REMARKS

Claims 36-56 are pending in the application.

Claims 36-56 have been rejected.

Claims **36**, **44**, **46**, **49**, **52**, and **53** have been amended. Support for the amendments can be found throughout the originally filed Application, and at least at, ¶¶ [0026], [0030], [0032]-[0034], [0044], [0045], [0080], [0082], and [0090].

Rejection of Claims under 35 U.S.C. § 112, first paragraph

Claims 36-56 stand rejected under 35 U.S.C. § 112, first paragraph, as purportedly failing to comply with the written description requirement. Applicants respectfully traverse these rejections. Applicants also submit that the original specification provides explicit support for the elements of claim 36.

For example, in ¶[0044] and in Figure 2, the specification describes how to the sequence identifies and flags connectors and steps that are non-executable, or not sufficiently defined. In addition, at step 245 of Figure 2, the technical user is prompted to complete each flagged item which has been identified at step 240. This process checks whether blocks in question are connected, and if the blocks are not connected properly (and/or other errors are found), the system can receive information/further information that defines/further defines such elements/connections.

As such, claims 36, 46 and 52 comply with the written description requirement. Therefore, Applicants respectfully request that the rejections of claims 36-56 under 35 U.S.C. § 112, first paragraph, be reconsidered and withdrawn.

Rejection of Claims under 35 U.S.C. § 103

Claims 36-39, 43, 44, 46-47, 52-53 and 56 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lovell et al., U.S. Patent No. 7,370,315 ("Lovell") in view of Dunn, U.S. Patent No. 4,656,603 ("Dunn"). Applicants respectfully traverse this rejection. Applicants respectfully submit that Lovell and Dunn, either alone or in any rational combination, fail to teach or suggest all the elements of independent claims 36, 46 and 52. As will be described below, Lovell and Dunn fail to show, teach or even fairly suggest the features of these claims for the following reasons.

First, the cited sections of Lovell describe an integrated development environment that provides a visual representation of various physical and logical entities in a software model (e.g., the design surface) and the underlying code structures that support the entities (e.g., the source code editor). *See* Lovell, Abstract. As an initial matter, it is worth noting that the objects defined within the design surface are mapped directly to the underlying source code thus, providing the programmer the capability to update either the design surface or the source code, with such a change in one being reflected in the other on a one-to-one basis. *See id.* ("[t]he model is a graphical representation of the actual code, thus providing two way updating, i.e., the model is updated when the programmer changes the code and vice versa"). Therefore, unlike the claimed invention which makes a distinction between technical and non-technical users (and the user interfaces used thereby), Lovell requires that a programmer work with both the design surface and the source code.

According to the Office Action, Lovell's software design model can be interpreted as some manner of non-technical interface, and discloses modeling a business process by designing a process that represents a non-technical model of the business process. *See* Office Action, Pg. 4. Applicants respectfully disagree. Notwithstanding the discrepancies extant in even drawing such a parallel, Lovell's software design model provides only a single (technical) interface, and is further saddled with the problematic requirement that a change in one representation is immediately reflected in the other. This being the case, and ignoring other deficiencies, Lovell can only teach a single model (even assuming that Lovell's constructs, even as interpreted by the Office Action, can be properly characterized as somehow comparable to the claimed models in the first place (a point Applicants in no way concede)). This reading of Lovell is borne out by Lovell's "tight coupling." Therefore, the design of a software application using Lovell's approach is performed by using either the design surface or by manipulating the underlying source code (or a combination of both) by a programmer, with changes in one immediately reflected in the other. For example, Lovell's design surface provides a graphical view by which the programmer defines the components of the software application (such as software modules, interfaces, libraries, databases, etc.). *See* Lovell, Col. 6, ll. 7-10. Lovell's source code editor allows the programmer to directly edit the source code for the software application. *See id.*, ll. 10-11. Changes made graphically within the design surface are immediately apparent within the source code editor. *See id.*, ll. 16-17. For Lovell's system to operate otherwise would, in fact, render Lovell's system unfit for its intended purpose.

