(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 September 2011 (29.09.2011)

PCT

(10) International Publication Number
WO 2011/117010 A1

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/EP2011/051622

(22) International Filing Date:
4 February 2011 (04.02.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
309/KOL/2010 24 March 2010 (24.03.2010) IN

(71) Applicant *(for all designated States except US)*: **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **PC, Anitha** [IN/IN]; 286,Pranava, 7th main, 9th cross, NGEF Layout, Nagarbhavi, Bangalore 560072 (IN).

(74) Common Representative: **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

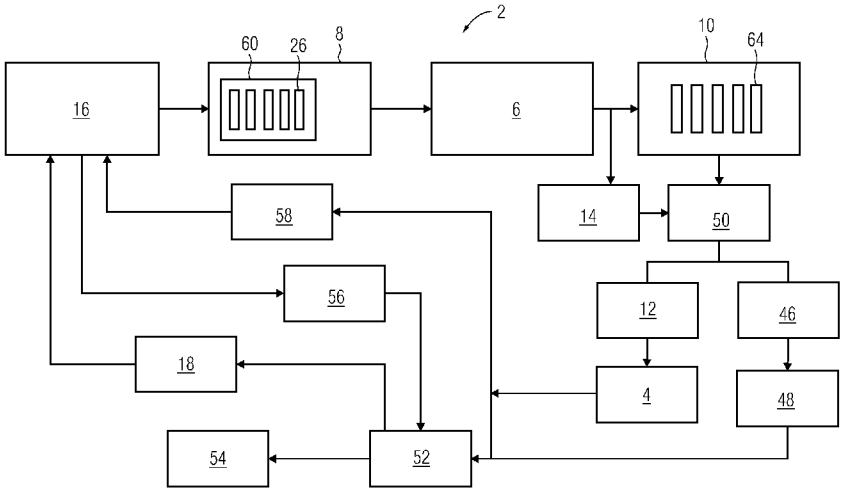
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: A SYSTEM TO GENERATE A PIECE OF CODE

FIG 1



(57) **Abstract**: A system (2) to generate a piece of code (4) includes a model transformation processor (6) adapted to process a domain specific model (8) to generate a unified model (10) therefrom and a code generator (12) adapted to generate said piece of code (4) based on said unified model (10). The system also includes a test case generator (46) adapted to generate a set of test cases (48) based on the said unified model (10).

WO 2011/117010 A1

A system to generate a piece of code

The invention relates to a system for generation of a piece of code. More specifically, the invention relates to generation of a piece of code from a domain specific model.

A domain specific model (DSM) involves a systematic way of graphic visualization of the various distinctive features for a problem specified for a specific domain like healthcare, automotives, energy etc. DSMs are considered as fourth generation languages that represent higher level abstractions of said problem, unlike general purpose modeling using unified modeling language (UML). A domain specific model carries implicit semantics specific to a domain, so that said domain specific model requires less effort and very few low-level details to specify said problem.

The domain specific model referring to said problem is converted into a piece of code which is understandable by a computing apparatus. Currently, said piece of code can be generated by using a code generator which uses unified model of said problem as input. In such a case an expert from a particular domain needs to write said problem in a unified model.

As modeling using unified modeling language requires specific skills to understand and model the requirements of said problem, so another possible way is to use a code generator which takes domain specific model from a particular domain as input. But, in such a case, the number of code generators is multiplied, as each domain would be requiring a separate code generator which can understand its need. But effectively, domain specific models for similar problem in a particular domain may result into said piece of code, after said domain specific models are used as input into a code generator for different domains.

It is an objective of the invention to generate a piece of code from a domain specific model using a code generator independent to a particular domain.

The object of the invention is achieved from a system according to claim 1 and a method according to claim 15.

The underlying idea of the invention is to process a domain specific model to generate a unified model, and said unified model is used further as an input to a code generator to generate a piece of code, so that, same code generator can be used to generate a piece of code for domain specific model from different domains.

According to one embodiment, a model transformation processor adapted to process said domain specific model to generate an extended data and said code generator generate said piece of code based on said extended data, so that, said piece of code contains clarity about features represented in said domain specific model.

According to another embodiment, said model transformation processor adapted to process said domain specific model from a plurality of domain specific editors. Such a model transformation processor provides a universal processor which transforms domain specific models independent of domains.

According to an exemplary embodiment, said domain specific model is represented in a universal language, so that it is easy to process and execute the domain specific model further by the model transformation processor.

