BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS | **BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE** | **CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN DES BREVETS**

Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 26 May 2025

Case Number:	T 0537/24 - 3.5.06
Application Number:	20205008.4
Publication Number:	3790200
IPC:	H03M7/30, G06F3/12, B41J2/175, G06F40/146, G06F40/154, G06F40/14, G06F40/123, G06F16/81
Language of the proceedings:	EN

Title of invention:

PRINT MATERIAL CONTAINER COMPRISING MEMORY STORING COMPRESSED XML AND PRINTER EXTRACTING THE XML TO CONTROL AN OPERATION

Applicant:

Hewlett-Packard Development Company, L.P.

Headword:

Print container storing compressed XML file/HEWLETT-PACKARD

Relevant legal provisions:

EPC Art. 52(1), 56, 84, 76(1), 111(1), 123(2)
RPBA 2020 Art. 13(2)

Keyword:

Inventive step - step-by-step analysis with cascaded problems
Inventive step - after amendment (yes)

Decisions cited:

T 1158/02, T 1132/20

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0

Case Number: T 0537/24 - 3.5.06

DECISION
of Technical Board of Appeal 3.5.06
of 26 May 2025

Appellant:
(Applicant)

Hewlett-Packard Development Company, L.P.
10300 Energy Drive
Spring, TX 77389 (US)

Representative:

Hoffmann Eitle
Patent- und Rechtsanwälte PartmbB
Arabellastraße 30
81925 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 13 December 2023 refusing European patent application No. 20205008.4 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman M. Müller
Members: M. Domingo Vecchioni
A. Jimenez

Summary of Facts and Submissions

I. This appeal is directed against the decision of the examining division to refuse European patent application No. 20205008, which is a divisional of European patent application No. 15908976 (the parent application).

II. The examining division held that the independent claims according to the then main request and auxiliary requests 1 to 3 did not involve an inventive step, Articles 52(1) and 56 EPC.

III. The decision cites, *inter alia*, the following documents:

D1: US 2011/0157647 A1,

D2: WO 2015/016863 A1,

D3: EP 1 610 228 A1,

D4: P. M. Tolani et al., "XGRIND: A query friendly XML compressor", Proceedings of the 18th International Conference on Data Engineering (ICDE'02), 2002, pages 225-234, XP010588214.

IV. In the statement of grounds of appeal, the appellant requested that the decision of the examining division be set aside and that a patent be granted on the basis of the main request or, alternatively, on auxiliary request 1 or 2, as considered in the contested decision.

V. Following a summons to oral proceedings, the board issued a communication pursuant to Article 15(1) RPBA setting out its preliminary opinion. In accordance with Article 114(1) EPC, the following documents were introduced:

D7: P. Skibinski et al., "Effective asymmetric XML compression", Software - Practice and Experience, vol. 38, 2008, pages 1027-1047, XP002563159,

D8: D. Salomon et al., Handbook of Data Compression, 5th edition, Springer-Verlag, 2010, pages 329-334, XP093270460.

The board raised several objections under Articles 84, 76(1) and 123(2) EPC, and concurred with the examining division that claim 1 of all requests lacked an inventive step, Articles 52(1) and 56 EPC, in view of a combination of D2 with D3. In addition, the board considered that an inventive step was also lacking in view of a combination of D2 with D7 (and common general knowledge exemplified by D8). A combination of D2 with D4 was also identified as potentially relevant.

VI. By letter dated 21 May 2025, the appellant filed claims for an auxiliary request 3.

VII. During the oral proceedings, the appellant submitted claims for new auxiliary requests 4 to 6, but ultimately maintained only auxiliary request 6.

The appellant's final request was that the decision of the examining division be set aside and that a patent be granted on the basis of the claims filed during the oral proceedings as "auxiliary request 6".

At the end of the oral proceedings, the chair announced the decision of the board.

VIII. The request filed as "auxiliary request 6" comprises only two claims.

Claim 1 reads:

A print container (400), comprising:
a supply of a print material (410);
a print material distributer (420); and
a memory (430) storing:

a tag index (432) containing a list of tag terms from an extensible markup language, XML, file containing instructions that control an operation of a printer and using a known subset of XML syntaxes, the tag index being organized according to a frequency of the tag terms in the XML file, where a tag term of the tag index refers to an instruction defining an action to be performed by the printer into which the print container is to be installed;

an attribute index (434) containing a list of attribute terms from the XML file organized according to a frequency of the attribute terms in the XML file, where an attribute term of the attribute index is used to specify a parameter of the action triggered by the tag term;

a value index (436) containing a list of value terms from the XML file organized according to a frequency of the value terms in the XML file, where a value term of the value index is used in association with the attribute term to specify the parameter of the action triggered by the tag term the attribute term is to modify; and

a condensed version of the XML file (120, 438) containing only pairs of characters and specialized

symbols (TE) indicating that a most recent open tag should be closed, wherein each of the pairs of characters (T2) corresponds to a specific tag, attribute, or value of the XML file, wherein a first portion of each of the respective pairs of characters identifies the tag index, the attribute index, or the value index, and a second portion of each of the respective pairs of characters identifies an index value in the tag index, the attribute index, or the value index,
wherein the tag index, the attribute index, the value index, and the condensed version of the XML file are stored in the memory in a compressed format.

Claim 2 reads:

