

**Internal distribution code:**

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

**Datasheet for the decision  
of 5 December 2017**

**Case Number:** T 1010/12 - 3.5.04

**Application Number:** 04779468.0

**Publication Number:** 1676245

**IPC:** G06T15/00

**Language of the proceedings:** EN

**Title of invention:**

System and method for a unified composition engine in a graphics processing system

**Applicant:**

Microsoft Technology Licensing, LLC

**Headword:**

**Relevant legal provisions:**

EPC 1973 Art. 84

**Keyword:**

Clarity - main request and auxiliary request (no)

**Decisions cited:**

**Catchword:**



**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  
Fax +49 (0)89 2399-4465

Case Number: T 1010/12 - 3.5.04

**D E C I S I O N**  
**of Technical Board of Appeal 3.5.04**  
**of 5 December 2017**

**Appellant:** Microsoft Technology Licensing, LLC  
(Applicant) One Microsoft Way  
Redmond, WA 98052 (US)

**Representative:** Goddar, Heinz J.  
Boehmert & Boehmert  
Anwaltspartnerschaft mbB  
Pettenkoferstraße 22  
80336 München (DE)

**Decision under appeal:** Decision of the Examining Division of the  
European Patent Office posted on 10 November  
2011 refusing European patent application  
No. 04779468.0 pursuant to Article 97(2) EPC.

**Composition of the Board:**

**Chairman** C. Kunzelmann  
**Members:** B. Willems  
B. Müller

## **Summary of Facts and Submissions**

- I. The appeal is against the decision of the examining division dated 10 November 2011 refusing European patent application No. 04 779 468.0, which has been published as international application WO 2005/045580 A2.
  
- II. The documents cited in the decision under appeal included the following:  
  
D1: EP 1 304 656 A2.
  
- III. The application was refused on the grounds that claim 1 of the then main request did not meet the requirements of Article 84 EPC and the subject-matter of claim 1 of the then auxiliary request lacked inventive step over the disclosure of D1 combined with the common general knowledge of the person skilled in the art (Article 56 EPC).
  
- IV. The applicant filed an appeal requesting that this decision be set aside. With the statement of grounds of appeal, the appellant filed claims according to a main request and an auxiliary request. It requested that a patent be granted on the basis of these claims.
  
- V. The board issued summons to oral proceedings. In a communication under Article 15(1) RPBA (Rules of Procedure of the Boards of Appeal, OJ EPO 2007, 536), annexed to the summons, the board gave its provisional opinion that claim 1 of the main request and the auxiliary request was not clear (Article 84 EPC), the subject-matter of claim 1 of the auxiliary request did not meet the requirements of Article 123(2) EPC and the subject-matter of claim 1 of both requests lacked

inventive step over the disclosure of D1 combined with the common general knowledge of the person skilled in the art (Article 56 EPC).

VI. With a letter dated 3 November 2017 the appellant filed amended claims according to a main request and an auxiliary request, both requests replacing the previous requests on file. It submitted reasons as to why the amended claims met the requirements of Articles 56 and 84 EPC.

VII. The board held oral proceedings on 5 December 2017. The appellant was represented.

The appellant requested that the decision under appeal be set aside and a European patent be granted on the basis of the claims of the main request or the auxiliary request filed with the letter dated 3 November 2017.

At the end of the oral proceedings, the chairman announced the board's decision.

VIII. Claim 1 of the main request reads as follows:

"A computing system comprising:

a visual system (214, 401), the visual system (214, 401) being configured to receive calls from one of a program and a desktop window manager to construct a hierarchical data structure (404) representing a displayable scene,

a master resource table (402) included in the visual system (214, 401), wherein the master resource table (402) comprises a list of resources used by the program

or the desktop window manager and contains device and resolution independent data for each resource, the master resource table (402) being configured to manage the lifetime of the resources via reference counting;

a unified composition engine (218, 420) being configured to receive data from the visual system (214, 401) including a representation of the hierarchical data structure (404) along with the appropriate resources, wherein the unified composition engine (218, 420) refers to the resources by handle;

a slave resource table (414) included in the unified composition engine (218, 420), wherein the slave resource table (414) includes a list of the resources sent from the master resource table (402) to the unified composition engine (218, 420), wherein the resources in the list of the resources in the slave resource table (414) also exist in the master resource table (402), wherein

the unified composition engine (218, 420) is configured to construct a compositor data structure (412) in response to the data received from the visual system (214, 401) to provide graphics output; and wherein

the master resource table (402) is configured to control the lifetime of slave resource table (414) resources via the reference counting and serialized requests via a change queue (408) of the unified composition engine (218, 420), and the slave resource table (414) is configured to not perform reference counting".