According to one embodiment, said universal language is Extended Markup Language Data Interchange (XMI), as XMI is a standard language for defining models whether a domain specific model or a unified model, hence it is easy to

maintain functioning of said system independent of said domain specific model or unified model.

According to yet another embodiment, said model transformation processor adapted to process said domain specific model based on a rule set and adapted to produce an extension to said domain specific model based on processing of said domain specific model. The extension to said domain specific model contains information related to mapping of a domain specific element to an unified modeling element, hence converting domain specific model into a more understandable format.

According one embodiment, said model transformation processor is adapted to process a part of said extension to said domain specific model by finding a pattern of said domain specific element, matching said pattern with a set of existing unified model elements and adapted to produce a part of said unified model for said domain specific model by replacing said pattern with said unified model element, wherein said set of existing unified model elements represents a part of existing unified model from a unified model database. Such model transformer uses existing unified model thus providing reusability of previously existing unified model.

According to an exemplary embodiment, said model transformation processor is adapted to determine said set of existing unified model elements to be mapped on said pattern by analyzing said extension to said domain specific model and said existing unified model stored in said unified model database. Such a transformation processor increases accuracy of processing of said domain specific model.

According to one embodiment, said model transformation processor is adapted to validate said unified model with a document type definition of said unified model from a document type definition database, hence providing a unified

model having semantics proper in respect to document type definition.

According to another embodiment, said model transformation processor comprises an interface to receive a user data, so that said user data determines parameter at least for processing said set of existing unified elements or for processing said extension to said domain specific model or for matching said pattern with said set of existing unified model elements or for producing a part of said unified model for said domain specific model. This provides a way for a domain expert to intervene for any changes or corrections during processing of said domain specific model.

According to yet another embodiment, system includes a test case generator adapted to generate a set of test cases from said unified model, hence said uniform model can be tested for alignment of specifications according to needs of said domain specific model.

According to an exemplary embodiment, system also includes a requirement tracer adapted to produce a requirement coverage graph by mapping said set of test cases to a set of requirements determined by said domain specific model; hence such requirement tracer keeps a track of coverage of all needs of said domain specific model.

According to one other embodiment, requirement tracer is further adapted to process said set of test cases to generate a set of amendments to amend said domain specific model. This helps the domain expert to amend the domain specific model to make changes in said domain specific model for generating a functionally correct piece of code.

According to another embodiment, system also includes a graphical marker adapted to highlight a part of said domain specific model being processed, so that a domain expert can

easily take a notice said part of domain specific model under processing.

FIG 1 is a schematic diagram of a system to generate a piece of code and a set of test cases from a domain specific model.

FIG 2 is a schematic diagram of functioning of a model transformation processor which is included in said system to generate said piece of code and said set of test cases.

FIG 1 illustrates a schematic diagram of a system 2 generating a piece of code 4 from a domain specific model 6. The system 2 includes a domain specific editor 16 providing a domain specific model 8, a model transformation processor 6 processing said domain specific model 8 to produce a unified model 10, a code generator 12 generating said piece of code 4 from said unified model 10, a test case generator 46 generating a set of test cases 48 from said unified model 10, a requirement tracer 52 producing a requirement coverage graph 54 and a graphical marker 58 highlighting a part 60 of said domain specific model 8 being processed.

A domain specific model 8 is written in a domain specific language following semantics and structures used in said domain and a unified model 10 is written in unified model language which is able to follow semantics and structures of all the domains in a more generalized way. Both domain specific language and unified model language are metamodel languages which use to provide information about various domain specific model elements 26 and unified model elements 64. While said domain specific model elements 26 combine together to form a domain specific model 8 and said unified model elements 64 combine together to form a unified model 10.

The domain specific editor 16 provides for writing into domain specific semantics, so that it becomes easier for any domain expert to describe problem related to certain domain

easily without going in much details of the low level modeling. The domain expert even need not have knowledge about the unified model. On a basis of problem described by said domain expert on said domain specific editor 16, said domain specific editor 16 generates a domain specific model 8 in an electronic format or any other format so that said domain specific model can be used as input into said model transformation processor 6. The domain specific editor 16 also displays said domain specific model 8 on any display device like, visual display unit or monitoring graph or any physical display, etc. There are many domain specific editors 16 which support model diagrams from different domains like automotive, medical, energy, etc. So, as per the domain specific editor 16 used, a domain specific model 8 representing a particular domain is drawn or written by the domain expert.

The domain specific model 8 is fed into said model transformation processor 6, wherein said model transformation processor 6 processes said domain specific model 8 to generate a unified model 10. Alternatively, the model transformation processor 6 need not be a specific processor rather it can be any computer processor which have generic computing facilities and specific features to understand the semantics of domain specific model 8 for purpose of processing said domain specific model 8 and to generate a unified model 10 corresponding to semantics and structure of unified modeling language.