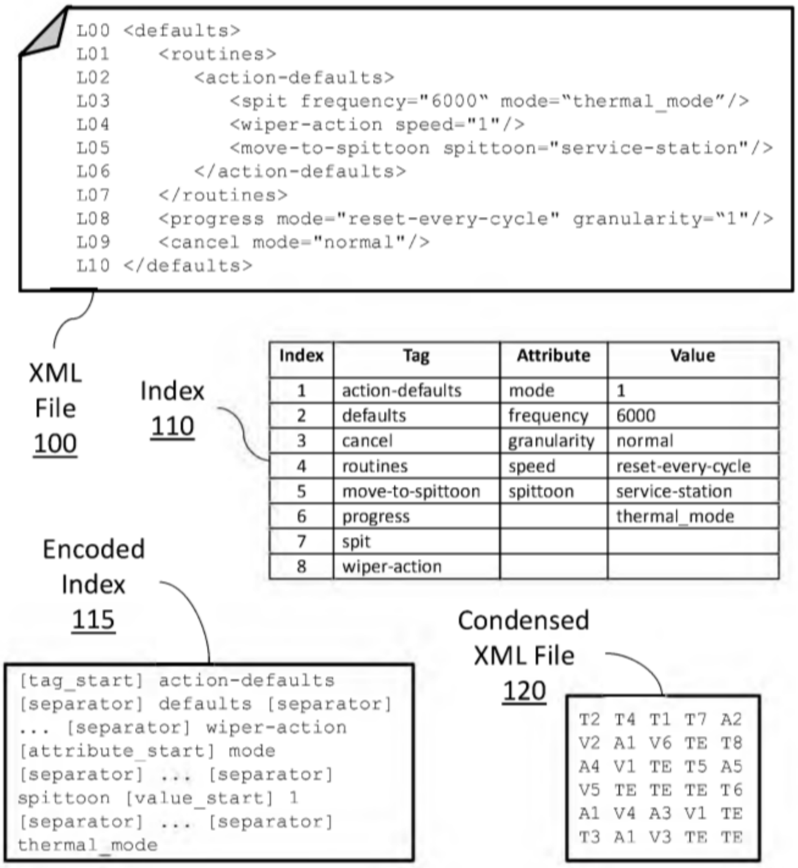
The printer container of claim 1, where the XML file is associated with the print container storing a particular print material and where the XML file is to control the printer to perform a maintenance function suitable for the particular print material.

Reasons for the Decision

The invention

1. Claim 1 is directed to a print container (e.g. an ink cartridge) and specifies how an XML file containing printer operation instructions is stored in a "condensed" and compressed form in the memory of the print container. This is achieved using a "tag index", an "attribute index" and a "value index", all derived from the original XML file.

2. The claimed subject-matter is based on the embodiment illustrated in figure 1:



```
<defaults>
   <routines>
      <action-defaults>
         <spit frequency="6000" mode="thermal_mode"/>
         <wiper-action speed="1"/>
         <move-to-spittoon spittoon="service-station"/>
      </action-defaults>
   </routines>
   <progress mode="reset-every-cycle" granularity="1"/>
   <cancel mode="normal"/>
</defaults>
```

Index	Tag	Attribute	Value
1	action-defaults	mode	1
2	defaults	frequency	6000
3	cancel	granularity	normal
4	routines	speed	reset-every-cycle
5	move-to-spittoon	spittoon	service-station
6	progress		thermal_mode
7	spit		
8	wiper-action		

```
[tag_start] action-defaults
[separator] defaults [separator]
... [separator] wiper-action
[attribute_start] mode
[separator] ... [separator]
spittoon [value_start] 1
[separator] ... [separator]
thermal_mode
```

```
T2 T4 T1 T7 A2
V2 A1 V6 TE T8
A4 V1 TE T5 A5
V5 TE TE TE T6
A1 V4 A3 V1 TE
T3 A1 V3 TE TE
```

The tag index (columns "index" and "tag" in index 110) contains a list of tag terms extracted from the XML file 100, sorted by frequency of occurrence within the file. Similarly, the attribute index (columns "index" and "attribute") and the value index (columns "index" and "value") are generated from the attribute terms and values found in the XML file (paragraph [0017] of the description as filed).

In the "condensed XML file" 120, each tag term, attribute term and attribute value from the original XML file 100 is replaced by a pair of characters. The first character identifies the relevant index (e.g. "T" for tag index, "A" for attribute index, and "V" for value index) and the second character the corresponding index value. For example, "T2" stands for the entry with index value 2 in the tag index, i.e. tag term "defaults" (paragraph [0021]).

The condensed XML file contains *only* these pairs of characters along with a specialized symbol ("TE") whenever the most recent tag is to be closed (corresponding to "/>" in the original XML file 100) (see fig. 1 and paragraphs [0021]-[0022]).

3. The tag index, attribute index, value index (which may be represented collectively as encoded index 115) and the condensed XML file are further reduced in size using a (general-purpose) compression technique. The resulting compressed data is stored in the memory of the print container (paragraphs [0023]-[0024]).

4. Upon installation of the print container into a printer, the data can be decompressed and the original XML file reconstructed from the condensed XML file using the three indexes and a known syntax of the XML file (paragraphs [0026]-[0027]).

Admittance

5. Auxiliary request 6 was filed during the oral proceedings before the board. As it overcame all objections raised by the board - including those introduced for the first time in its preliminary opinion - and did not give rise to any new issues, the board exercised its discretion under Article 13(2) RPBA to admit this request into the proceedings.

Articles 76(1) and 123(2) EPC

6. Article 123(2) EPC

6.1 Claim 1 is based on the claim-like "paragraph 8" on page 15 in the application as filed (which corresponds to independent claim 8 of the parent application as filed), with the following amendments:

- the XML file is *"containing instructions that control an operation of a printer"* (based on paragraph [0031]);

- the XML file is *"using a known subset of XML syntaxes"* (based on paragraphs [0016] and [0026] and the claim-like "paragraph 6" on page 14);

- *"where a tag term of the tag index refers to an instruction defining an action to be performed by the printer into which the print container is to be installed"* (based on paragraphs [0014] and [0025]);

- *"where an attribute term of the attribute index is used to specify a parameter of the action triggered by the tag term"* (based on paragraph [0015]);

- *"where a value term of the value index is used in association with the attribute term to specify the parameter of the action triggered by the tag term the attribute term is to modify"* (based on paragraph [0015]).

Furthermore, instead of specifying the condensed version of the XML file as being "created by translating the XML file according to the tag index, the attribute index, and the value index" (which the board considered to be an unclear product-by-process feature in its preliminary opinion), claim 1 now specifies it as

- *"containing only pairs of characters and specialized symbols (TE) indicating that the most recent open tag should be closed"* (based on fig. 1, condensed XML file 120, and paragraphs [0021]-[0022]),

- *"wherein each of the pairs of characters (T2) corresponds to a specific tag, attribute, or value of the XML file"* (based on fig. 1, condensed XML file 120 and XML file 100, and paragraph [0021]),

- *"wherein a first portion of each of the respective pairs of characters identifies the tag index, the attribute index, or the value index, and a second portion of each of the respective pairs of characters identifies an index value in the tag index, the attribute index, or the value index"* (the skilled person would derive, directly and unambiguously, from fig. 1, condensed XML file 120 and index 110, and paragraph [0021], the general teaching that a "first portion" of each pair of characters, i.e. its first character (e.g. "T" in "T2"), identifies which index is relevant (e.g. the tag index for "T"), and a "second portion" of the pair, i.e. its second character ("2" in "T2"), identifies the relevant index value in that index (hence "T2" stands for tag "defaults" according to index 110)).