IX. Claim 1 of the auxiliary request reads as follows (features added to claim 1 of the main request are in *italics*):

"A computing system comprising:

a visual system (214, 401), the visual system (214, 401) being configured to receive calls from one of a program and a desktop window manager to construct a hierarchical data structure (404) representing a displayable scene,

a master resource table (402) included in the visual system (214, 401), wherein the master resource table comprises a list of resources used by the program or the desktop window manager and contains device and resolution independent data for each resource, the master resource table (402) being configured to manage the lifetime of the resources via reference counting;

a unified composition engine (218, 420) being configured to receive data from the visual system (214, 401) including a representation of the hierarchical data structure (404) along with the appropriate resources, wherein the unified composition engine (218, 420) refers to the resources by handle;

a slave resource table (414) included in the unified composition engine (218, 420), wherein the slave resource table (414) includes a list of the resources sent from the master resource table (402) to the unified composition engine (218, 420), wherein the resources in the list of the resources in the slave resource table (414) also exist in the master resource table (402);

*a notify queue (410) and a change queue (408) included in the unified composition engine (218, 420), wherein*

*the unified composition engine (218, 420) is configured to construct a compositor data structure (412) in response to the data received from the visual system (214, 401) to provide graphics output; wherein*

*the master resource table (402) is configured to control the lifetime of slave resource table (414) resources via the reference counting and serialized requests via the change queue (408), and the slave resource table (414) is configured to not perform reference counting; and wherein, if a resource lookup in the slave resource table (414) fails, the unified composition engine (218, 420) is configured to queue a message via the notify queue (410) to the visual system (214, 401) and skip the processing requiring that resource".*

- X. In view of the amendments made to the claims, the clarity objections raised in the decision under appeal are not relevant for the present decision. However, in a communication annexed to the summons to oral proceedings, the examining division had raised other clarity objections which were no longer pursued in the decision under appeal. Since these other objections are relevant to the present decision, they are summarised below:

It was not clear what was meant by the master resource table controlling the lifetime of the resources of the slave resource table. In particular, it was not clear what was meant by "*controlling the lifetime*" of resources, as this could be interpreted in a variety of ways. For example, "*controlling the lifetime*" may be



interpreted as administrating the validity of resources, or as actively allocating and de-allocating these resources (see annex to the summons, point 3.2).

XI. The appellant's arguments which are relevant to the present decision may be summarised as follows:

The lifetime of a resource indicated for how long or how often the resource can be used. The lifetime was determined using reference counting, the latter being defined in the description, page 16, lines 31 and 32, as referring "*to the number of time a particular resource is utilized*". Hence, reference counting could mean either incrementing/decrementing the counter when the resource is accessed/no longer accessed or simply incrementing the counter each time an application accessed a resource. Reference counting was not to be limited to its implementation in the known practice of garbage collection, i.e. incrementing/decrementing a counter and de-allocating a resource which was no longer referenced. Moreover, "*to control the lifetime of slave resource table resources*" was broad, as it was left open whether at the end of the resource's lifetime a pointer to the resource was deleted from a list or the resource itself was de-installed. However, this broad definition did not result in a lack of clarity.

The functional relationship between the visual system and the composition engine was clearly set out in the description. The visual system acted as a client of the composition engine (see page 10, line 32; page 11, lines 13 to 19; page 13, lines 25 to 27) which combined the lower-level composition for both the API composition engine and the desktop composition engine (page 12, lines 15 to 19). The system and the engine

communicated via a change queue or a notify queue of an interface (see Figure 4), but operated at different rates (see page 11, lines 3 to 5). Nevertheless, in each visual system there was a composition component that was decoupled from the components implementing the rendering of graphics from the scenes of multiple applications. "At times" some of the rendering could happen at higher levels (see page 12, lines 7 to 14).

### **Reasons for the Decision**

1. The appeal is admissible.
2. *Main request - clarity (Article 84 EPC 1973)*
  - 2.1 Claim 1 specifies "*the master resource table (402) being configured to manage the lifetime of the resources via reference counting*".
  - 2.2 It is not clear how the master resource table manages the lifetime of these resources via reference counting.
  - 2.3 The appellant submitted that the feature quoted in point 2.1 above is not to be interpreted as limited to reference counting in garbage collection. Lifetime management via reference counting encompassed various possibilities of controlling for how long or how often a resource could be used. The appellant agreed with the board that the claimed lifetime management could include, for instance, barring access to a resource once the number of accesses had exceeded a number set in a DRM (digital rights management) licence.