The domain specific model 8 is represented in a universal language by the domain specific editor 16. The universal language is language of processing of said model transformation processor 6. The model transformation processor 6, hence further process said domain specific model 8 in said universal language to generate said unified model 10 in said universal language format. The universal language used by said model transformation processor 6 is an Extended Markup Language Data Interchange (XMI) which is a standard

for exchanging metadata information via extensible Markup Language (XML). In an alternate embodiment, the universal language can be any generally available language which can represent said domain specific model 8 and as well said unified model 10 and could be understood by said model transformation processor 6 and said code generator 12 for processing said domain specific model 8 and said unified model 10 respectively.

In yet another embodiment said domain specific model 8 and said unified specific model 10 is represented and processed in a distinctive language by said model transformation processor 6 and said code generator 12. In such a case, said system 2 has language transformation interface to transform the unified model 10 into the language of processing of said code generator 12.

The system 2 includes a decision maker 50 adapted to decide whether to generate said piece of code 4 from said code generator 12 or to generate said set of test cases 48 from said test case generator 46 or both. Alternatively, the decision maker 50 can be any general purpose processor or a central processor unit of a computer system enabled to decide over said unified model 10 processing through said code generator 12 or said test case generator 46. In another embodiment, said decision maker 50 may decide to generate both said piece of code 4 and said set of test cases 48. Alternatively, said decision maker 50 need not be part of said system 2 and said system 2 can generate both said piece of code 4 and said set of test cases 48 without deciding by said decision maker 50 on said processing of said unified model 10 by said code generator 12 and said test case generator 46.

The code generator 12 processes said unified model 10 to generate said piece of code 4 in said universal language, XMI. Alternatively, the code generator 12 need not process said unified model 10 in XMI, rather said code generator 12

can process said unified model 10 in any other universal language. In yet another exemplary embodiment, the code generator 12 processes said unified model 10 in any other language for which said code generator 12 is programmed to process. The code generator 12 can also be any general purpose processor or any central processing unit programmed to process said unified model 10 to generate said piece of code 4.

The test case generator 46 processes said unified model 10 and generates said set of test cases 48 in said universal language, XMI. Alternatively, the test case generator 46 processes said unified model 10 in any other universal language to produce said set of test cases 48 in said universal language. In an exemplary embodiment, the test case generator processes said unified model 10 in any other language for which said code generator 12 is programmed to process and generate said set of test cases 48 in said language. The test case generator 46 alternatively can also be programmed to process said unified model 10 in a different language in respect to language in which said test case generator 46 generates said set of test cases 48. The test case generator 46 can also be any general purpose processor or any central processing unit programmed to process said unified model 10 to generate said set of test cases 48. In one embodiment, said piece of code 4 is tested on a basis of some standard and generalized test cases available, thus said test case generator is not used.

The requirement tracer 52 produces a requirement coverage graph 54 by mapping said set of test cases 48 to a set of requirements 56 determined by said domain specific model 8. The requirement tracer 52 after analyzing said set of test cases 48 also specifies a set of amendments 18, on a basis of which changes are required to be made in said domain specific model 8. On a basis of said set of amendments, the domain expert makes further changes in said part 60 of domain specific model 8 for which said set of amendments 18 is

generated by said requirement tracer. Alternatively, the requirement tracer need not generate said set of amendments, rather on a basis of said coverage graph 54; said domain expert can make amendments in said domain specific model 8. Still alternatively, the requirement tracer 52 need not generate said requirement coverage graph, rather said domain expert can make amendments in said domain specific model on a basis of said set of amendments 18 generated by said requirement tracer 52.

The set of requirements 56 are established by said model transformation processor 6. Alternatively, said set of requirement can also be established by any other processing apparatus of said system 2, i.e., said code generator 12, said test case generator 46, etc. Requirement tracer 52 takes said set of test cases 48 as input and maps them to said set of requirements 56. Alternatively, said requirement tracer 52 can take said piece of code 4 individually or said set of test cases together 48 with said piece of code 4 to generate said requirement coverage graph 54. In yet another embodiment, said set of requirements 56 are mapped onto set of test cases 46 by any domain expert manually or by any other generally available processor enabled to map said set of requirements 56 onto said set of test cases 46. The requirement coverage graph 54 shows how many requirements are covered by said set of test cases 48. Alternatively, said requirement coverage graph 54 can determine coverage of requirements by said piece of code 4 or by said piece of code 4 together with said set of test cases 48 in case when requirement tracer 52 takes said piece of code 4 as input or said set of test cases 48 together with said piece of code 4 as input respectively.