Finally, claim 1 also includes the feature:

- *"wherein the tag index, the attribute index, the value index, and the condensed version of the XML file are stored in memory in a compressed format"* (based on the claim-like "paragraph 11" on page 15).

6.2 The additional feature of claim 2 is based on paragraph [0031] (see also original claim 3).

6.3 Accordingly, the board finds that claims 1 and 2 comply with Article 123(2) EPC.

7. Article 76(1) EPC

The parent application as filed includes the claim-like "paragraphs" found on pages 14-16 of the present application as actual claims. It also contains the same description and drawings as the present application as filed.

Accordingly, the board considers that claims 1 and 2 also comply with Article 76(1) EPC.

Article 84 EPC

8. The objections raised by the board under Article 84 EPC in its preliminary opinion (points 15 to 18) — in particular, those concerning product-by-process features and features relating to actions carried out by an entity different from the one claimed (e.g. actions performed by a printer in a claim directed to a print container) — have been overcome through the amendments to claim 1 and the deletion of all other claims except claim 2.

9. Claim 1 refers to a "condensed version of the XML file" and specifies that it is stored in the print container's memory in a "compressed format".

9.1 The description draws a distinction between the terms "condensation" and "compression". The term "condensation" is consistently used to describe the "translation" of the original "XML file 100" into the "condensed XML file 120" (see fig. 1), which the skilled person would understand as a form of compression specifically adapted for XML data. The term "compression", by contrast, refers to a subsequent step involving a (general-purpose) compression technique applied to the condensed XML file and the three associated indexes, to further reduce their size.

9.2 Although the terms "condensation" and "condensed" are not established terms in the fields of print containers, XML files handling or data compression, the board considers that, in the context of the present application, it would be clear to the skilled person that the expression "condensed version of the XML file" in claim 1 refers to the compressed representation of the XML file that is specified in the claim. The storage of this version in a "compressed format" in claim 1 is also clear and implies that the "condensed version of the XML file" has been *further* compressed using some compression algorithm.

Inventive step, Article 56 EPC

10. The examining division found that claim 1 of the requests then on file lacked an inventive step, starting from either D1 or D2 in combination with D3 (contested decision, points 11 to 17).

In its preliminary opinion, the board agreed with the examining division's conclusion and also considered additional combinations, namely D2 in combination with either D7 or D4, instead of D3.

11. However, the board notes that present claim 1 differs substantially from the versions examined by the

examining division. In the light of these amendments, the board finds that present claim 1 involves an inventive step, Articles 52(1) and 56 EPC.

The various combinations of prior art documents considered will be discussed in detail below.

12. Starting point

12.1 Documents D1 and D2, like claim 1, relate to print containers storing printer operation instructions, and are therefore regarded as suitable starting points for assessing inventive step.

12.2 By contrast, D3, D4 and D7 are concerned with general XML compression methods, and do not suggest any application in the context of print containers or printer operation instructions. The board considers that starting from any of these documents would, in the present case, require the formulation of *artificial* objective technical problems and, as such, would not provide a basis for a convincing argument of lack of inventive step.

12.3 The disclosures of D1 and D2 are largely similar in the aspects relevant to claim 1. However, D2 includes some additional pointers towards the use of XML (as discussed below). In the present case, it is thus sufficient to consider D2 as starting point.

12.4 The board therefore selects D2 as starting point for assessing inventive step of claim 1.

13. Document D2

13.1 D2 discloses a print cartridge 100 (fig. 1A) designed to be inserted into a printer 102, which "may provide any appropriate supply to the printer including ink, dry toner, wet toner, liquids, other materials" (paragraphs [0014]-[0015]).

13.2 The print cartridge includes a memory 106, in which "setting data" 110 are stored. The setting data 110 "indicates how the printer should operate with the print cartridge by providing temperature settings [e.g. heating parameter 114 in fig. 1A], service parameters, printer functional data [116], [and] other data" (paragraphs [0012], [0015]; fig. 1A).

For example, heating parameter 612 may include "an indication of the temperature value to set a heater of the printer at while operating [the] print cartridge" (paragraph [0044]). Likewise, "printer functional data" 116 are defined as "data that determines how the printer will operate" (paragraph [0021]) and as functional data "that the host device [i.e. the printer] can use to set the parameters for operating the device with the consumable product [i.e. the print cartridge]" (paragraph [0039]).

13.3 In the embodiment of fig. 1B, the setting data 110 (including heating parameter 114 and printer functional data 116) are stored in memory 106 in a tag-length-value (TLV) format 150 (fig. 1B; paragraph [0024]).

D2 explains that "[d]ata stored in TLV formats are easily searched using generalized parsing functions" and that "TLV elements are often used in a binary format which makes parsing faster and the data smaller" (paragraph [0026]).

D2 also notes that "in some examples, it may be easier to generate XML from data stored in a TLV format than to make human inspection of the data" directly in the TLV format (paragraph [0026]).

Although the TLV format is used in this embodiment, D2 clarifies that "any appropriate format may be used to store the setting data", and that "the data may be stored using text based fields, static fields, other fields, or combinations thereof" (paragraph [0027]).

14. Comparison of claim 1 with D2

14.1 The board considers that D2 discloses, using the terminology of claim 1, **a print container** (fig. 1A, print cartridge 100), **comprising a supply of print material** (e.g. ink, paragraph [0014]), **a print material distributer** (implicit in a print cartridge providing ink to a printer, paragraph [0014]), **and a memory** (memory 106) **storing instructions that define actions to be performed by a printer into which the print container is to be installed and thereby control an operation of that printer** (heating parameter 106 and printer functional data 116 are such "instructions").

14.2 In its statement of grounds of appeal (page 6, second paragraph), the appellant argued that "the 'tags' [in D2] are merely identifiers/indicators of setting data, but do not, as such, control an operation of the printer", so that, contrary to the examining division's view, a "tag" in D2 is not an "instruction defining an action to be performed by a printer", as defined in claim 1.

The board agrees that D2 does not directly and unambiguously disclose that the tags in the TLV format shown in fig. 1B of D2 do, as such, represent a particular instruction for a printer. Paragraph [0024] merely states that the tag section indicates the category of data (e.g. printer functional data) associated to the data stored in the corresponding value section.

