



(12) 发明专利

(10) 授权公告号 CN 110968308 B

(45) 授权公告日 2024. 10. 15

(21) 申请号 201910920527.9

(22) 申请日 2019.09.27

(65) 同一申请的已公布的文献号
申请公布号 CN 110968308 A

(43) 申请公布日 2020.04.07

(30) 优先权数据
102018124064.3 2018.09.28 DE

(73) 专利权人 德斯拜思有限公司
地址 德国帕德博恩

(72) 发明人 M·迈尔

(74) 专利代理机构 中国贸促会专利商标事务所
有限公司 11038
专利代理师 楼震炎

(51) Int.Cl.

G06F 8/35 (2018.01)

(56) 对比文件

CN 102262693 A, 2011.11.30

CN 104391730 A, 2015.03.04

审查员 林婉娟

权利要求书3页 说明书11页 附图4页

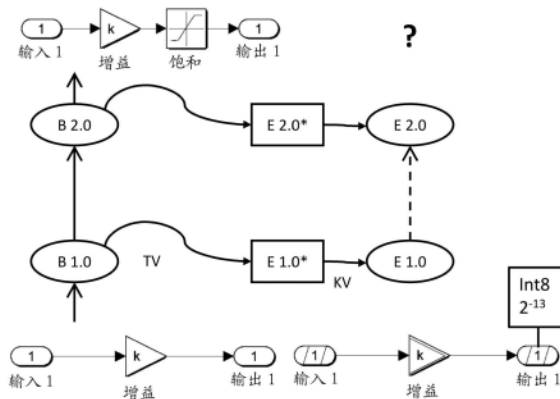
(54) 发明名称

一种用于改变用于生成源代码的模型的方法

(57) 摘要

本发明涉及一种用于在方块图之间传输修改的方法,这些方块图包括由用信号连接的块构成的周期性地被计算的模型,其中基本模型包括一个或多个基本块,其中基本块具有一个或多个基本参数,其中扩展模型包括一个或多个基本块和至少一个扩展块,其中扩展块具有至少一个附加的扩展参数。第一方块图包括第一版本的基本模型,第二方块图包括由第一版本的基本模型产生的扩展模型,而第三方块图包括第二版本的基本模型,其中第一版本比第二版本更老。按照本发明,将多个转换规则应用于第一方块图,以便获得第一中间模型;将第二方块图与第一中间模型进行比较;根据所述比较来确定至少一个配置规则;将多个转换规则应用于第三方块图,以便获得第二中间模型;而且将所述至少一个配置规则应用于第二中间模型,以便获得第四方块图。本发明还涉及一种用于配置控制设备的方法、一

种计算机程序产品和一种计算机系统。



CN 110968308 B

1. 一种用于在方块图之间传输修改的方法,所述方块图包括由用信号连接的块构成的周期性地被计算的模型,其中,基本模型包括一个或多个基本块,其中,基本块具有一个或多个基本参数,其中,扩展模型包括一个或多个基本块和至少一个扩展块,其中,扩展块具有至少一个附加的扩展参数,所述扩展参数是一个或多个用于代码生成的参数,其中,第一方块图包括第一版本的基本模型,其中,第二方块图包括由第一版本的基本模型产生的扩展模型,其中,第三方块图包括第二版本的基本模型,而且其中第一版本比第二版本更老,所述方法包括如下步骤:

- 将多个转换规则应用于第一方块图,以便获得第一中间模型,其中,转换规则包括用于标识块的准则和要应用于所识别出的块的修改,其中加入至少一个扩展块和/或删除至少一个基本块,

- 将第二方块图与第一中间模型进行比较,

- 由所述比较来确定至少一个配置规则,其中,配置规则包括用于标识块或参数的规则和要在至少一个参数上应用的修改,

- 将所述多个转换规则应用于第三方块图,以便获得第二中间模型,以及

- 将所述至少一个配置规则应用于第二中间模型,以便获得第四方块图。

2. 根据权利要求1所述的方法,

其特征在于,

确定配置规则包括:显示在第二方块图与第一中间模型之间的区别以及接收用户输入,其中,用户能编辑、启用或者舍弃从所述区别中得出的配置规则,而且其中所编辑的或者所启用的配置规则被存储。

3. 根据权利要求1或2所述的方法,

其特征在于,

转换规则和/或配置规则保存在数据库中,而且向用户提供操作界面,在所述操作界面内,能定义和/或编辑转换规则和/或配置规则,接着新的或经修改的规则被存储在数据库中。

4. 根据权利要求3所述的方法,

其特征在于,

将两个或更多个被存储的配置规则在共同点上进行比较,以便获得通用配置规则;而且在操作界面内显示通用配置规则,所述操作界面能够实现对所述通用配置规则的编辑、启用或者舍弃,其中在编辑或启用的情况下,所述通用配置规则被存储。

5. 根据权利要求1或2所述的方法,

其特征在于,

分级地定义方块图,其中,一个在更高层中的块能包括多个下层的块,将下层的块分配给更高层的块;而且对块的标识依据说明了功能性的块类型和/或依据在分级模型中描述块的位置的模型路径和/或依据块名称来实现。

6. 根据权利要求1或2所述的方法,

其特征在于,

在应用配置规则之后,检查至少一个合理性条件,其中,在不满足的情况下输出错误报告和/或显示用于修改参数的图形用户界面。

7. 根据权利要求1或2所述的方法，其特征在在于，

分级地定义方块图，其中，一个在更高层中的块能包括多个下层的块，其中，将下层的块分配给更高层的块，而且在两个方块图之间进行比较之前，将一个方块图的块与另一个方块图的块等同。

8. 根据权利要求1或2所述的方法，其特征在在于，

对于将块加入到方块图中的规则而言，所加入的块与至少一个到目前为止存在的块通过信号来连接，按照附加的规则，依据在信号流中相邻的块的一个或多个参数值来选择所加入的块的至少一个参数的值。

9. 根据权利要求1或2所述的方法，其特征在在于，

能够将来自库中的事先定义的提供功能性的块加入到方块图中，其中，所定义的具有相同功能性的块能多次存在于所述方块图中；而且至少一个预先定义的块分配有一个或多个转换规则。

10. 根据权利要求1或2所述的方法，其特征在在于，

规则包括：如果在被修改的方块图中，缺少了原来方块图的在信号流中依次布置的三个块中的中间块或者所述中间块被删除了，所述中间块对输出信号所具有的缩放不同于处在信号流上游的块，而且处在信号流下游的块按照设定应该继承或采用所述缩放，则自动地加入缩放块。

11. 根据权利要求1或2所述的方法，其特征在在于，由所述第四方块图生成源代码。

12. 根据权利要求1或2所述的方法，其特征在在于，

配置规则包括用于标识块或参数的规则和要在扩展参数上应用的修改。

13. 根据权利要求6所述的方法，其特征在在于，

在不满足的情况下输出错误报告和/或显示用于修改扩展参数的图形用户界面。

14. 根据权利要求8所述的方法，其特征在在于，

相邻的块包括紧挨着的前块和/或紧挨着的后块。

15. 一种用于在方块图之间传输修改的方法，所述方块图包括由用信号连接的块构成的周期性地被计算的模型，其中，基本模型包括一个或多个基本块，其中，基本块具有一个或多个基本参数，其中，扩展模型包括一个或多个基本块和至少一个扩展块，其中，扩展块具有至少一个附加的扩展参数，所述扩展参数是一个或多个用于代码生成的参数，其中，第一方块图包括第一版本的扩展模型，第二方块图包括第一版本的基本模型，第三方块图包括第二版本的扩展模型，而且其中第一版本比第二版本更老，所述方法包括如下步骤：