The graphical marker 58 highlights a part 60 of said domain specific model 8 which is being processed. When said domain model 8 is under processing by said model transformation processor 6, the graphical marker 58 highlights the relevant part of said domain specific model 8, so as to support said

domain expert to identify said part 60 of domain specific model 8 under processing. This helps said domain expert to provide a user data 44 while said domain specific model 8 is under processing by said model transformation processor 6. The graphical marker 58 can also keep a record of other parameters for processing said domain specific model 8 like processing time, error tracking while processing of said domain specific model 8, an inter-relating highlighting on requirement coverage graph 54 while said domain specific model 8 is being processed, monitoring of unified model 10 or any such functions which are needed by said domain expert to monitor processing of said domain specific model 8.

Figure 2 illustrates a schematic diagram of functioning of a model transformation processor 6 which is included in said system 2 to generate said piece of code 4. The model transformation processor 6 processes said domain specific model 8 to generate a unified model 10 therefrom.

The model transformation processor 6 processes said domain specific model 10 based on a rule set 20 and produces an extension 22 to said domain specific model 8. The rule set 20 is having various rules according to particular domain which helps to generalize said domain specific model 8. The extension 22 to said domain specific model 8 is a generalized version of said domain specific model 8 which contains a mapping information 24 about said domain specific model 8 to correlate each of domain specific element 26 to each of unified model elements 64 in said unified model 10.

The model transformation processor 6 processes a part 64 of said extension 22 to said domain specific model 8 by finding a pattern 30 of said domain specific element 26, matching said pattern 30 with a set of existing unified model elements 32 and produces a part 62 of said unified model 10 for said domain specific model 8 by replacing said pattern 30 with said set of existing unified model elements 32, wherein said set of existing unified model elements 32 represents a part

64 of existing unified model 34 from a unified model database 36.

The pattern 30 of said domain specific elements 26 is a combination of said domain specific elements 26 which are correlated to each other through certain activities, like said domain specific elements are sequential in occurrence or related to each other due to time of occurrence or frequency of occurrence or any other way which sets certain relation between said domain specific elements which are part of said pattern.

The set of existing unified model elements 32 is a group of existing unified model elements which is taken from an existing unified model 34. The existing unified model 34 is stored in a unified model database 36. The model transformation processor 6 on finding said pattern 30 from said domain specific model 8 searches said unified model database 36 for said set of existing unified model elements 32. On delivery of said set of existing domain specific elements 32 by said unified model database 36, said model transformation processor 6 matches said pattern 30 with said set of existing unified specific elements 32. The unified model database 36 stores said set of existing unified model elements 32 based on previous matches by said model transformation processor 6. Alternatively, said unified model database 36 can also store said set of existing domain specific elements 32 based on any other factors like sequential occurrence of existing unified model elements in said existing unified model 34 or co-existence of existing unified elements in more than one set of existing unified model elements 32 or various other factors which can determine or establish some correlations between said existing unified elements in said set of existing unified elements 32. The factors determining said set of existing unified elements 32 is fed into said unified model database 36. On a basis of these factors, the unified model database 36 stores various sets of existing unified model elements 32

and provide said sets of existing unified model elements 32 to said model transformation processor 6. If said model transformation processor 6 finds a match between said set of existing unified model elements 32 and said pattern 30, than said set of existing unified model elements 32 is saved with a reference to said match, so that if a similar pattern re-occurs, the unified model database 36 can provide same set of existing unified model elements 32 to said model transformation processor 6.

The model transformation processor 6 searches said unified model database 36 again and again for said set of existing unified model elements 32 till a match is found for said pattern 30. The searching by said model transformation processor 6 is done randomly. In an alternate embodiment, said searching can be done on a basis of certain algorithm or method, so that to reach right match for said pattern 30 with said set of existing unified model elements 32. Alternatively, the model transformation processor 6 need not search the unified model database 36 rather it can just query said unified model database 36, while said unified model database 36 can search within itself for said set of existing unified elements 32. In yet another embodiment, the model transformation processor 6 can query another processor processing said unified model database and said processor for said set of existing unified model elements 32 from said unified model database 36 and that processor searches said unified model database 36 and provide said set of existing unified model elements 32 to said model transformation processor 6.

The model transformation processor 6 on finding match for said pattern 30 with said set of existing unified elements 32, replaces said pattern 30 with said set of existing unified elements 32 inside said extension 22 to said domain specific model 8. On such replacement, a unified model 10 is produced.