However, D2 does discloses that setting data — which include the heating parameter and printer functional data — are stored in the memory 106 in a TLV format. These data are instructions defining actions to be performed by a printer into which the print container is to be installed (see point 13.2 above).

14.3 Claim 1 differs thus from the disclosure of D2 in that:

F1 the printer operation **instructions** are encoded in an **XML file** comprising tag terms, attribute terms and value terms, where

F1.1 a **tag term** refers to an instruction defining an action to be performed by a printer into which the printer container is to be installed,

F1.2 an **attribute term** is used to specify a parameter of the action triggered by the tag term, and

F1.3 a **value term** is used in association with the attribute term to specify the parameter of the action triggered by the tag term the attribute term is to modify;

F1.4 the XML file **uses a known subset of XML syntaxes;**

F2 the **XML file** is **stored** in the memory of the print container as

F2.1 a **tag index** containing a list of tag terms from the XML file organized according to a frequency of the tag terms in the XML file,

F2.2 an **attribute index** containing a list of attribute terms from the XML file organized according to a frequency of the attribute terms in the XML file,

F2.3 a **value index** containing a list of value terms from the XML file organized according to a frequency of the value terms in the XML file, and

F2.4 a **condensed version of the XML file** containing **only pairs of characters** and **specialized symbols indicating that a most recent open tag should be closed**, wherein each of the pairs of characters corresponds to a specific tag, attribute, or value of the XML file, wherein a **first portion** of each of the respective pairs of characters **identifies the tag index**, **the attribute index**, **or the value index**, and a **second portion** of each of the respective pairs of characters **identifies a index value in the tag index**, **the attribute index**, **or the value index**,

F3 wherein the tag index, the attribute index, the value index, and the condensed version of the XML file are stored in memory in a **compressed format.**

14.4 The board agrees with the appellant (statement of grounds of appeal, page 7, first paragraph) that it is not the XML file that is organized according to a frequency of tag terms (as suggested by the formulation of differentiating feature f2 in the contested decision, point 11.2.1) but the list of tag terms contained in the tag index that is so organised. This distinction is reflected in feature F2.1 above.

14.5 The board also agrees with the appellant (statement of grounds of appeal, page 7, third paragraph) that, in the assessment of obviousness, feature F1 — which specifies that the instructions are encoded in an XML file comprising tag terms, attribute terms and value terms — and features F2 to F3 — which define a specific manner in which that XML file is stored — cannot be treated entirely separately from each other, as would be the case under a "partial problems" approach.

This is because, starting from D2, in which no XML file is used, feature F1 is a prerequisite for features F2 to F3 to serve any technical purpose, i.e. the latter are functionally dependent on the former. In such a situation, an *entirely* separate treatment of these two groups of features cannot be justified merely on the ground that there is no "combinatorial effect over the sum of the individual effects" (as arguably suggested in the contested decision, point 11.2.1, last paragraph).

Similarly, features F1.1 to F1.4 are functionally dependent on feature F1.

14.6 Nevertheless, the board observes that - despite the suggestion in point 11.2.1 of the contested decision - the examining division did not in fact treat the differentiating features entirely separately from each other. In point 11.2.2, the examining division explained why the skilled person would have arrived at differentiating features "f1" (essentially F1 and F1.1 to F1.3). In point 11.2.3, it argued that the skilled person "starting from either D1 or D2 and implementing storing using the standard XML format", i.e. having already incorporated feature f1, "would *then* be faced with the technical problem of storage limitation in printer cartridges (corresponding to features f2)" (emphasis by the board).

Such a step-by-step obviousness analysis may be justified under certain conditions (see T 1158/02 Vehicle web access/ICO SERVICES, point 3.9, T 1132/20 Multi-tenant cloud service/MICROSOFT, point 8.8, and Case Law of the Boards of Appeal, Edition 2022, I.D.9.21.10 "Several obvious steps"). The board adopts a similar approach in the following analysis.

14.7 The appellant argued in its letter of 21 May 2025 (points II.a.3 and 4) that the separation of the differentiating features into the above "categories" (F1, F1.1, ..., F2, F2.1, etc.) represents an artificial separation of aspects that necessarily belong together, allegedly leading to incorrect conclusions. As an example, the appellant pointed out that the "tag index" feature in claim 1 includes "tag terms", such that a definition of a "tag term" implies a limitation of the "tag index".

14.7.1 The board notes that patentability turns on the claimed subject-matter rather than the specific words used to define it. The same applies to the individual differentiating features. Thus, what matters is that the difference between the claimed invention and the prior art is correctly characterized in terms of distinguishing features and that these are duly taken into account in the assessment of inventive step.

In the present case, the limitation referred to by the appellant results from the combination of features F1, F1.1, F2 and F2.1.

14.7.2 Furthermore, the categorisation adopted by the board reflects a distinction between conceptual, functional or implementation aspects of the claimed invention and is, therefore, not "artificial" but a result of the board's claim construction. Disentangling these aspects clarifies their respective nature and role in the invention, as required to assess their contribution over the prior art.

For example, the fact that the tag terms listed in the *tag index* refer to instructions defining actions to be performed by a printer is merely a consequence of the semantic meaning attributed to the tag terms in the *original XML file*.

15. Objective problem

15.1 Encoding the instructions for the printer in XML - as specified in differentiating features F1 and F1.1 to F1.4 - has the effect of facilitating their readability by a human. Storing the XML file in a manner defined by features F2 to F3 has the effect of reducing its memory footprint when stored in the print container.

15.2 Accordingly, the differentiating features may be considered to solve, over D2, the following objective problem: *to provide an alternative way of encoding and storing the instructions for the printer in the memory of the print container that facilitates human inspection and has a small memory footprint*.

15.3 The board considers that a skilled person starting from D2 would have considered this problem without inventive activity, as D2 explicitly suggests that alternative formats may be used to store the setting data (paragraph [0027]), and also suggests that it is generally desirable to make the stored data "smaller" as well as to facilitate human inspection of the data (paragraph [0026]).

15.4 At the oral proceedings, the appellant argued that the formulation of the objective problem should not refer to human inspection or readability as the condensed XML file was not intended for that purpose. The appellant further submitted that the mention of an "alternative format" in paragraph [0027] of D2 should be understood as referring to alternatives to the binary format for storing the TLV elements discussed in paragraph [0026], and not as a suggestion to use XML instead of the TLV format.

The board is not persuaded by these arguments.