2.4 The board has not been convinced by the appellant's argument that this broad, general definition of lifetime management does not affect the clarity of the claim.

Although the clarity of a claim is not diminished by the mere breadth of terms in it, upon proper interpretation the meaning of these terms has to be unambiguous.

As conceded by the appellant, in the context of the application, the expression in claim 1 "*to manage the lifetime*" of resources is not to be interpreted in a narrow manner to mean de-allocating a resource if it was no longer referenced. Instead, any method operating on an entry in a table or on the resource itself and controlling how often or for how long the resource could be used (via an unspecified manner of reference counting) would manage **a** lifetime of the resource. However, the technical meaning of "lifetime" depends on the method's purpose (e.g. DRM or resource allocation) and so does *a fortiori* the technical meaning of "*to manage the lifetime*". Thus, "*to manage the lifetime*" of a resource is ambiguous, because the term "lifetime" is ill-defined and the nature of the management is left open. This ambiguity is increased by the further feature that the "control" of the lifetime rather than its management is "*via the reference counting and serialized requests via a change queue*".

2.5 Claim 1 specifies a "*unified composition engine [which] is configured to construct a compositor data structure (412) in response to the data received from the visual system*". Claim 2 specifies that the "*unified composition engine (218, 420) comprises a first composition service decoupled from a second composition*

*service, the first composition service incorporated into the visual system".*

2.6 From the term "unified" it is inferred that all components constructing the compositor data structure are integrated in the engine. This interpretation is confirmed by the description; see page 3, lines 27 and 28 ("*Combining the two composition efforts...*") and page 12, lines 15 to 19 ("*It is the lower-level composition described above for the API composition engine that is now shared between both the API composition engine and the desktop composition engine. Combining the lower-level composition for the both the API composition engine and the desktop composition engine results in the unified composition engine of the present invention*").

2.7 However, this interpretation is contrary to claim 2, which specifies the integration of some services of the composition engine in the visual system. Reading claims 1 and 2 in conjunction, the question arises as to which tasks are carried out by the visual system and which by the composition engine.

2.8 The appellant reasoned that the distinction between the visual system at a higher level and the composition engine at a lower level was clear from *inter alia* the following passages of the description:

(a) page 11, lines 1 to 5: "*the visual system 214 includes a user interface component that typically performs more computationally-expensive operations than the compositor 218, and thus this aspect of the visual system 214 typically operates at a relatively slower rate with respect to the operating rate of the compositor*";

(b) page 12, lines 7 to 14: "*there is a composition component in each visual system that is decoupled from the composition components (in a different process) that implement the rendering of graphics from the scenes of multiple applications. Note, that **at times** it is advantageous for some of the rendering to happen at higher levels*" (emphasis added).

- 2.9 The board has not been persuaded by the argument that the quoted passages clearly disclose which tasks are performed at the higher level or at the lower level. The allocation of tasks is flexible and according to unspecified criteria, i.e. at times some of the "services" of the composition engine are allocated to the visual system. Thus, any distinction between the visual system and the composition engine implied by the wording of claim 1 is based on unspecified criteria.
- 2.10 This lack of a clear distinction between the visual system and the composition engine is further demonstrated by the fact that the wording "*a change queue (408) of the unified composition engine*" in claim 1 specifies that the queue is implemented in the composition engine. It is, however, apparent from page 14, lines 7 and 8, and Figure 4 that the queue is implemented in the composition device **interface** and used for queuing messages to be sent from the visual system to the composition engine. The queue is not implemented in the composition engine as such.
- 2.11 It follows from the above that claim 1 of the main request does not meet the requirements of Article 84 EPC 1973. Hence, the main request is not allowable.

3. *Auxiliary request - clarity (Article 84 EPC 1973)*
- 3.1 Claim 1 of the auxiliary request includes the wording objected to in section 2 above.
- 3.2 The appellant argued that the additional features of claim 1 of the auxiliary request would clarify the claim.
- 3.3 The board is not convinced that additionally specifying what happens if a resource lookup in the slave resource table fails can remedy the lack of a clear distinction between the tasks allocated to the visual system and those allocated to the composition engine. Moreover, the additional features do not resolve the ambiguity arising from the wording "*to manage the lifetime*" of resources.
- 3.4 It follows from the above that claim 1 of the auxiliary request does not meet the requirements of Article 84 EPC 1973. Hence, the auxiliary request is not allowable.
4. Since neither of the appellant's requests is allowable, the appeal is to be dismissed.

**Order**

**For these reasons it is decided that:**

The appeal is dismissed.

The Registrar:

The Chairman:



K. Boelicke

C. Kunzelmann

Decision electronically authenticated