In an alternate embodiment, said extension 22 to said domain specific model 8 can have more than one pattern 30 of domain specific elements 26 and said model transformation processor 6 processes said extension 22 to said domain specific model 6 by finding plurality of said pattern 30 of domain specific elements 26, matching said plurality of patterns 30 individually with a plurality of sets of said existing unified model elements 32 and replacing each of said patterns 30 with sets of existing unified model elements 32 for whom the match was found.

The model transformation processor 6 integrates said part 62 of said unified model 10 to said extension 22 to said domain specific model 8 to generate said unified model 10. Alternatively, the model transformation processor 6 integrates a plurality of said part 62 of said unified model 10 to said extension 22 to said domain specific model 8 to generate said unified model 10. In yet another embodiment, said parts 62 of said unified model 10 are integrated together to generate said unified model 10.

In furtherance, the model transformation processor 6 validates said unified model 10 with a document type definition 38 of said unified model 10 from a document type definition database 40. The document type definition defines structures and semantics to be followed by said unified model 10. The model transformation processor 6 validates said unified model 10 in regards to the structures and semantics, so that said unified model 10 would be aligned in a format that could be understandable by any machine or processor which is meant to understand and process said unified model 10. The document type definitions 38 are stored in a document type definition database 40 and when said model transformation processor 6 validates said unified model 10, said domain data type database 40 can provide said domain type definition 38 to be matched to said unified model 10. In an alternate embodiment, the model transformation processor 6 need not validate said unified model 10 to said document type

definition 38, rather an expert in a field related to said unified model 10 or any other processor meant to validate said unified model 10 can check said unified model 10 in regards to structures and semantics. In yet another embodiment, said unified model 10 need not be validated rather said unified model is used for further application of said unified model 10 without validating the unified model 10.

The model transformation processor includes an interface 42 to receive a user data 44. The interface 42 is a human machine interface for providing input to said model transformation processor 6 like keyboard, touch based interface, graphical interface or any such interface which allows said model transformation processor 6 to receive said user data 44. The user data 44 determines parameter at least for determining said set of existing unified model elements 32 or for processing said extension 22 to said domain specific model 8 or for matching said pattern 30 with said set of existing unified model elements 32 or for producing said part 34 of said unified model 10 for said domain specific model 8 or for regulating any such functions of said model transformation processor 6.

Patent Claims

1. A system (2) to generate a piece of code (4) comprising:
- a model transformation processor (6) adapted to process a domain specific model (8) to generate a unified model (10) therefrom,
- a code generator (12) adapted to generate said piece of code (4) based on said unified model (10).

2. The system (2) according to claim 1, wherein said model transformation processor (6) is further adapted to generate an extended data (14) and said code generator (12) is further adapted to generate said piece of code (4) based on said extended data (14), wherein said extended data defines information for mapping domain specific elements to unified modeling elements and providing scope for code generator to generate said piece of code.

3. The system (2) according to any of the preceding claims, wherein said model transformation processor (6) is adapted to process said domain specific model (8) from a plurality of domain specific editors (16).

4. The system (2) according to any of the claims 1 to 3, wherein said domain specific model (8) is represented in a universal language.

5. The system (2) according to claim 4, wherein said universal language is Extended Markup Language Data Interchange (XMI).

6. The system (2) according to any of the preceding claims, wherein said model transformation processor (6) adapted to process said domain specific model (8) based on a rule set (20) and adapted to produce an extension (22) to said domain specific model (8), wherein said extension (22) to said domain specific model (8) comprises a mapping information (24) related to a domain specific element (26).

7. The system (2) according to claim 6, wherein said model transformation processor (6) is adapted to process a part (28) of said extension (22) to said domain specific model (8) by finding a pattern (30) of said domain specific element (26), matching said pattern (30) with a set of existing unified model elements (32) and adapted to produce a part of said unified model (10) for said domain specific model (8) by replacing said pattern (30) with said set of existing unified model elements (32) based on said match between said pattern (30) and said set of existing unified model elements (32), wherein said set of existing unified model elements (32) represents part of existing unified model (34) from a unified model database (36).

8. The system (2) according to claim 7, wherein said model transformation processor (6) is adapted to integrate said part of said unified model (62) to generate said unified model.

9. The system (2) according to claim 8, wherein said model transformation processor (6) is adapted to validate said unified model (10) with a document type definition (38) of said unified model (10) from a document type definition database (40).

