Amendments to the Claims

This listing of claims, if entered, will replace all prior versions and listings of claims in the above-identified application.

1-35. (Cancelled)

36. (**Currently Amended**) A method comprising:

modeling a business process, wherein said modeling comprises

designing a process, wherein

said process represents a non-technical model of said business process,

said designing is performed using a non-technical user interface, and

said designing further comprises

identifying an element, wherein

said element is an insufficiently-defined element,

identifying a connector, wherein

said connector is an insufficiently-defined connector, and

receiving information, wherein

said information is configured to allow said insufficiently-defined element and said insufficiently-defined connector to be completed,

in response to an indication that said process is complete, transferring access to said process from said non-technical user interface to a technical user interface, and

implementing said process, wherein

said implementing implements said process as a technical model of said business process, and

said implementing is performed using **[[a]]** **said** technical user interface.

37. (Previously Presented) The method of claim 36, wherein said modeling further comprises:

editing said process, wherein

said editing is performed using said non-technical user interface.

38. (Previously Presented) The method of claim 37, wherein

said process is an existing process.

39. (Previously Presented) The method of claim 38, wherein said modeling further comprises:

selecting said existing process from a plurality of existing processes, wherein

a business process library comprises said existing processes.

40. (Previously Presented) The method of claim 36, wherein said transferring comprises:

analyzing said process, wherein

said analyzing is performed in response to said process being checked-in from said non-technical user interface;

if a step is missing from said process, identifying said step as a missing step;

if said process comprises a new element, identifying said new element; and

if said process comprises a new connection, identifying said new connection.

41. (Previously Presented) The method of claim 40, wherein said transferring further comprises:

if said missing step is identified, flagging said missing step;

if said new element is identified, flagging said new element; and

if said new connection is identified, flagging said new connection.

42. (Previously Presented) The method of claim 36, wherein said implementing comprises:

displaying a systems view of said process;

determining whether an element needs to be added to said process; and if said element needs to be added to said process, indicating said element needs to be added to said process.

43. (**Currently Amended**) The method of claim 36, wherein said implementing comprises:

identifying an **additional** element, wherein
said **additional** element is a non-executable element;
identifying a**n additional** connector, wherein
said **additional** connector is a non-executable connector; and
receiving **additional** information, wherein
said **additional** information is configured to allow said non-executable element and said non-executable connector to be completed.

44. (Previously Presented) The method of claim 36, wherein said modeling further comprises:

integrating said process with an external service.

45. (Previously Presented) The method of claim 44, wherein said integrating comprises:

defining a source, wherein
said source defines a location of said external service, and
said source further defines an access mode for said external service;
defining a format, wherein
said format defines a first format for addressing said external service, and
said format further defines a second format to be used to transfer data from said external service; and
defining a transform, wherein
said transform defines a transformation between said first format and said second format.

46. (**Currently Amended**) A **non-transitory** computer program product comprising:
a plurality of instructions, comprising

a first set of instructions, executable on a computer system, configured to model a business process, wherein

said first set of instructions comprise

a first subset of instructions, executable on said computer system, configured to design a process, wherein

said process represents a non-technical model of said business process,

said designing is performed using a non-technical user interface, and

said first subset of instructions comprise

instructions, executable on said computer system, configured to identify an element, wherein

said element is an insufficiently-defined element,

instructions, executable on said computer system, configured to identify a connector, wherein

said connector is an insufficiently-defined connector, and

instructions, executable on said computer system, configured to receive information, wherein

said information is configured to allow said insufficiently-defined element and said insufficiently-defined connector to be completed,

a second subset of instructions, executable on said computer system, configured to transfer access to said process from said non-technical user interface to a technical user interface, in response to an indication that said process is complete, and a third subset of instructions, executable on said computer system, configured to implement said process, wherein

said third set of instructions is configured to implement said process as a technical model of said business process, and

said implementing is performed using **[[a]]** **said** technical user interface, and

a computer readable storage medium, wherein said instructions are encoded in said computer readable storage medium.

47. (**Currently Amended**) The **non-transitory** computer program product of claim 46, wherein said further first set of instructions comprises:

a fourth subset of instructions, executable on said computer system, configured to edit said process, wherein

said editing is performed using said non-technical user interface, and

said process is an existing process; and

a fifth subset of instructions, executable on said computer system, configured to select said existing process from a plurality of existing processes, wherein

a business process library comprises said existing processes.