· 将至少一个反转的转换规则应用于第三方块图，以便获得第一中间模型，其中，转换规则包括用于标识块的准则和要应用于所识别出的块的修改，其中，按照转换规则加入至

少一个扩展块和/或删除至少一个基本块,其中,反转的转换规则使对所分配的转换规则的修改撤销,

- 将第一中间模型与第二方块图进行比较,以便确定被修改的基本块和/或基本参数,
- 将所确定的修改传输到第一方块图,以便获得第二中间模型,以及
- 将至少一个转换规则应用于第二中间模型,以便获得第四方块图。

16.一种用于配置控制设备的方法,其中,所述控制设备包括至少一个计算单元,所述方法包括如下步骤:

- a. 利用根据权利要求1至15之一所述的方法来创建第四方块图,
- b. 由所述第四方块图生成源代码,
- c. 针对所述计算单元来编译所述源代码,使得生成可执行的代码,
- d. 将所述可执行的代码传输到所述控制设备,以及
- e. 将所述可执行的代码保存在所述控制设备的非易失性存储器上和/或通过所述控制设备的计算单元来执行所述可执行的代码。

17.根据权利要求16所述的方法,其特征在于,所述控制设备包括至少一个传感器和/或至少一个执行器,以便检测物理过程的数据和/或对所述物理过程施加影响。

18.一种计算机程序产品,所述计算机程序产品具有计算机可读存储介质,在所述计算机可读存储介质上置入指令,当所述指令由处理器来执行时,所述指令引起:所述处理器被设立为执行根据权利要求1至17之一所述的方法。

19.一种计算机系统,所述计算机系统包括人机接口、非易失性存储器和处理器,其中,所述处理器被设立为执行根据权利要求1至17之一所述的方法。

一种用于改变用于生成源代码的模型的方法

技术领域

[0001] 本发明涉及一种对可执行的模型、尤其是方块图的改变以及由方块图来生成用于控制设备的编程的源代码。

背景技术

[0002] 在多种应用中使用控制设备,以便检测过程的物理量和/或借助于所连接的执行器对过程起作用;例如可以涉及制动过程的防抱死控制。确定过程的动态行为的时间常数通常引起1ms或者更短的周期时间,使得需要控制设备的实时能力。出于成本原因,控制设备通常具有微控制器,该微控制器具有微小的存储器和有限的计算能力,因此可执行的代码的大小和效率非常重要。

[0003] 为了加快对控制设备的设计,经常依据在像MATLAB/Simulink那样的计算环境下的模型来开发控制策略。因此,过程和/或调节器或通常控制设备的行为首先可以被仿真,并且可以检查所希望的特性是否存在。这些模型尤其可以是方块图,这些方块图包括执行如计算那样的运算块,其中,块例如可以由多个输入信号来计算一个输出信号。通常,方块图周期性地被执行,其中,所有块持久地保持在存储器中而且每个块每个时间步长被执行一次。尤其是,块可以将一个或多个运算应用于来自上一步骤的输入信号,以便产生当前步骤的输出信号。除了周期性地执行的子模型之外,为了描述近似时间连续的行为,方块图也可以包括用于描述离散行为的子模型,在该子模型中定义了多个状态和过渡条件。

[0004] 借助于代码生成器,由这些模型可以直接生成用于控制设备的编程的源代码。例如,从H.Hanselmann等人的文献“Production Quality Code Generation from Simulink Block Diagrams”,1999年计算机辅助控制系统设计国际研讨会论文集,柯哈拉海岸,夏威夷,公知一种用于生成在产品质量方面的源代码的代码生成器。在这种情况下,所生成的产品代码可以通过对块或块变量的附加的注释来适配。尤其可以给最初为了利用高精度的浮点变量来进行计算而创建的模型配备设置缩放(Skalierung),由此控制设备可以以定点数来进行计算,以便在精度方面在微小的损失的情况下实现计算耗费的显著降低。

[0005] 在边缘计算领域(在该边缘计算领域内,出现的大的数据量必须高效地被分析),数据流导向的编程语言在此期间也投入使用。这样从US2017/0277521A1公知一种方法,其中,在开发环境下标识多个数据流,一系列功能应该被应用于所述多个数据流。输入数据流作为具有多个元素的队列而存在,这些元素例如应该由“合并”功能组合成共同的输出数据流。所设计的程序在二级翻译工具中被翻译成经优化的平台特定的指令。在这种情况下,被应用于各个数据流的转换功能以及“合并”功能被翻译并且组合成确定性有限自动机。通过该优化,不必针对“合并”功能的中间级提供缓冲,使得能够实现数据流的高效的拼合。对程序的改变可以在包含可视化组件的图形用户界面内进行,用于分布式开发的辅助工具在该文献中未公开。

[0006] 在较大的开发项目中,开发的不同区域通常由多个人员来执行,这些人员也可以在不同的企业任职。创建和改进模型的功能开发者通常可以以浮点数来进行工作。曾经创

建的功能模型也可以被用于不同的项目并且在功能性方面予以适配。对于移植到控制设备而言所需的附加注释通常通过代码生成的专门人员来实现,该模型在相应的项目阶段被移交给该专门人员。对于该步骤而言,通常只有几天时间供支配,在这几天时间里,必须进行大量手动修改。如果项目包括多个阶段,在这些阶段中分别应该进行代码生成,则问题变得尖锐;即使作为基础的模型基本上保持不变,附加注释也经常是手动工作。

[0007] 对于以分别具有大量单词的行来排列的文本、如高级语言的源代码,存在比较程序,这些比较程序也经常能够实现三路比较。尤其是当文本的多个彼此不同的版本是基于初始版本来创建的、例如由两个在源代码方面并行地工作的编程来创建的时,该三路比较是有帮助的。方块图或一般图形模型的(关于文本方面)较复杂的结构使在模型之间的比较变得困难。为了改善手动可操作性,US2018/0060457 A1提出:借助于脚本来过滤在方块图之间的比较结果或对该结果进行后处理。特别是在包括大量不同的层级的分级方块图中,传输修改只是非常有限地可能的。从US 8151244 B2公知一种用于“合并”或拼合图形程序的方法,其中,考虑共同的原始程序。但是,在这方面必要的是:模型使用相同版本的开发环境而且没有不同的附加注释,使得不能将在功能模型方面的修改自动化地传输到经注释的模型以进行代码生成。

发明内容

[0008] 在该背景下,本发明的任务是进一步改进现有技术并且尤其是支持在不同的模型类型之间传输修改。

[0009] 该任务通过根据本发明所述的用于在方块图之间传输修改的方法、根据本发明所述的计算机程序产品和根据本发明所述的计算机系统来解决。

[0010] 因此,提供了一种用于在方块图之间传输修改的方法。方块图包括由用信号连接的块构成的周期性地被计算的模型,其中基本模型包括一个或多个基本块,其中基本块具有一个或多个基本参数,其中扩展模型包括一个或多个基本块和至少一个扩展块,其中扩展块具有至少一个附加的扩展参数、尤其是一个或多个用于代码生成的参数。第一方块图包括第一版本的基本模型,第二方块图包括由第一版本的基本模型产生的扩展模型,其中第三方块图包括第二版本的基本模型,其中第一版本比第二版本更老。按照本发明,该方法包括如下步骤:

[0011] • 将多个转换规则应用于第一方块图,以便获得第一中间模型,其中转换规则包括用于标识块的准则和要应用于所识别出的块的修改,其中加入至少一个扩展块和/或删除至少一个基本块,

[0012] • 将第二方块图与第一中间模型进行比较,

[0013] • 由所述比较来确定至少一个配置规则,其中配置规则包括用于标识块或参数的规则和要在至少一个参数、尤其是扩展参数上应用的修改,

[0014] • 将所述多个转换规则应用于第三方块图,以便获得第二中间模型,而且

[0015] • 将所述至少一个配置规则应用于第二中间模型,以便获得第四方块图。适宜地,第四方块图包括第二版本的扩展模型而且尤其是准备用于代码生成。

[0016] 如果两个或更多个块用信号来连接,则这在模型中适宜地用信号线来显示,而且信号或数据可以在被连接的块之间传输;在这种情况下,尤其是,第一块输出一个值或根据

定义来输出多个相关联的值,而第二块接收所述多个相关联的值并且在确定第二块的一个或多个相关联的输出值时考虑所述多个相关联的值。信号可包含标量变量和/或像数组那样的结构化的数据类型,如这例如在总线中就是如此情况。不同于信号,块的参数在模型执行或代码生成期间保持固定;这些参数尤其用于规定块或所生成的代码的特性并且因而通常只在建模期间被修改。

[0017] 本发明基于如下考虑:对于具有不同的语义层模型(其中即存在至少一个基本模型和至少一个扩展模型)的而言,直接的三路合并是没有意义的。作为替代,在具有不同的语义层、但是同一版本的模型之间的区别与不同版本的模型之间的区别分开地来考虑。在从一个语义层过渡到另一个语义层时,尤其可以通过扩展块来代替基本块;基本块也可以通过一个或多个块来代替,所述一个或多个块中有至少一个块是扩展块。在这种情况下,基本模型和扩展模型的所执行的功能性通常在很大程度上相同。在所计算出的信号值之间的一定的区别例如可能由于在扩展模型中使用定点数据类型而引起。如果用户例如已经加入了一组块,则在不同版本的模型之间可能出现显著的功能上的区别。

[0018] 对各个块和/或分割开的块组的处理适宜地被表达为转换规则,该转换规则包括用于标识块的准则和要应用于所识别出的块的修改,其中加入至少一个扩展块和/或删除至少一个基本块,按照另一构思,转换规则通常可以固定地预先给定,因为与基本块的位置无关地总是可以用同一扩展块或同一组块来代替该基本块。例如,转换规则可以表明:输出信号的基本端口通过相应的扩展端口来代替;这一般适用而且因而可以简单地被自动化。适宜地,针对基本模型的每个块,检查是否满足标识规则,而且如果是则分别应用预先给定的修改。

[0019] 附加属性或所述至少一个扩展参数尤其可以包括用于代码生成的设定,如缩放、值域、变量类型或数据类型。例如,扩展块的缩放也与相邻的扩展块、即局部环境的缩放以及必要时相邻的扩展块、即局部环境的运算有关。根据另一构思,配置规则因此可以是与上下文有关,而且因此适宜地由同一版本的两个模型的比较来确定。通过将第一或老的版本的基本模型转换成在语义层方面对应于扩展模型的中间模型,提供了适合的比较基础,以便可以自动化地承担这些与上下文有关的修改中的尽可能多的修改。用户不必做那么多的手动适配来准备新版本的模型用于代码生成。

[0020] 如果模型经历多个不同的版本,则通常也可以继续应用在过去确定的配置规则,使得随着模型版本的增加用于创建相应的扩展模型的手动耗费显著降低。因此,通过存储学习到的规则或通过学习机制,用户可以更加显著地减轻负担。

[0021] 原则上,如果例如附加的块库总是包括具有一个或多个其它参数的块,则也可能存在超过两个语义层。从用于代码生成的模型的“简单的”注释过渡到(例如用于与AUTOSAR适配的代码生成的)全面地被注释的模型也可以被视为或处理为在语义层之间的过渡。接着可以进行逐步的适配,其中转换规则优选地总是包括在第一语义层与下一个语义层之间的过渡,其中相应的扩展块具有来自该层的扩展参数的预先定义的集合中的一个或多个扩展参数。

[0022] 优选地,确定配置规则包括:显示第二方块图与第一中间模型之间的区别以及接收用户输入,其中用户可以编辑、启用或者舍弃从该区别中得出的配置规则,而且其中所编辑的或者所启用的配置规则被存储。尤其是,该区别可以在图形用户界面中显示,其中用户

启用、适配或者舍弃所确定的配置规则。因此,适宜地,依据两个方块图的比较来确定多个单个的配置规则,这些单个的配置规则分别涉及各个块或者由各个块构成的组。

[0023] 特别优选地,转换规则和/或配置规则保存在数据库中,而且向用户提供操作界面,在该操作界面内,可以定义和/或编辑转换规则和/或配置规则,接着新的或经修改的规则被存储在数据库中。转换规则可以以不同的方式来设计,例如被设计为脚本环境下的脚本,该脚本可以操控模型编辑器。也可以规定:将版本分配给转换规则,而且对将转换规则应用于具有预先给定的版本和/或预先给定的创建时间的基本或扩展模型进行限制,例如以便考虑在语义层之内的修改。

[0024] 特别优选地,将两个或更多个被存储的配置规则在共同点上进行比较,以便获得通用配置规则,而且在操作界面内显示通用配置规则,该操作界面能够实现对该通用配置规则的编辑、启用或者舍弃,其中在编辑或启用的情况下,该通用配置规则被存储。适宜地,可以给配置规则分配优先级,使得例如首先应用通用配置规则并且只用特殊的配置规则来处理未被覆盖的情况。对优先级的分配可以由用户或者自动化地实现,例如依据配置规则的通用型的程度来自动化地实现。

[0025] 优选地,分级地定义方块图,其中,一个在更高层中的块可包括多个下层的块,其中,将下层的块分配给更高层的块,而且对块的标识依据说明了功能性的块类型和/或依据在分级模型中描述块的位置的模型路径和/或依据块名称来实现。如果块包括用于规定图示的掩模(Masken),则块类型也可以依据掩模类型(如TL_CustomCode)来确定。

[0026] 优选地,在应用配置规则之后,检查至少一个合理性条件,其中在不满足的情况下输出错误报告和/或显示用于修改参数、尤其是扩展参数的图形用户界面。适宜地,被注释的扩展模型依据预先给定的规则集合以及可能存在的用户规则对合理性予以检查。在错误情况下,用户可以在图形操作界面内对参数进行适配,使得换言之自动地对关键位置进行寻址。合理性条件可以预先给定或者由用户来表达,其中不仅一般的、即总是要检查的而且特殊的、即被分配给具体的配置规则的合理性条件可以被定义或检查。