10. The system (2) according to claim 9, wherein said model transformation processor (6) comprises an interface (42) to receive a user data (44), wherein said user data (44) determines parameter at least for determining said set of existing unified elements (32) or for processing said extension (22) to said domain specific model (8) or for matching said pattern (30) with said set of existing unified model elements (32) or for producing said part (34) of said unified model (10) for said domain specific model (8)

11. The system (2) according to any of the preceding claims, further comprising:

- a test case generator (46) adapted to generate a set of test cases (48) from said unified model (10).

12. The system (2) according to any of the preceding claims, further comprising:
- a decision maker (50) adapted to decide whether to generate said piece of code (4) from said code generator (12) or to generate said set of test cases (48) from said test case generator (46) or both.

13. The system (2) according to any of the preceding claims further comprising:
- a requirement tracer (52) adapted to produce a requirement coverage graph (54) by mapping said set of test cases (48) to a set of requirements (56) determined by said domain specific model (8).

14. The system (2) according to any of the preceding claims, wherein said requirement tracer (52) further adapted to process said set of test cases (48) to generate a set of amendments (18) to amend said domain specific model (8).

15. The system (2) according to any of the preceding claims, further comprising:
- a graphical marker (58) adapted to highlight a part (60) of said domain specific model (8) being processed.

16. A method to generate a piece of code (4) comprising:
- processing a domain specific model (8) to generate a unified model (10),
- generating said piece of code (4) based on said unified model (10).

17. A method according to claim 16, further comprising processing the domain specific model (8) to provide an extended data (14) and generating said piece of code (4) on a basis of said extended data (14), wherein said extended data (14) defines information for mapping domain specific elements

(26) to unified modeling elements (64) and providing scope for code generator (12) to generate said piece of code (4).

18. The method according to any of the claims 16 and 17, further comprising:
- processing said domain specific model (8) from a plurality of domain specific editors (16) to provide said unified model (8).

19. The method according to any of the claims 16 to 18, further comprising:
- wherein said domain specific model (8) is represented in a universal language.

20. The method according to any of the claims from 16 to 19, comprising:
- processing said domain specific model (8) based on a rule set (20),
- producing an extension (22) to said domain specific model (8) based on processing of said domain specific model (8), wherein said extension (22) to said domain specific model (8) comprises a mapping information (24) related to a domain specific element (26).

21. The method according to claim 20, further comprising:
- processing a part (28) of said extension (22) to said domain specific model (8) by finding a pattern (30) of said domain specific element (26),
- matching said pattern (30) with a set of existing unified model elements (32),
- producing a part (62) of said unified model (10) for said domain specific model (8) by replacing said pattern (30) with said set of existing unified model elements (32) based on said match between said pattern (30) and said set of existing unified model elements (32),

where in said set of existing unified model elements (32) represents a part (64) of existing unified model (34) from a unified model database.

22. The method according to claim 21, further comprising:
- integrating said part of said unified model to generate said unified model

23. The method according to claim 22, further comprising:
- validating said unified model (10) with a document type definition (38) of said unified model (10) from a document type definition database (40).

24. The method according to claim 23, further comprising:
- receiving a user data (44), wherein said user data (44) determines parameter at least for determining said set of existing unified elements (32) or for processing said extension (22) to said domain specific model (8) or for matching said pattern (30) with said set of existing unified model elements (32) or for producing said part (34) of said unified model (10) for said domain specific model (8).

25. A method according to any of the claims from 16 to 24 further comprising:
- generating a set of test cases (48) by using said unified model(10).

26. A method according to any of the claims from 16 to 25, comprises:
- deciding whether to generate said piece of code (4) from said code generator (12) or to generate said set of test cases (48) from said test case generator (46) or both.

27. The method according to claims 25 and 26, further comprising:
- producing a requirement coverage graph (54) by mapping said set of test cases (48) to a set of requirements (56) determined by said domain specific model (8).

28. The method according to any of the claims 25 to 27, further comprising:

- processing said set of test cases (48) to generate a set of amendments (18) to amend said domain specific model (8).

29. The method according to any of the claims from 16 to 28, further comprising:

- highlighting a part (60) of said domain model (8) being processed.