48. (**Currently Amended**) The **non-transitory** computer program product of claim 46, wherein said second subset of instructions comprises:

a first sub-subset of instructions, executable on said computer system, configured to analyzing said process, wherein

said first sub-subset of instructions are executed in response to said process being checked-in from said non-technical user interface;

a second sub-subset of instructions, executable on said computer system, configured to identify a step as a missing step, if said step is missing from said process;

a third sub-subset of instructions, executable on said computer system, configured to identify a new element, if said process comprises said new element; and a fourth sub-subset of instructions, executable on said computer system, configured to identify a new connection, if said process comprises a new connection.

49. (**Currently Amended**) The **non-transitory** computer program product of claim 48, wherein said second subset of instructions further comprises:

a fifth sub-subset of instructions, executable on said computer system, configured to flag said missing step, if said missing step is identified;

a sixth sub-subset of instructions, executable on said computer system, configured to flag said new element, if said new element is identified; and

a seventh sub-subset of instructions, executable on said computer system, configured to flag said new connection, if said new connection is identified.

50. (**Currently Amended**) The **non-transitory** computer program product of claim 46, wherein said first set of instructions further comprises:

a fourth subset of instructions, executable on said computer system, configured to display a systems view of said process;

a fifth subset of instructions, executable on said computer system, configured to determine whether an element needs to be added to said process; and

a sixth subset of instructions, executable on said computer system, configured to indicate said element needs to be added to said process, if said element needs to be added to said process.

51. (**Currently Amended**) The **non-transitory** computer program product of claim 46, wherein said first set of instructions further comprises:

a fourth subset of instructions, executable on said computer system, configured to integrate said process with an external service, wherein

said fourth subset of instructions comprises

a first sub-subset of instructions, executable on said computer system, configured to define a source, wherein

said source defines a location of said external service, and

said source further defines an access mode for said external service, a second sub-subset of instructions, executable on said computer system, configured to define a format, wherein

said format defines a first format for addressing said external service, and

said format further defines a second format to be used to transfer data from said external service, and

a third sub-subset of instructions, executable on said computer system, configured to define a transform, wherein

said transform defines a transformation between said first format and said second format.

52. (Previously Presented) A computing system comprising:

a processor; and

a computer-readable storage medium, wherein

said computer-readable storage medium and said processor are coupled to one another,

said computer-readable storage medium has instructions encoded therein, and

said instructions are configured to cause said processor to perform modeling of a business process by virtue of said instructions comprising

a non-technical interface module, wherein

said non-technical interface module is configured to be employed in designing a process, wherein

said designing further comprises

identifying an element, wherein

said element is an insufficiently-defined element,

identifying a connector, wherein

said connector is an insufficiently-defined connector, and receiving information, wherein

said information is configured to allow said insufficiently-defined element and said insufficiently-defined connector to be completed,

said process represents a non-technical model of said business process, and

said non-technical interface module is configured to generate a non-technical user interface,

transfer and flagging logic, wherein

said transfer and flagging logic and said non-technical interface module are coupled to one another, and

said transfer and flagging logic is configured to transfer access to said process from said non-technical user interface to a technical user interface, in response to an indication that said process is complete, and

a technical interface module, wherein

said technical interface module and said transfer and flagging logic are coupled to one another,

said technical interface module is configured to implement said process as a technical model of said business process, and

said technical interface module is configured to generate **[[a]]** **<u>said</u>** technical user interface.

53. (Previously Presented) The apparatus of claim 52, wherein said instructions further comprise:

check-in logic, wherein

said check-in logic and said non-technical interface module are coupled to one another, and

said check-in logic is configured to generate said indication, upon said process being complete.

54. (Previously Presented) The apparatus of claim 53, wherein said instructions further comprise:

validation logic, wherein

said validation logic is coupled to said check-in logic, said transfer and flagging logic, and said non-technical interface module, and

said validation logic is configured to analyze said process by virtue of being configured to determine

whether a step is missing from said process,

whether said process comprises a new element, and

whether said process comprises a new connection.

55. (Previously Presented) The apparatus of claim 52, wherein said instructions further comprise:

integration logic, wherein

said integration logic and said technical interface module are coupled to one another, and

said integration logic is configured to integrate said process with an external service by virtual of being configured to

define a source, wherein

said source defines a location of said external service, and

said source further defines an access mode for said external service,

define a format, wherein

said format defines a first format for addressing said external service, and

said format further defines a second format to be used to transfer data from said external service, and

define a transform, wherein

said transform defines a transformation between said first format and said second format.

56. (Previously Presented) The method of Claim 36, further comprising:
drilling down to a sub-process and to steps of the sub-process wherein
said drilling down enables displaying and editing the steps of the sub-process.