[0027] 对于分级地定义方块图的情况(其中一个在更高层中的块可包括多个下层的块,其中将下层的块分配给更高层的块)而言,有利的是,在两个方块图之间进行比较之前,将一个方块图的块与另一个方块图的块等同。即从比较程序的角度,这两个块是同一个块。该等同说明了两个模型之间的相对布局而且当例如加入了附加的层级时避免了多个容易误解的错误报告。用户可以在图形操作界面内说明该布局或所要等同的块。也可以规定:该分配以规则的形式来保存,其中在模型的比较之前应用关系。还可以规定:对于不同的层级或像子系统那样的分开的区域而言,分别说明两个块之间的固有的关系,以便针对相应的层规定相对位置。

[0028] 优选地,对于将块加入到方块图中的规则而言,其中所加入的块与至少一个到目前为止存在的块通过信号来连接,按照附加的规则,依据在信号流中相邻的块、尤其是紧挨着的前块和/或紧挨着的后块的一个或多个参数值来选择所加入的块的至少一个参数的值。适宜地,该规则是转换规则而该附加的规则是配置规则。例如,对于新加入的块而言,信号的缩放可以依据尤其是在所布置的块之上的信号流中的邻块来确定;一般也可以采用信号特性或者标签或标记。

[0029] 优选地,可以将来自库中的事先定义的提供功能性的块加入到方块图中,其中所

定义的具有的功能性的块可多次存在于该方块图中,而且至少一个被定义的块分配有一个或多个转换规则。适宜地,所述一个或多根转换规则作为附加信息被存储在块库中。转换规则也可以包括:在方块图中存在块的多个实例的情况下,针对这些实例定义共同的功能并且针对每个实例定义具有块变量的数据结构,其中视所要执行的块而定,将相应的数据结构交给该功能。

[0030] 优选地,规则包括:如果在被修改的方块图中,缺少了原来的方块图的三个在信号流中依次布置的块中的中间块或者该中间块被删除了,该中间块具有对输出信号的不同于处在信号流上游的块的缩放,而且处在信号流下游的块按照设置应该继承或采用该缩放,则自动地加入缩放块。

[0031] 优选地,该方法附加地包括从第四方块图生成源代码。

[0032] 本发明还涉及一种用于在方块图之间传输修改的方法,所述方块图包括由用信号连接的块构成的周期性地被计算的模型,其中基本模型包括一个或多个基本块,其中基本块具有一个或多个基本参数,其中扩展模型包括一个或多个基本块和至少一个扩展块,其中扩展块具有至少一个附加的扩展参数、尤其是一个或多个用于代码生成的参数,其中第一方块图包括第一版本的扩展模型,其中第二方块图包括第一版本的基本模型,其中第三方块图包括第二版本的扩展模型,而且其中第一版本比第二版本更低、也就是说更老,该方法包括如下步骤:

[0033] • 将至少一个反转的转换规则应用于第三方块图,以便获得第一中间模型,其中转换规则包括用于标识块的准则和要应用于所识别出的块的修改,其中按照转换规则加入至少一个扩展块和/或删除至少一个基本块,其中反转的转换规则使对所分配的转换规则的修改撤销,

[0034] • 将第一中间模型与第二方块图进行比较,以便确定被修改的基本块和/或基本参数,

[0035] • 将所确定的修改传输到第一方块图,以便获得第二中间模型,而且

[0036] • 将至少一个转换规则应用于第二中间模型,以便获得第四方块图。

[0037] 在这种形式下,按照本发明的方法能够将对新版本的扩展模型的修改传输到老版本的扩展模型,其中在不同的版本之间也可能发生在语义层之内的修改。在系列控制设备(**Seriensteuergerät**)中,通常使代码生成器的版本或整个工具链保持不变。这里,按照本发明的方法例如能够将后来创建的补丁应用于老的模型版本,该老的模型版本是利用老的工具链或代码生成器的老版本来创建的。由于采用经由基本模型的途径,避免了对新版本的不兼容的语言工具的重新复制。适宜地,在不同版本的转换规则之间进行区别,其中在规则中根据版本来应用或避免一种或多种语言工具。

[0038] 本发明还涉及一种用于配置控制设备的方法,其中该控制设备包括至少一个计算单元而且优选地具有至少一个传感器和/或至少一个执行器,以便检测物理过程的数据和/或对该物理过程施加影响,该方法包括如下步骤:

[0039] a. 利用按照本发明的用于传输修改的方法来创建第四方块图,

[0040] b. 由第四方块图生成源代码,

[0041] c. 针对所述计算单元来编译源代码,使得生成可执行的代码,

[0042] d. 将可执行的代码传输到控制设备,而且

[0043] e.将所述可执行的代码保存在控制设备的非易失性存储器上和/或通过控制设备的计算单元来执行该可执行的代码。

[0044] 本发明还涉及一种计算机程序产品,该计算机程序产品具有计算机可读存储介质,在该计算机可读存储介质上嵌入指令,当这些指令由处理器来执行时,这些指令引起:该处理器被设立为执行按照本发明的用于传输修改的方法。

[0045] 本发明还涉及一种计算机系统,该计算机系统包括人机接口、非易失性存储器和处理器,其中该处理器被设立为执行按照本发明的用于传输修改的方法。

附图说明

[0046] 随后,本发明参考附图进一步予以阐述。在这种情况下,给相同的部分配上相同的附图标记。所示出的实施方式非常示意性而且没有按比例。

[0047] 其中:

[0048] 图1示出了计算机系统的一个优选的实施方式;

[0049] 图2示出了优选地存在于计算机系统上的软件组件的示意图;

[0050] 图3示出了生成源代码的一般方案;

[0051] 图4示出了两个方块图之间的比较的示意图;

[0052] 图5示出了模型的不同实施方案的示意图;

[0053] 图6示出了模型的被转换的形式的示意图;

[0054] 图7示出了两个分级模型的示意图;而

[0055] 图8示出了示例性的操作界面的示意图。

具体实施方式

[0056] 图1示出了计算机系统PC的一个优选的实施方式。该计算机系统具有:处理器CPU,该处理器尤其可以实现为多核处理器;工作存储器RAM;和总线控制器BC。优选地,计算机系统PC被设计为由用户直接手动地操作,其中优选地通过显卡GPU来连接显示器DIS并且通过外设接口HMI来连接键盘KEY和鼠标MOU。原则上,计算机系统PC的人机接口也可以构造为触摸界面。该计算机系统还包括:非易失性数据存储器HDD,该非易失性数据存储器尤其可以实施为硬盘和/或固态硬盘;以及接口NET、尤其是网络接口。通过接口NET可以连接控制设备ES。原则上,一个或多个任意的接口、尤其是有线接口可存在于计算机系统PC上而且能分别用于与控制设备ES的连接。适宜地,可以使用根据以太网标准的网络接口;接口NET也可以无线地来实施,如尤其是实施为WLAN接口或者也可以实施为根据像蓝牙那样的标准的接口。

[0057] 控制设备ES可以实施为系列控制设备或者实施为用于目标平台的评估板。适宜地,该控制设备包括:接口NET,用于与计算机系统PC连接;微控制器MCR,其具有与计算机系统的处理器不同的架构;工作存储器RAM;和非易失性存储器NVM。对于本发明的实施方案而言,不需要可适宜地被用于模型的测试的控制设备ES。