First, the effect of facilitating human inspection or readability arises from the use of XML - rather than a TLV format - to encode the printer operating instructions (features F1 and F1.1 to F1.4). This is acknowledged in the application itself, which states that XML is "used to encode documents in a format readable both by machines and people" (paragraph [0001]).

Second, regarding the interpretation of paragraph [0027] of D2, the board notes that this passage refers to an "alternative format [...] to store the setting data" and thus to alternatives to the choice made in the main embodiment of D2 of storing the setting data in the TLV format, as mentioned in the first sentence of paragraphs [0024] and [0026].

16. First step towards a solution and sub-problems

16.1 The board considers that it would have been obvious to a skilled person starting from D2 and faced with the objective problem to choose XML as the format to encode the printer instructions. The Extensible Markup Language (XML), which uses tags, attributes and values, was widely known as a format that is readable by both machines and humans (as acknowledged in the present application, paragraph [0001]). Furthermore, D2 itself highlights certain similarities between XML and the TLV format used in the embodiment of figure 1B (paragraph [0026]). Encoding the instructions directly in XML (instead of TLV) would have facilitated both the programming of the instructions and their readability. This leads to **feature F1**.

16.2 Having made the decision, in the first step, to encode the printer instructions in an XML file to address the objective problem, the skilled person would then necessarily have been confronted with two separate sub-problems:

(1) How to encode the printer instructions in the XML file?

(2) How to store the XML file in the memory of the print container such that it requires only a small memory footprint?

17. Sub-problem (1)

17.1 Solving sub-problem (1) by encoding a printer instruction in XML, such as setting the printer's heater temperature to 300°, using a line like

<heater temperature="300" unit="Celsius" />

would be a straightforward choice for the skilled person. This leads to **features F1.1 to F1.3**.

17.2 The appellant contests the examining division's view that features F1.1 to F1.3 represent "a trivial choice made in accordance with non-technical, cognitive modelling considerations" (contested decision, point 11.2.2). The appellant argues that instructions defining an action to be performed by a printer, along with associated parameters, were "control items for controlling an operation by a printer which, of course, refers to technical considerations related to the operation of a printer (i.e. controlling a machine, see G1/19, Nr. 85)" (statement of grounds of appeal, page 8, first paragraph).

The board interprets the examining division's argument on non-technicality as focusing on a programmer's choice of how to represent printer instructions (already disclosed in D2 and thus not part of the differentiating features) within the XML programming (or modelling) language. The board tends to agree with the examining division, particularly since the application does not indicate that the specific choice made in features F1.1 to F1.3 contributes to a technical effect *beyond* any (potential) technical effect associated with the printer instruction *itself* as a "control item".

However, the board considers that this point does not need further elaboration, as features F1.1 to F1.3 would, in any case, have been straightforward for the skilled person.

17.3 Moreover, it is common practice to require an XML file to conform to a Document Type Definition (DTD) or an XML schema, which imposes constraints on the syntax and structure of the XML file (see e.g. D7, page 1029, second paragraph). In the present case, it would have been obvious to do so, for instance, in order to ensure interoperability with different printers. This leads to **feature 1.4.**

18. Sub-problem (2)

To address sub-problem (2), the skilled person would have naturally considered existing *XML file compression techniques.* In doing so, the skilled person would have equally considered the teachings of documents D3, D4 and D7, which disclose different approaches to XML compression.

However, as will be explained in the following sections, a combination of D2 with any of D3, D4 or D7 would not have led the skilled person to the claimed solution.

19. Sub-problem (2) - Combination of D2 with D3

19.1 Document D3

19.1.1 D3 addresses the problem of reducing the data volume required for the transmission and storage of a markup language file, such as XML, while maintaining efficient data access. This is presented as being particularly relevant in the context of a "portable equipment with a small data storage capacity and low processing ability". The transmission may occur over a network or via media with limited capacity (paragraphs [0003]-[0006], [0053], [0010], [0011], [0019] and [0046] "XML").

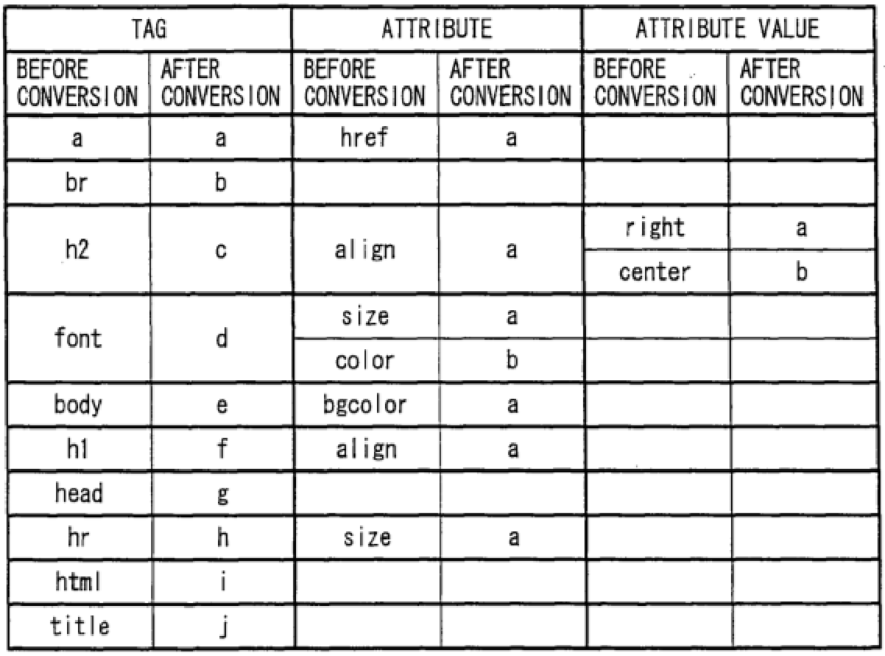
The markup language file to be "converted" (compressed) is referred to as "description tag data". An example of such a file is shown in Figure 34 and comprises tags, attributes and attribute values (paragraph [0013]).

The converted version of this file is referred to as "conversion tag data" (paragraph [0029]). It is generated by replacing the tag names, attribute names and attribute values in the original file with shorter character strings (e.g. "a", "b", "c", ...) according to a "conversion rule" (paragraphs [0033], [0042]).

19.1.2 There are two embodiments in D3 as regards how the conversion rule is generated. The most relevant one, relied upon by the examining division, is the "second embodiment" (described from paragraph [0095] onwards).

