fest, dass die Programmierung mit der vorgeschlagenen Programmiersprache durch eine grafische Benutzeroberfläche unterstützt wird, nicht aber durch welche konkreten Mittel.

Zusammenfassend also interpretiert die Kammer den Gegenstand von Anspruch 1 aller vorliegenden Anträge als ein computergestütztes Verfahren, demgemäß **die Programmierung einer gewünschten imperativen Programmiersprache durch eine im Wesentlichen undefinierte grafische Benutzerschnittstelle unterstützt** wird.

In Übereinstimmung mit T 1539/09 ist die Kammer der Ansicht, dass **die Definition der gewünschten Programmiersprache keine technische Aufgabe löst** und somit keinen Beitrag zur erfinderischen Tätigkeit leisten kann. Im Rahmen des Aufgabe-Lösungs-Ansatzes dürfen die entsprechenden Merkmale somit, gemäß T 641/00 (ABl. EPA, 2003, 352; Orientierungssatz II), bei der Formulierung der Aufgabe als Teil der Rahmenbedingungen für die zu lösende technische Aufgabe aufgegriffen werden, insbesondere als eine zwingend zu erfüllende Vorgabe. Die von der Erfindung gemäß Anspruch 1 aller Anträge **gelöste Aufgabe** kann somit darin gesehen werden, die Entwicklung von Programmen in einer **Programmiersprache mit den geforderten Merkmalen zu unterstützen**.

Programmierumgebungen mit grafischen Benutzeroberflächen sind zu diesem Zwecke allgemein bekannt und daher **ohne Weiteres naheliegend**. Daher kommt die Kammer zu dem Schluss, dass der Gegenstand von Anspruch 1 aller Anträge schon gegenüber dem allgemeinen Fachwissen keine erfinderische Tätigkeit im Sinne von Artikel 56 EPÜ 1973 aufweist.