[0058] 原则上,该计算机系统也可以设计为客户端-服务器系统,其中一个或多个软件组件在外部服务器上被实施。

[0059] 在图2中,示出了优选地被安装在计算机系统PC上的软件组件的方案。这些软件组

件使用操作系统OS的机制,以便例如访问非易失性存储器HDD或者经由网络接口NET来建立与外部计算机的连接。

[0060] 技术计算环境TCE能够生成模型并且从这些模型生成源代码。在建模环境MOD下,优选地可以通过图形用户界面来创建动态系统的模型。在这种情况下尤其可以涉及方块图,这些方块图包括多个块并且描述了动态系统的时间行为和/或内部状态。这些块中的至少一些块通过信号来连接,即用于交换数据的定向连接,这些数据可以是标量或者组合式的。块可以是原子式的,即在一个步骤中提供一个预先定义的功能性。如果方块图是分级的,则多个在下层的块可以描述在上层的块的结构。适宜地,原子式的块也可以包括多个在下层的块。组合式的块尤其可以是子系统;子系统可具有附加的特性,如在单独的功能方面的实现和/或通过专用信号对子系统的实施的触发。在子系统中可以布置特定的块,以便进一步详细规定子系统的特性。计算环境TCE优选地包括一个或多个库,从所述一个或多个库中可以选出块或模块用于建立模型。在脚本环境MAT下,指令可以交互地或者通过批处理文件来输入,以便执行计算或者对模型进行调整。计算环境TCE还包括仿真环境SIM,该仿真环境被设立为演绎或执行方块图,以便检查系统的时间行为。优选地,利用高精度的浮点数在计算机系统的微处理器CPU的一个或多个核上进行这些计算。

[0061] 由所创建的模型,借助于代码生成器PCG可以优选地以编程语言C来生成源代码。适宜地,在定义数据集DDT中,附加信息变成模型,尤其是变成块变量。适宜地,给这些块变量分配值域和/或缩放,以便支持利用定点指令对模型的计算。源代码的所希望的特性、例如与如MISRA那样的标准的一致性(**Konformität**)可以在定义数据集DDT中被设定或保存。适宜地,每个块变量都分配有预先给定的变量类型,而且设定一个或多个所希望的特性、诸如像变量综合那样的优化的容许性。代码生成器PCG优选地分析定义数据集DDT的设定并且在生成源代码时考虑这些设定。定义数据集DDT可具有树状结构或作为简单的文件被保存在计算机系统的存储器;可选地,可以规定:将定义数据保存在专用数据库系统中。定义数据集可具有程序接口和/或导入/导出功能。技术计算环境TCE还包括转换器CNV,该转换器将具有对于代码生成而言重要的参数的模型转换成适合于比较的中间形式。

[0062] 计算机系统PC具有编译器COM和链接器,该编译器和链接器适宜地被设立用于生成能在控制设备ES和/或计算机系统PC上执行的二进制文件。原则上,可存在多个编译器、尤其是针对不同的目标平台的交叉编译器,以便支持具有不同的处理器架构的控制设备或评估板ES。此外,还存在比较程序MOC,该比较程序接收以中间形式的两个模型而且确定并优选地显示区别。附加地,由一个或多个所确定的区别可以确定或得出配置规则。优选地,比较程序包括图形用户界面,在该图形用户界面内,用户可以看到并且紧接着编辑、确认或者舍弃所得出的配置规则。此外还可存在版本控制系统VCS,该版本控制系统将不同版本的方块图和/或所生成的中间形式存储在数据库中。

[0063] 图3示出了借助于代码生成器来生成源代码的一般方案。该方法可以完全由计算机系统PC的一个优选的实施方式的处理器来实施;但是,该方法也可以被设置用于在客户端-服务器环境下利用操作计算机和一个或多个经由网络连接的服务器来实施,其中尤其是在这些服务器上执行计算密集型步骤。

[0064] 在第一步骤S1、即转换中,所选择的由方块图BLD的一个或多个块构成的模型被转换成中间图示IR,该中间图示优选地包括一个或多个分级图形。在这种情况下尤其可以涉

及数据流图、控制流图或者树状结构。除了方块图BLD之外,适宜地,来自定义数据集DDT的附加信息也在生成中间图示时被考虑或对该中间图示有影响。这也可包括如下情况,在这些情况下,基于定义数据集DDT中的信息来生成元素或者元素的特性或对于代码生成而言重要的设定、诸如变量的数据类型从定义数据集DDT中被提取。

[0065] 在第二步骤S2、即优化中,对分级图形进行优化,以便减少所需的变量的数目和/或存储器使用、诸如堆栈占用和/或运算或处理器指令的数目和/或源代码的执行时间。该优化可包括多个中间步骤,在这些中间步骤中,在模型/方块图与源代码/程序文本之间生成其它中间图示。尤其可以规定:在每个中间步骤中,都将原来的分级图形的集合转换成经修改的分级图形的另一集合,其中,应用一个或多个优化规则。在优化期间可以应用不同的策略、如“Constant folding (常量叠算)”或者对“Dead Code (死代码)”的清除。也可以规定:在步骤S1的转换的范畴内已经应用一些优化,尤其是当这些优化可以在方块图图示上较简单地被执行时。

[0066] 在第三步骤S3、即翻译中,将由全部被执行的中间步骤得到的经优化的中间图示IR或经优化的分级图形翻译成文本编程语言的源代码PCO、如尤其是C代码。在该步骤中,也可以进行进一步的优化,尤其是使得所生成的指令是原则上被该语言所包括的指令的子集和/或所生成的控制结构是原则上被该语言所包括的控制结构的子集。这能够恰好满足所定义的规则。可选地或补充地,可以规定:生成附加信息、诸如程序行与方块图BLD的块之间的关系并且尤其是将这些附加信息以注解的形式集成到源代码中,以便改善源代码PCO的可读性和/或简化调试。

[0067] 在代码生成期间或在代码生成之后,可以将关于当前的方块图的信息或者代码生成的结果、诸如报警存储在定义数据集中。这些信息例如可以被用于:影响对所生成的源代码的编译或者针对其它工具提供元信息、诸如以ASAP2格式的校准信息或者用于根据AUTOSAR标准来生成中间层的信息。在可选的实施方式中,可以规定:从方块图中生成以硬件描述语言的代码或者可编程的硬件模块的配置。

[0068] 图4示意性地示出了在两个方块图之间的比较。在这种情况下,在第一步骤中,所考虑的第一方块图在技术计算环境TCE1的模型编辑器MOD中被打开并且用转换器CNV转换成中间形式INF1。该中间形式特别适合于比较而且包括当前的参数值,这些参数值在脚本环境MAT下被适配。脚本环境MAT可以被用于对模型编辑器MOD和转换器的自动化的操控;此外,还可以规定:借助于过滤脚本来对比较结果进行过滤、修改或者补充。紧接地或者并行地,在第二步骤中,所考虑的第二方块图在技术计算环境TCE2的模型编辑器MOD中被打开并且用转换器CNV转换成中间形式INF2,其中,脚本环境MAT可以进行操控和/或过滤。在顺序处理中,TCE1和TCE2可以表示同一技术计算环境,而中间形式INF1和INF2至少暂时被存储在文件系统或数据库中。在并行执行时,技术计算环境的两个独立的实例(Instanz) TCE1、TCE2被打开。

[0069] 两个中间形式被提供给比较程序MOC,该比较程序生成带有区别DIFF的列表。适宜地,该比较程序也包括如下组件,该组件依据在第一与第二方块图之间的区别来确定配置规则。因此,由两个方块图的比较可以(根据差别的数目)得出多个配置规则。所获得的配置规则可以被保存在数据库中,其中,优选地对是否已经存在相应的配置规则进行检查,使得不应进行重新的存储。