In this embodiment, a conversion rule as shown in Figure 21 is generated "dynamically from each contents", i.e. generated *from the description tag data* (the markup language file) to be converted (paragraphs [0108], [0132]).

FIG. 21



TAG		ATTRIBUTE		ATTRIBUTE VALUE	
BEFORE CONVERSION	AFTER CONVERSION	BEFORE CONVERSION	AFTER CONVERSION	BEFORE CONVERSION	AFTER CONVERSION
a	a	href	a		
br	b				
h2	c	align	a	right	a
				center	b
font	d	size	a		
		color	b		
body	e	bgcolor	a		
h1	f	align	a		
head	g				
hr	h	size	a		
html	i				
title	j				

The first column lists the distinct tag names appearing in the markup language file, sorted in descending order of frequency of appearance in the file (paragraphs [0105], [0107]; fig. 20).

The second column assigns each tag name a replacement character string, allocated "in alphabetic order starting from "a" [...] in descending order of the number of appearances of the [tag names] in the description tag data" (paragraph [0107]). For instance, tag name "body" will be replaced by string "e" in the conversion process.

The third column comprises the attribute names appearing in the markup language file. For each row in the first column corresponding to a tag (e.g. "font"), the third column comprises as many (sub-)rows as there are distinct attribute names (e.g. "size" and "color") appearing in the file *in association with that particular tag*, and *these attribute names* are listed according to their frequency of appearance in the file *in association with that tag in the file* (see fig. 20: 2 resp. 1 appearance for attribute name "size" resp. "color" in association with tag name "font" in the file shown in fig. 34) (paragraphs [0107], [0113]). An attribute name may thus appear several times in the third column, namely if it appears in association with different tag names in the description tag data (e.g. the attribute name "size" in fig. 21, as it is used with the tags "font" and "hr" in the file shown in fig. 34).

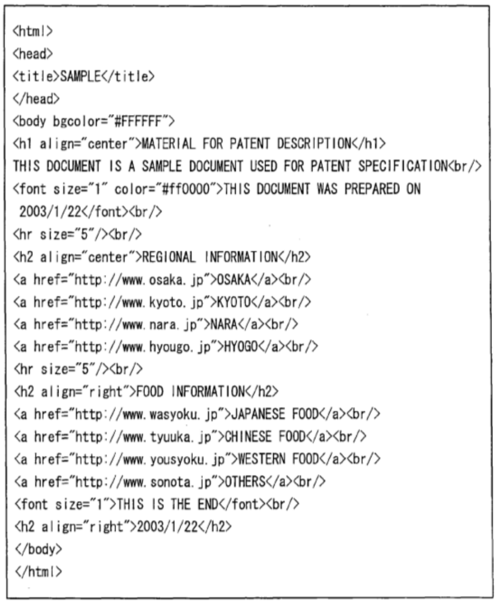
The fourth column assigns each attribute name in a row of the third column (and thus in association with a particular tag) a character string into which this attribute name will be replaced *whenever it is used in association with that tag name in the file*. The character strings are allocated to the attribute names *appearing in association with a particular tag name in the file* alphabetically, starting with "a", in decreasing number of appearance in the file *in association with that particular tag name* (e.g. "size" to "a" and "color" to "b" when associated with tag "font") (paragraph [0107]; fig. 20, 21 and 34).

The fifth and sixth columns are organized in a similar manner as the third and fourth columns but for the value names ("attribute values") instead of the attributes names and in association with particular attribute names instead of with tag names (paragraph [0107]; fig. 20, 21 and 34).

19.1.3 D3 teaches further to store the "conversion rule" (fig. 21) in the compact form of a "tag conversion table" as illustrated in fig. 22 (paragraphs [0108], [0109]).

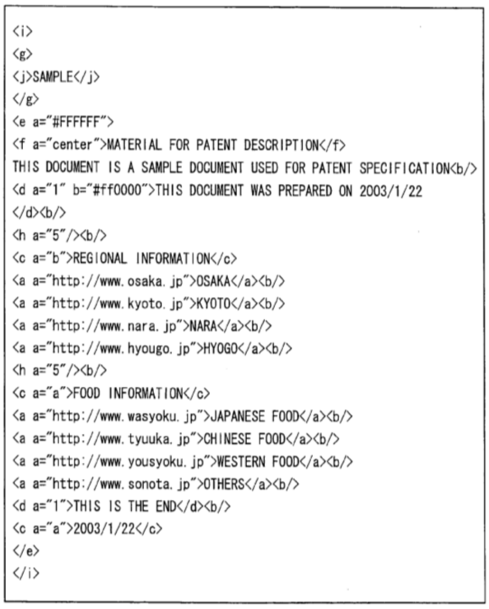
19.1.4 Using this table, the description tag data (the markup language file) shown in fig. 34 is converted into the "converted tag data" shown in fig. 23 by a "tag conversion process" (paragraph [0110]).

FIG. 34



```
<html>
<head>
<title>SAMPLE</title>
</head>
<body bgcolor="#FFFFFF">
<h1 align="center">MATERIAL FOR PATENT DESCRIPTION</h1>
THIS DOCUMENT IS A SAMPLE DOCUMENT USED FOR PATENT SPECIFICATION<br/>
<font size="1" color="#ff0000">THIS DOCUMENT WAS PREPARED ON
 2003/1/22</font><br/>
<hr size="5"/><br/>
<h2 align="center">REGIONAL INFORMATION</h2>
<a href="http://www.osaka.jp">OSAKA</a><br/>
<a href="http://www.kyoto.jp">KYOTO</a><br/>
<a href="http://www.nara.jp">NARA</a><br/>
<a href="http://www.hyougo.jp">HYOGO</a><br/>
<hr size="5"/><br/>
<h2 align="right">FOOD INFORMATION</h2>
<a href="http://www.wasyoku.jp">JAPANESE FOOD</a><br/>
<a href="http://www.tyuuka.jp">CHINESE FOOD</a><br/>
<a href="http://www.yousyoku.jp">WESTERN FOOD</a><br/>
<a href="http://www.sonota.jp">OTHERS</a><br/>
<font size="1">THIS IS THE END</font><br/>
<h2 align="right">2003/1/22</h2>
</body>
</html>
```