Given the foregoing, it is clear that Lovell suffers from several shortcomings. First, Lovell's software design model requires considerable knowledge of computer programming and software development, at least because even Lovell's design surface, which according to the Office Action supposedly represents a non-technical model, requires the definition of complex software components and the like. Second, Lovell not only fails to show, teach, or suggest any construct comparable to the claimed business process, but further, does not delegate the creation of any such construct between a non-technical user interface and a technical user interface. This is evident from the fact that changes made graphically within the design surface are immediately reflected in the source code editor. As such, Lovell merely provides a synchronized system that permits graphical changes to the underlying source code and vice-versa, and does not separate any two design functions, much less design functions comparable to the claimed modeling and designing of the business process between a technical user and non-technical user. Further in this regard, other distinctions notwithstanding, Lovell provides no showing, teaching or suggestion of multiple user interface types based on complexity of capabilities provided thereby (and certainly, that would allow for disparate skill levels). Further, using Lovell's approach, completion in one representation to mandates completion in the other - the claimed identifying and flagging have no meaning when there is no possibility of an incomplete module in one representation that might need to be noted for completion in another representation.

By marked contrast, the claimed invention models a business process by designing a process in which the designing is performed using a non-technical user interface by a non-technical user. *See* Application, ¶ [0026]. Only the implementation of the process is performed by a technical user upon transferring access of the process from the non-technical user interface to the technical user interface. Unlike Lovell, which requires that a programmer both design and implement a software solution, the claimed invention separates the design portion and implementation portion of the claimed business process by delegating the design portion to the non-technical user (e.g., a business analyst) and the implementation portion to the technical user (e.g., a programmer).

Second, according to the Office Action, the cited sections of Lovell purportedly disclose identifying an element where the element is an insufficiently-defined element and receiving information configured to allow the insufficiently-defined element to be completed. *See* Office Action, Pg. 4. Applicants respectfully disagree. Lovell's interface is attached to a software class, which, under any reasonable interpretation, is not equivalent to the claimed insufficiently-defined element. Even disregarding this shortcoming, Lovell's connection between interfaces is not equivalent to the claimed receiving information configured to allow the insufficiently-defined element to be completed because Lovell's interfaces are connected to the providers of the interface only after Lovell's programmer specifies whether the interface is required or optional. *See* Lovell, Col. 10, ll. 62-64. Moreover, the completion of the connection between interfaces takes place within the design surface itself and the programmer merely edits the properties of the connection. *See id.* Col. 11., ll. 38-42.

By further contrast, the claimed invention does not need any instruction from the technical user to determine whether to complete the insufficient-defined elements. Unlike Lovell, which requires the programmer to specify whether the interface is required or optional, the claimed invention's technical user merely completes the insufficient-defined element identified by the non-technical user (at the least, a user other than the technical user). In addition, unlike Lovell where the connection (and even its editing) takes place within the design surface (and not within the underlying source code), the claimed invention completes the insufficient-defined element in the technical user interface (and not in the non-technical user interface). In any case, Lovell provides no teaching as to the claimed completing of an insufficiently-defined element by a technical user in a technical user interface (after the insufficiently-defined element has been identified by a non-technical user) because Lovell is entirely silent on how connections are made between its interfaces and who makes such connections.

Third, and finally, the cited sections of Lovell do not show, teach or suggest transferring the claimed process from the non-technical user to the technical user and implementing the claimed process as a technical model of the business process using the technical user interface, as noted earlier. Lovell's graphical model of the software application to be developed involves a programmer (not a non-technical user) defining various components of the software application and their interactions. *See* Lovell, Col. 8, ll. 16-17. The programmer converts the components of the model into physical implementations by binding an object to a compiler corresponding to a programming language in which the object will be implemented. Not only does Lovell's design and implementation process require computer programming skills, the implementation of design changes in Lovell's system are effected immediately, in all representations, because Lovell's design surface merely visually illustrates the aforementioned binding.

By marked contrast, the claimed invention transfers the claimed process from the non-technical user to the technical user who then implements the claimed process as a technical model of the business process using the technical user interface. Unlike Lovell, the claimed implementation is not immediately apparent because implementation of insufficiently-defined elements is performed by the technical user using a technical interface after the insufficiently-defined elements and connectors are identified by the non-technical user. Moreover, Lovell does not provide any teaching comparable to the claimed transfer and delegation of the design and implementation portion of the claimed business process. For these reasons, the cited sections of Lovell do not show, teach or suggest the claimed features of independent claims 36, 46 and 52.

Furthermore, the cited sections of Dunn fail to provide the missing elements of independent claims 36, 46 and 52. Even if Lovell and Dunn could be properly combined (a point Applicants in no way concede), Dunn's purported disclosure of open ended connectors and their associated function icons does little to remedy the lack of disclosure pertaining to identifying an insufficient-defined connector and receiving information configured to complete the insufficient-defined connector.