[0070] 在图5中,示意性地示出了如下情况,在该情况下,在基本模型上的修改应该被传输到扩展模型。尤其是,基本模型可以是Simulink模型而扩展模型可以是TargetLink模型。

[0071] 示出了老版本的基本模型B1.0、扩展模型E1.0和新版本的基本模型B 2.0。基本模型B1.0包括:输入端口In1;增益块Gain,该增益块将输入信号放大了k倍(或者也许是衰减,在 $k < 1$ 的情况下);和输出端口Out1。在扩展模型E1.0中,输入和输出端口通过具有附加参数的相应的端口来代替(通过端口中的两条对角线来表示)。增益(Gain)块也是通过相应的扩展块来代替的(通过双线条来表示)。对于输出端口而言,示例性地说明了两个扩展参数,也就是说输出变量的所希望的数据类型Int8以及所希望的缩放 2^{-13} 。新版本基本模型B 2.0包括饱和块Saturation,该饱和块将增益块的输出信号限制到预先给定的区间。对于代码生成而言,需要新版本的扩展模型E 2.0。

[0072] 在原来的基本模型B1.0与经修订的基本模型B 2.0之间的区别可以依据利用比较程序MOC对两个模型比较来确定。为了阻止大量所谓的区别的出现,一般适宜的是:这些模型在这种情况下“对齐”,即用户定义在老的基本模型B1.0中的某个块相应于在新的基本模型B 2.0中的某个块或与该块等同。这尤其是适用于具有多个层级和多个单独块的范围广泛的模型。

[0073] 扩展模型E1.0与原来的基本模型B1.0之间的区别可以被分成:一般转换、诸如通过在扩展图示中的输出端口来代替基本形式的输出端口;和特殊配置、诸如对输出端口上的确定的缩放的选择。因此,可以将转换规则TV与配置规则KV区分开。如果只将转换规则应用于基本模型,则形成中间模型E1.0*或E 2.0*,所述中间模型包含扩展块,但是由于缺乏适合的属性而不适合于代码生成。适合的属性或扩展参数通常与上下文有关而且因而随着将相应的块在模型中的放置和/或相邻的块或者这些相邻的块的参数来发生变化。不同于此,曾经定义的转换通常可以未经修改地被应用于预先给定类型的基本块。

[0074] 图6示意性地示出了在图5中概述的情况的新模型的经转换的形式。新的基本模型B 2.0的输入端口已借助于转换规则通过相应的具有附加参数的扩展端口来代替。附加参数的值可以通过如下配置规则来适配,该配置规则由在经转换的基本模型B1.0、即中间模型E1.0*中的(被扩展的)输入端口与在扩展模型E1.0中的被扩展的输入端口的比较来确定或者已由在经转换的基本模型B1.0、即中间模型E1.0*中的(被扩展的)输入端口与在扩展模型E1.0中的被扩展的输入端口的比较来确定。增益块已利用另一转换规则通过被扩展的增益块来代替。用于附加参数的值可以通过如下配置规则来适配,该配置规则由在中间模型E1.0*中的(被扩展的)增益块与在扩展模型E1.0中的被扩展的增益块的比较来确定或者已由在中间模型E1.0*中的(被扩展的)增益块与在扩展模型E1.0中的被扩展的增益块的比较确定。类似地,输出端口也可以通过一般转换规则和与位置有关的配置规则被传输到新的扩展模型E 2.0。示例性地示出了用于数据类型的值Int8和用于在输出端口上的缩放的值 2^{-13} 。

[0075] 在新的基本模型B 2.0中的饱和块已借助于转换规则通过被扩展的饱和块来代替(通过上方和下方的双线条来表示)。用于数据类型的值Dt和用于缩放的值Sk不能由扩展模型E1.0与中间模型E1.0*的比较来确定,因为在基本模型B1.0中不存在饱和块。因此,能直接应用的位置特定的配置规则不可用。

[0076] 在当前情况下(其中在前块(增益块)与后块(输出端口)之间加入新的块(饱和

块)),有利的是:从对相邻的块的考虑获得用于扩展参数的可能值;可能会谈及通用或自适应配置规则。例如,这里可以采用被扩展的输出端口的扩展参数的值。在一个可选的处理方式中,可以将前块和后块的值进行比较,其中,在一致的情况下,对于要添加的块可以采用共同的值。这一种处理方式更适宜还是另一种处理方式更适宜,对于每个单个的扩展参数而言都可能是不同的。因而,该比较程序优选地包括操作界面,在该操作界面内,用户可以在不同的可能性之间进行选择。接着,所选择的处理方式优选地被存储为新的配置规则。

[0077] 图7示出了两个要进行比较的分级模型的示意图。在左侧示出了第一方块图,该第一方块图具有第一层级E1和第二层级E2。在右侧示出了第二方块图,该第二方块图具有三个层级E1、E2和E3。

[0078] 在层级E1中,第一方块图包括一个定义块D1和四个用信号连接的块B1、B2、B4、B5。块B4是分级块,该分级块的功能性通过多个在下层层级E2中的块来定义。为了简便起见,假定:块B1、B2、B5是没有下层层级的简单的块;但是,原则上也可能存在其它分级的块,那么这些分级的块分别会在分开的下层层级中实现。

[0079] 在下层层级E2中,第一方块图包括一个输入端口P21、一个输出端口P22和三个用信号连接的块B21、B2、B23,这三个用信号连接的块执行计算运算。

[0080] 在第一层级E1中,第二方块图包括一个定义块D1和五个用信号连接的块B1、B2、B3、B4、B5。相对于第一方块图,在B2与B5之间加入了附加的块(B3)。如上文所阐述的那样,可以由前块与后块的比较来确定用于被转换的块的扩展参数的值。但是,例如对于缩放而言也可能出现:新添加的块使得后块的到目前为止的值无效。如果块B3尤其是具有大的增益的增益块,则进入到B5中的信号的到目前为止的数据类型也可能溢出。因而,适宜地,在加入在这方面有问题的块之后建议针对随后的块的“worst case (最坏情况)”缩放。具体地,例如可能会通过16位宽的整数变量来代替8位宽的整数变量。优选地,在图形用户界面内将所建议的修改呈现给用户,使得该用户可以确认、编辑或者舍弃该修改。补充地,也可以规定:在经修改的模型中呈现具有这种建议或关键位置的列表。由此,用户可以有针对性地容许跳到相应的位置并且分别选择合适的处理方式。

[0081] 为了简化,在当前情况下假定:只有第二方块图的第一层级的块B4是分级块。第二层级包括输入端口P21、块B21和输出端口P22。块B21是分级块,该分级块的实现通过在下层的第三层级E3的其它块来建模。第三层级E3包括:输入端口P31;三个块B31、B32、B33;和输出端口P32。在第一方块图与第二方块图直接比较中,在相应第一层级之下的全部块都会被呈现为区别。这可以是有利的;例如当左侧的块B4已通过语义上完全不同的块来代替时。但是,也可能容易误解的是:例如第一方块图的层E2与第二方块图的层B3一致,其中附加层级已作为用于稍后的扩展的空间被加入(如对输入信号的预先过滤)。直接比较会得到大量容易误解的区别。在这种情况下适宜的是:使左侧的块B4与右侧的块B21等同,因为接着可以在参数偏离方面来检查下层的在结构上一致的建模。用户有利地做出该判断,而且该判断例如可以作为单独的、要在比较之前执行的规则来保存。