FIG. 23



```
<i>
<g>
<j>SAMPLE</j>
</g>
<e a="#FFFFFF">
<f a="center">MATERIAL FOR PATENT DESCRIPTION</f>
THIS DOCUMENT IS A SAMPLE DOCUMENT USED FOR PATENT SPECIFICATION<b/>
<d a="1" b="#ff0000">THIS DOCUMENT WAS PREPARED ON 2003/1/22
</d><b/>
<h a="5"/><b/>
<c a="b">REGIONAL INFORMATION</c>
<a a="http://www.osaka.jp">OSAKA</a><b/>
<a a="http://www.kyoto.jp">KYOTO</a><b/>
<a a="http://www.nara.jp">NARA</a><b/>
<a a="http://www.hyougo.jp">HYOGO</a><b/>
<h a="5"/><b/>
<c a="a">FOOD INFORMATION</c>
<a a="http://www.wasyoku.jp">JAPANESE FOOD</a><b/>
<a a="http://www.tyuuka.jp">CHINESE FOOD</a><b/>
<a a="http://www.yousyoku.jp">WESTERN FOOD</a><b/>
<a a="http://www.sonota.jp">OTHERS</a><b/>
<d a="1">THIS IS THE END</d><b/>
<c a="a">2003/1/22</c>
</e>
</i>
```

19.1.5 Optionally, the converted tag data may be further compressed by applying an "arbitrary compression scheme such as Huffman coding or Zip" (paragraphs [0045], [0122]).

19.1.6 The "converted tag data", together with the "tag conversion table", are transmitted to and stored in the portable equipment with small storage capacity (paragraph [0112], the "viewer" being this portable equipment in an example, see paragraphs [0003]-[0006], [0011]).

19.1.7 D3 teaches that it is possible to reconstruct the original "description tag data" using the "converted tag data" and the "tag conversion table". First, a "reverse-conversion table" is generated from the "tag conversion table" (as shown in fig. 12 as reverse table to the table in fig. 6). Then it is used to replace the strings in the converted tag data by the corresponding tag names, attributes names and attribute values (paragraphs [0071]-[0076]; fig. 11).

19.2 The board considers that the skilled person, starting from D2 and applying the teaching of D3 to address sub-problem (2), would **not** have arrived at **features F2.2, F2.3 and F2.4**.

19.2.1 In the terminology of present claim 1, the description tag data in D3 (fig. 34) would correspond to an "XML file", the conversion rule (fig. 21) to an index comprising a "tag index" (first and second columns), an "attribute index" (third and fourth columns) and a "value index" (fifth and sixth columns), and the conversion tag data (fig. 23) to a "condensed version of the XML file".

19.2.2 However, the board agrees with the appellant (and departs in this respect from its preliminary opinion) that the attribute terms in the third column of the conversion rule do not constitute "a list of attribute terms [...] organized according to a frequency of the attribute terms in the XML file", as required by feature F2.2.

As explained at point 19.1.2 above, each entry in the third column of the conversion rule represents an attribute term *in association with a particular tag term* (see fig. 21: "align" in the third row stands for attribute "align" in the context of tag "h2"). These entries are primarily sorted and thus "organized" according to the frequency of appearance of the associated *tag term* in the XML file (see fig. 21: the position of "align" in the third row is determined by the frequency of tag "h2" in the file, as illustrated in fig. 20, not by the frequency of the attribute "align" as such).

The same reasoning applies to the values in the fifth column of the conversion rule, which do not satisfy the requirements of feature F2.3.

19.2.3 Moreover, the conversion method in D3 replaces tag terms, attribute terms and attribute values in the description tag data by codewords (e.g. "a", "b", "c") according to the conversion rule, but retains all other symbols that define the syntactic structure of the description tag data (e.g. <, >, </,/>, =, ", etc.) in the conversion tag data (see fig. 23 compared with fig. 34).

Even if the codewords were implemented as pairs of characters (e.g. "00", "01", "02", etc.), the resulting conversion tag data - and thus any purported "condensed version of the XML file" - would not meet the requirements of feature F2.4, which specifies that condensed XML file must contain "*only* pairs of characters and specialized symbols indicating that a most recent open tag should be closed" (emphasis by the board). Furthermore, these pairs of characters would not comprise a "first portion identif[ying] the tag index, attribute index, or the value index", as required by this feature.

20. Sub-problem (2) - Combination of D2 with D7

20.1 Document D7

20.1.1 D7 relates generally to the compression of XML documents.

20.1.2 In section 3, first to third paragraphs, D7 sets out general considerations for XML compression. It observes that XML documents exhibit significant redundancy, particularly "with tag and attribute names, but attribute values or some of the element content words can also appear many times". D7 notes that "the list of frequent words vary greatly between XML documents with different content types" and that "this redundancy can be exploited with the help of a semi-dynamic dictionary" created "in a preliminary scan over the document to form the dictionary". Following this approach, "every time a dictionary word is found in the document, it can be replaced by its dictionary index". If properly encoded, dictionary indices are always shorter than the words they reference. While the semi-dynamic dictionary must be stored explicitly in the output generated by the preprocessor (which is encoding the XML document using the semi-dictionary), the size (in bytes) of the stored dictionary is relatively small compared to the total encoded length of the document, as the words included in the dictionary are highly frequent.

20.1.3 In the subsequent sections, D7 discloses a specific XML compression method named "XML-WRT", based on the aforementioned principles. This method employs a *single dictionary* that includes the *most frequent words* found the XML document, which may include tag terms, attribute terms and values (section 4.3). The dictionary entries are sorted based on their frequency of appearance in the XML file (section 4.2; section 4.6, second paragraph). In the compressed file, these words are replaced with codewords generated from the dictionary using a byte-oriented prefix code (section 4.6, third paragraph).

20.1.4 The words selected for the dictionary are written, along with separators, at the beginning of the output file (section 4.2, second paragraph; section 4.3, last sentence). The output file is then subjected to a general-purpose compression algorithm (section 4.6, third and sixth paragraphs, "back-end compressor").

20.2 The board considers that the skilled person, starting from D2 and applying the teaching of D7 to address sub-problem (2), would **not** have arrived at **feature F2.4**.

20.2.1 A direct application of the XML-WRT compression method disclosed in D7 would result in the use of a single dictionary, not in the use of three separate dictionaries/indexes for tag terms, attribute terms and values, as required by features F2.1 to F2.4.

20.2.2 In its preliminary opinion, the board had suggested that the general considerations in section 3 of D7 would have rendered obvious to the skilled person to use three separate semi-dynamic dictionaries for tag terms, attribute terms and values, and further to apply this approach to all such terms, not just the most frequent ones. The appellant contested both assertions.