For example, the cited sections of Dunn purportedly disclose a methodology designer with open ended connectors. *See* Dunn, Col. 17, ll. 35 – Col. 18, ll. 35.

According to the Office Action, Dunn's open ended connectors are interpreted as being somehow equivalent to the claimed insufficiently-defined connector. *See* Office Action, Pgs. 4-5. Applicants respectfully disagree. Dunn's system identifies open ended connectors and prompts the methodology designer to assign input or output values to the open ended connectors. *See* Dunn, Col. 4, l. 66 – Col. 5, l. 4. The assigned input and output values merely establish a transfer function across a function icon. *See id.*, Col. 5, ll. 4-6. Further, Dunn's methodology designer is required to indicate whether each open ended connector is an input, an output, or bi-directional and the connector line types applicable to each connection point. *See id.*, Col. 5, ll. 15-23. Even ignoring the foregoing deficiencies, Dunn provides nothing in the way of disclosure that addresses Lovell's mandatory, immediate synchronization of the constructs described therein, and so fails to remedy at least this fatal flaw, whereby Lovell fails to possess the ability to support the claimed identifying/flagging functionality as a result of all Lovell's constructs being in one state (completed) or the other (incomplete), thereby rendering the claimed identifying/flagging meaningless in Lovell, Dunn, <u>and</u> their combination.

Given the foregoing, it is clear the Dunn's open ended connectors function as pathways and/or tracks between connection points (e.g., they require connector line styles) and therefore are not remotely equivalent to the claimed insufficiently-defined connector which can be identified by the claimed non-technical user and completed by the claimed technical user. Moreover, because Dunn's open ended connector is limited to an input, output or bi-directional role, Dunn is not cognizant of any reason of a methodology that permits completion of an insufficiently-claimed connector (e.g., a connector between processes and sub-processes, not between icons) by a technical user. Therefore, Dunn, either alone or in combination with Lovell, would still not remedy the above mentioned infirmities and therefore, cannot be properly combined with Lovell to disclose the elements of independent claims 36, 46 and 52.

For at least these reasons, Applicants respectfully submit that independent claims 36, 46 and 52 are patentably distinguishable over the cited sections of Lovell and Dunn. Applicants therefore respectfully submit that independent claims 36, 46 and 52 and all claims depending therefrom are in condition for allowances. Applicants therefore respectfully request the Examiner's reconsideration and withdrawal of the rejection to these claims and an indication of the allowability of same.

Independent Claim 52

The Office Action rejects claims 36 and 52 under the same rationale. *See* Office Action, Pg. 6. It should be noted that claim 52 claims a transfer and flagging logic, a limitation that is not recited in claim 36. Therefore, claim 52 cannot be rejected under the same rational used to reject claim 36. In any case, Applicants respectfully submit that claim 52 is patentably distinguishable over the cited references, at least for the same reasons as discussed above.

Claims 40-42, 45, 48-50, 51, 54, and 55

Claims 40-42, 48-50 and 54 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lovell in view of Dunn as applied to claim 36 above, and in further view of Iborra et al., U.S. Patent No. 7,137,100 ("Iborra"). Claims 45, 51 and 55 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lovell in view of Dunn as applied to claim 36 above, and in further view of Nwabueze, U.S. Patent Publication No. 2002/0144174 ("Nwabueze"). Applicants traverse these rejections. For at least the reasons that claims 40-42, 45, 48-50, 51, 54, and 55 are dependent upon independent claims 36, 46 and 52, Applicants respectfully submit that the rejections of these claims under 35 U.S.C. § 103(a) is overcome.

CONCLUSION

In view of the amendments and remarks set forth herein, the application and the claims therein are believed to be in condition for allowance without any further examination and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is invited to telephone the undersigned.

If any extensions of time under 37 C.F.R. § 1.136(a) are required in order for this submission to be considered timely, Applicant hereby petitions for such extensions. Applicant also hereby authorizes that any fees due for such extensions or any other fee associated with this submission, as specified in 37 C.F.R. § 1.16 or § 1.17, be charged to Deposit Account 502306.

Respectfully submitted,

/ Samuel G. Campbell III /

Samuel G. Campbell III
Attorney for Applicants
Reg. No. 42,381
Telephone: (512) 439-5084
Facsimile: (512) 439-5099