[0082] 图8示意性地示出了示例性的操作界面,该操作界面能够实现对转换和/或配置规则的编辑。原则上,规则包括过滤器、即要检查的条件或要在一致性方面进行检查的样式和动作、即要在条件满足或样式一致时应用的修改。

[0083] 由在扩展模型与所分配的中间模型之间的比较来确定的配置规则通常首先是特

定的或与位置有关的。在分级模型的情况下,位置例如可以以块路径的形式而言明,该块路径包括上层子系统或层级。适宜地,在大量配置规则已被确定并且被保存在数据库中之后,检查多个单独的配置规则是否可以通用。附加地,用户可以预先给定优先级,其中,也可以规定对通用配置规则的基本偏好。可选地,也可以规定:在标识和保存通用配置规则之后,将被其所覆盖的单独的配置规则中的一个、多个或者所有配置规则从数据库中删除或者标记为无效。类似于转换规则,通用配置规则可包含依据块类型的过滤;但是也可以规定:确定的环境的存在、即被定义为前块和/或后块的块被用作过滤标准。

[0084] 在所示出的用户界面内,左侧示出了“Filter (过滤器)”部分,在该“Filter (过滤器)”部分内,用户可以输入过滤器。为此,设置字段“Pfad (路径)”和所分配的输入区以及带有所分配的输入区的字段“Blocktyp (块类型)”。原则上,也可以将其它字段或特性用于过滤。因此在所示出的示例中,用户可以仅仅在具体的路径或仅在具体的块类型上对过滤进行规定。此外,也可设想的是:用户像“路径包括确定的子系统”和“块类型具有确定的值”那样来表述组合式标准。

[0085] 在所示出的用户界面内,右侧示出了“动作”部分,在该“动作”部分内,用户说明了在条件满足时或在满足该条件的所有位置处所要执行的动作或所要应用的修改。为此,设置分别带有所分配的输入区的字段“Typ (类型)”、字段“Name (名称)”和字段“Klasse (等级)”。与此相应地,所要应用的修改可以包括:将块类型设置为在“Typ (类型)”下说明的值和/或将块名称设置为在“Name (名称)”下说明的值和/或将变量等级设置为在“Klasse (等级)”下说明的值。原则上,也可以应用其它的、未说明的修改或执行动作。

[0086] 在所示出的用户界面内,下方示出了“**Plausibilität** (合理性)”部分,在该“合理性”部分内,用户可以规定要在应用修改之后进行检查的条件。为此,设置分别带有所分配的输入区的字段“Parameter (参数)”、字段“Min. Wert (最小值)”和字段“Max. Wert (最大值)”。附加地(未示出),也可能的是,用户可以规定:在合理性条件不满足时,是应该显示一般的报警还是应该显示在模型中的相应的位置;或在缺乏合理性时,是否应该使所应用的修改撤销。

[0087] 按照本发明的方法或相应的系统能够使用户逐渐建立对配置规则或转换规则的集合。尤其是当应该由基本模型的多个连续的版本来分别生成代码时,适配耗费随着版本的增加而明显降低。