However, this point may be left open. Even assuming that the skilled person would have considered using three separate dictionaries, they would not have arrived at feature F2.4 for the following reasons: Like D3, D7 does not teach or suggest to keep in the output file *only* the codewords and specialised symbols indicating that a most recent tag should be closed. In the specific method disclosed in D7, in which a single dictionary is used and the codewords do not indicate by themselves whether the encoded term is an tag, attribute or value, it would not be readily possible to reconstruct the syntactic structure of the original XML file solely from the sequence of codewords and tag-closure symbols. There is no suggestion in D7 that using three separate dictionaries/indexes for tag terms, attribute terms and values in combination with codewords comprising a first portion identifying the respective index - and thus the category of the encoded term - could *enable* a compression method in which the output file contains *only* such codewords and specialised tag-closing symbols, while still allowing reliable reconstruction of the syntactical structure of the XML file from the sequence of codewords and tag-closing symbols.

21. Sub-problem (2) - Combination of D2 with D4

21.1 Document D4

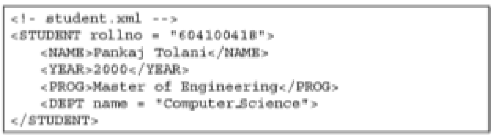
21.1.1 D4 discloses an XML compression method named "XGRIND", which retains the structure of the original XML document and supports querying in the compressed domain - a feature particularly useful when the XML document represents a database (D4, abstract).

The method applies a dictionary-based compression approach separately to tag terms and attribute terms, while attribute values are encoded using a context-free compression scheme.

21.1.2 For XML documents that conform to a Document Type Definition (DTD), the tag and attribute dictionaries are generated by parsing the DTD (fig. 3) associated with the XML document (fig. 2) (section 3.4). Each tag term is encoded as "T" followed by a unique tag-ID (i.e. T0, T1, ...), and each attribute term by as "A" followed by a unique attribute-ID (i.e. A0, A1, ...) (sections 3.1.1, 3.3, second paragraph, and 3.4).

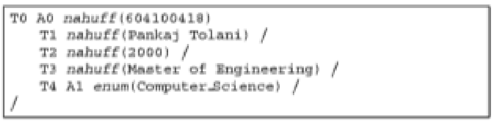
21.1.3 Attribute values are encoded using a non-adaptive Huffman compression algorithm or, if they are of an enumerated type, a $\log_2 K$ encoding scheme is applied (sections 3.1.2 and 3.1.3).

21.1.4 The resulting compressed XML file comprises only the pairs of characters corresponding to tag terms (T0, T1, ...) and attribute terms (A0, A1, ...), the encoded attribute values (fig. 4: nahuff(...), enum(...)), and the special symbol "/" to indicate the end of a last tag (sections 3.1.1, 3.3, 3.4), as illustrated in figure 4 (the original XML file is shown in fig. 2):



```
<!- student.xml -->
<STUDENT rollno = "604100418">
   <NAME>Pankaj Tolani</NAME>
   <YEAR>2000</YEAR>
   <PROG>Master of Engineering</PROG>
   <DEPT name = "Computer_Science">
</STUDENT>
```

Figure 2. Fragment of the Student DB



```
T0 A0 nahuff(604100418)
   T1 nahuff(Pankaj Tolani) /
   T2 nahuff(2000) /
   T3 nahuff(Master of Engineering) /
   T4 A1 enum(Computer_Science) /
/
```

Figure 4. Abstract view of XGrind document

21.2 The board considers that the skilled person, starting from D2 and applying the teaching of D7 to solve sub-problem (2), would **not** have arrived at **features F2.3 and F2.4**.

21.2.1 The dictionaries created in D4 for the tag and attribute terms correspond, respectively, to a "tag index" and an "attribute index". However, the "unique element/attribute IDs" associated to the entries in these dictionaries, i.e. the index values, are "determined by the DTD parser", that is they are derived from the DTD itself (section 3.4). As a result, these index values do not reflect the frequency of appearance of the tag or attribute terms *in the XML file*.

That said, the board considers that it would have been obvious for the skilled person to additionally sort the entries in the tag and attribute dictionaries by frequency of occurrence, as this is a common approach in dictionary-based compression methods to improve compression efficiency (see e.g. textbook D8, section 6.2, first paragraph). The required frequencies could be readily computed by the "XML Parser", which already computes other frequencies for attribute values (see section 3.3, first paragraph). On this basis, the skilled person would have arrived at features F2.1 and F2.2.

21.2.2 However, the board is not convinced that the skilled person, *having started from D2 and performed all the steps considered so far, including relying on D4 to address sub-problem (2)*, would have been motivated to *further modify* the compression method of D4 to encode the attribute values using a third dictionary, in place of the context-free compression techniques taught in D4.

The board notes that the XML documents considered in D4 are primarily databases (see section 4.1), and in such contexts, attribute values typically do not exhibit sufficient redundancy to justify the overhead of building and storing a dictionary for them.

To proceed further, the skilled person would have had to recognise that the attribute values in an XML file encoding printer operation instructions in a print container may show greater redundancy.

The board finds that considering such a further step to be obvious would overstretch the chain of reasoning starting from D2 and relying on D4 to address sub-problem (2).

21.2.3 For similar reasons, the board does not consider it obvious that, at this stage of the problem-solving-process, the skilled person would have been motivated, to consult D7 to further modify the method of D4 when applying it to address sub-problem (2).

21.2.4 Hence, the skilled person would not have arrived at feature 2.3 and feature 2.4 (insofar as it is dependent on the value index defined in feature 2.3).

22. Conclusion

22.1 The board considers that at least sub-problem (2) - how to store the XML file in the memory of the print container such that it requires only a small memory footprint? - is a *technical* problem. Claim 1 makes thus a non-obvious technical contribution over D2 and, for similar reasons, over D1.

22.2 The board therefore judges that claim 1 involves an inventive step, Articles 52(1) and 56 EPC. This also applies to claim 2 as it is dependent on claim 1.

22.3 The claims are allowable but the description still needs to be adapted to them. The case is remitted to the first-instance for that purpose, Article 111(1) EPC. The appellant had no objection to this course of action.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the examining division with the order to grant the patent on the basis of claims 1 and 2 of the auxiliary request 6 filed during the oral proceedings before the board and a description to be adapted thereto.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

L. Stridde Martin Müller

Decision electronically authenticated