FIG 1

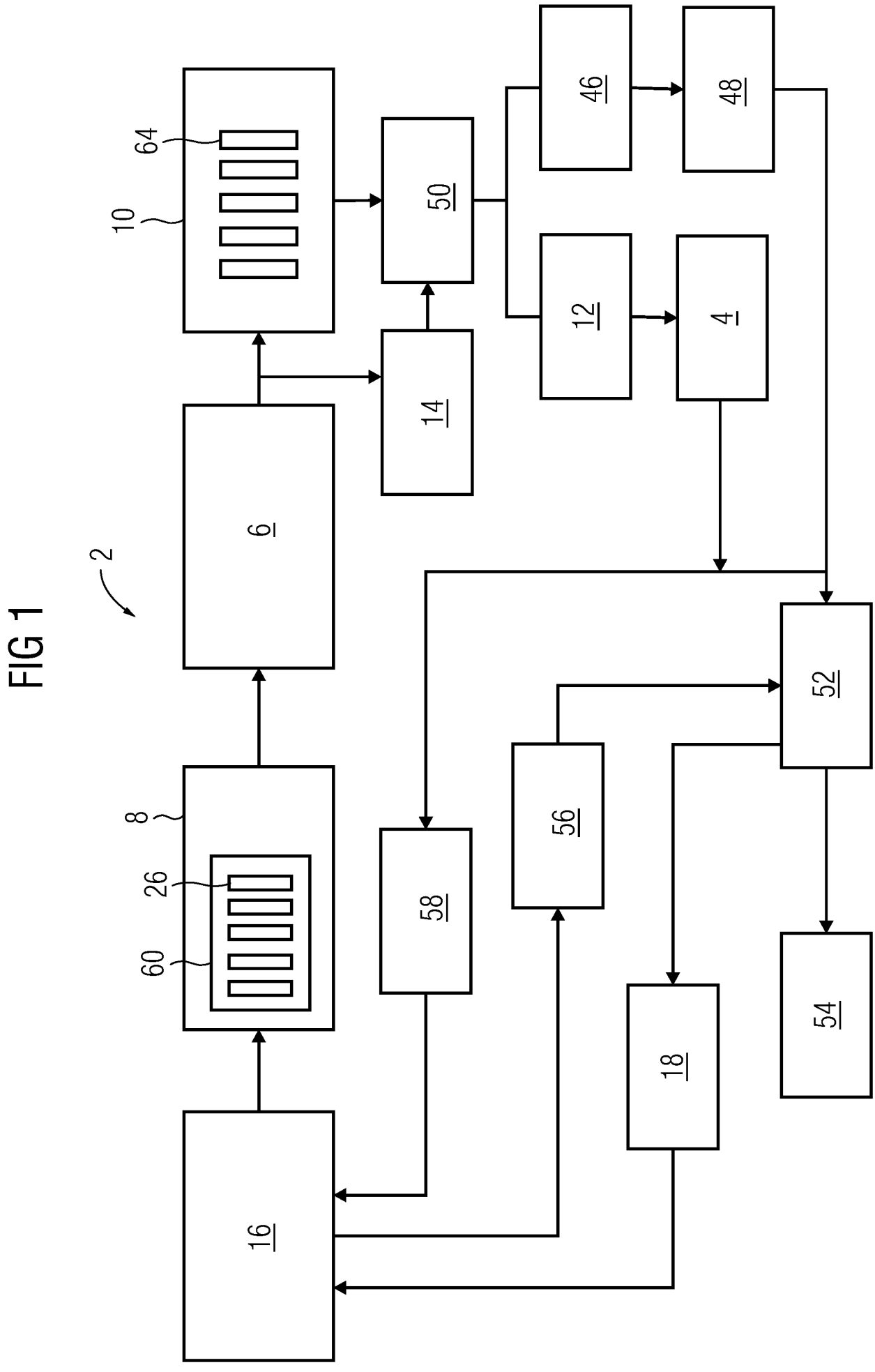
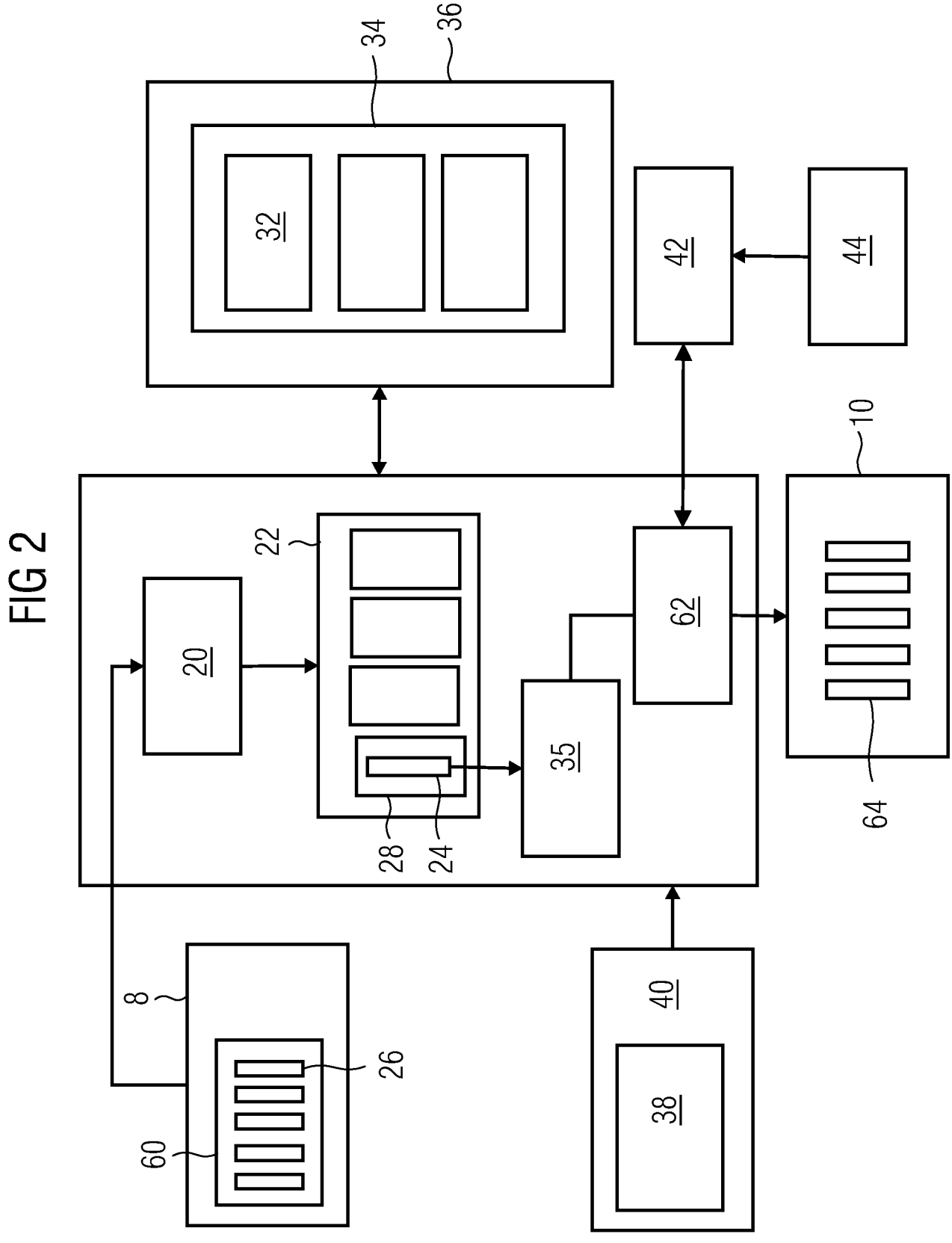


FIG 2



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/051622

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 199 195 B1 (GOODWIN RICHARD GLENN [US] ET AL) 6 March 2001 (2001-03-06) abstract column 2, line 64 - column 3, line 25 column 4, line 1 - column 5, line 10 column 5, line 36 - line 65 column 6, line 10 - column 8, line 36 column 8, line 41 - column 16, line 16 column 16, line 36 - column 17, line 50 claims 1-20; figures 1-7	1-29

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 May 2011	29/06/2011

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Eftimescu, Nicolae

1

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	VOELTER M: "A Catalog of Patterns for Program Generation", INTERNET CITATION, 14 April 2003 (2003-04-14), XP002303341, Retrieved from the Internet: URL:http://www.voelter.de/data/pub/ProgramGeneration.pdf [retrieved on 2004-10-29] the whole document page 4, paragraph 2 - paragraph 6 -----	1-29
X	BORLAND TOGETHER: "User Guide for Together ControlCenter and TogetherSolo", INTERNET CITATION, 24 March 2004 (2004-03-24), pages 1-921, XP009110320, Retrieved from the Internet: URL:http://techpubs.borland.com/together/tcc62/en/userGuide.pdf [retrieved on 2004-03-24] the whole document -----	1-29
A	KOCH T ET AL: "Model Driven Architecture", INTERNET CITATION, 28 November 2001 (2001-11-28), XP002303339, Retrieved from the Internet: URL:http://www.omg.org/cgi-bin/doc?ormsc/2002-09-04 [retrieved on 2004-10-29] the whole document -----	1-29

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2011/051622

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6199195 B1	06-03-2001	AU 5924800 A	30-01-2001
		WO 0104726 A2	18-01-2001