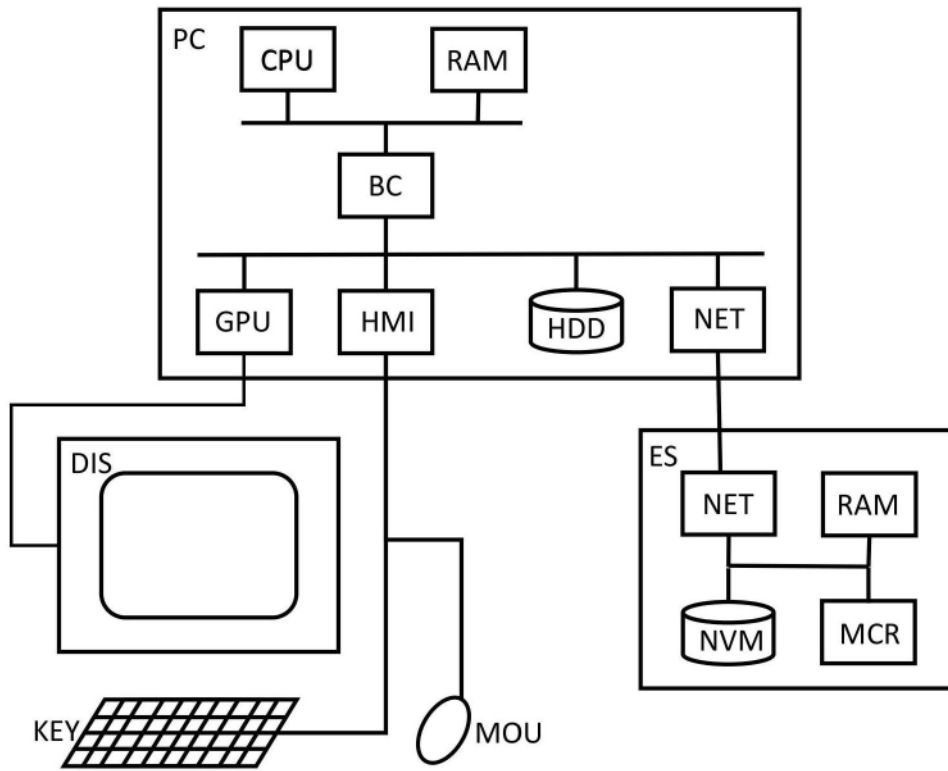


图1

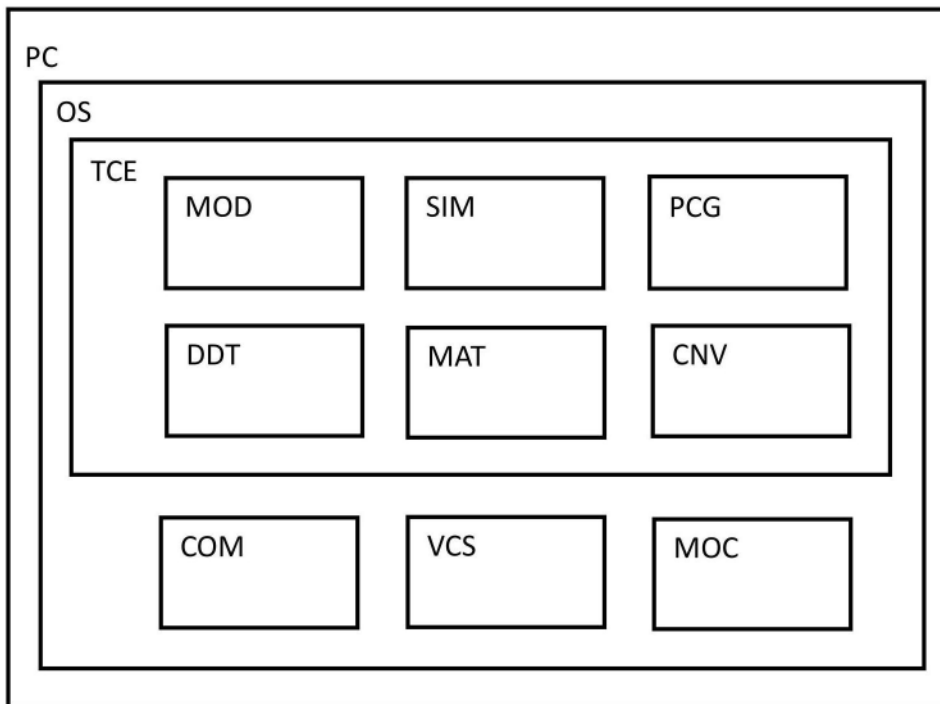


图2

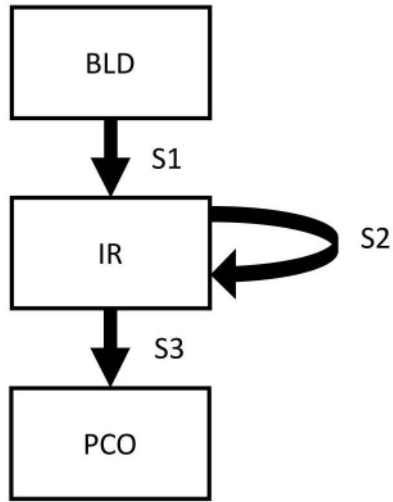


图3

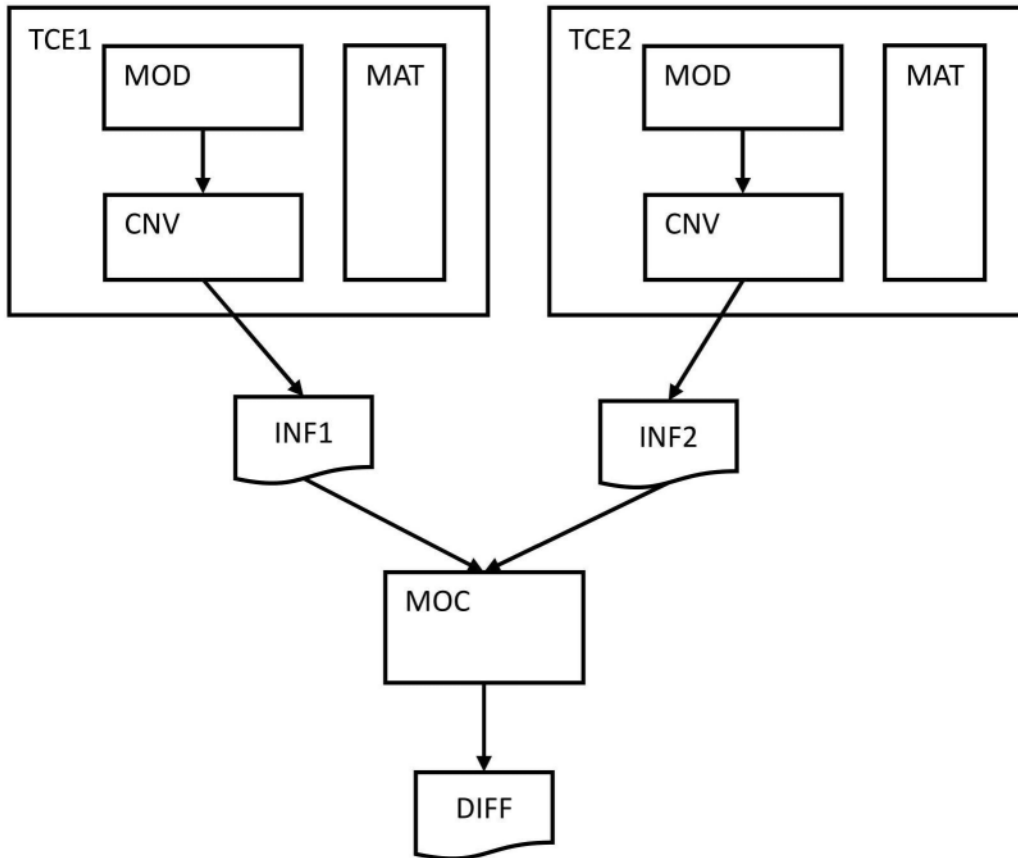


图4

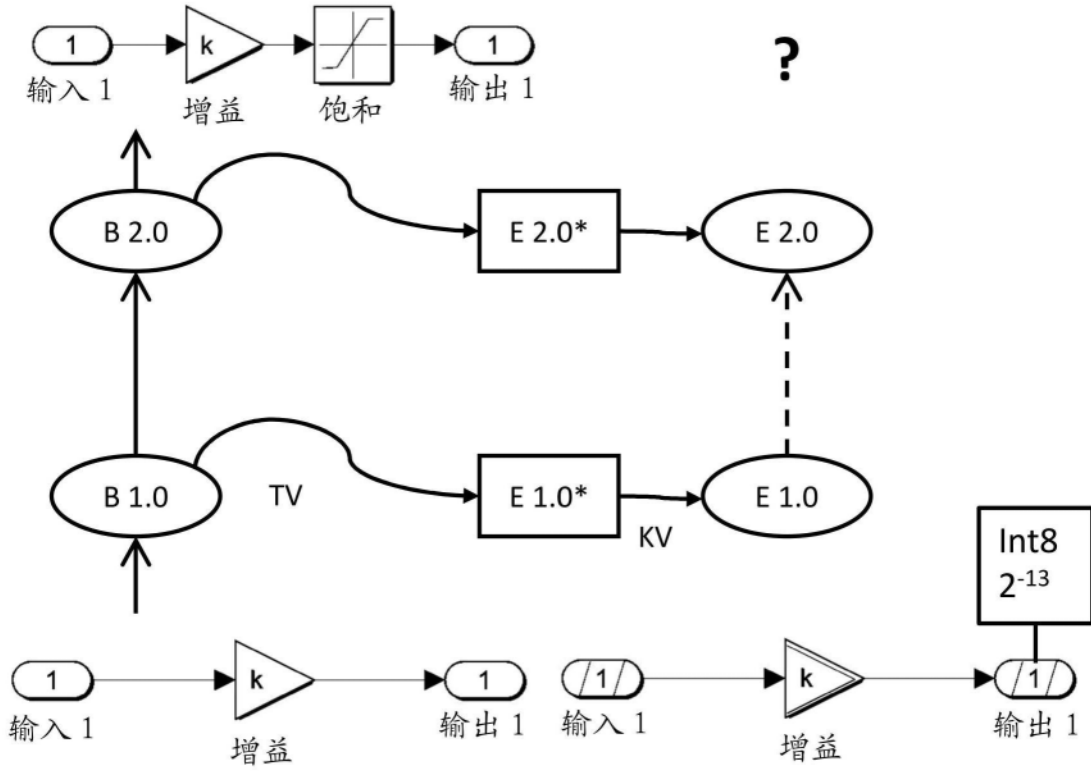


图5

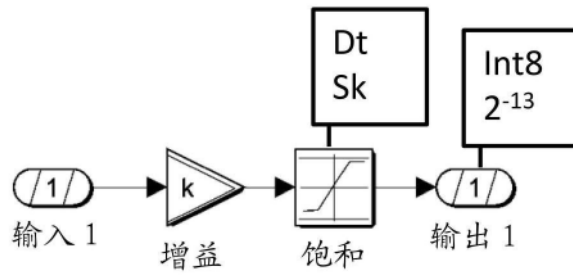


图6

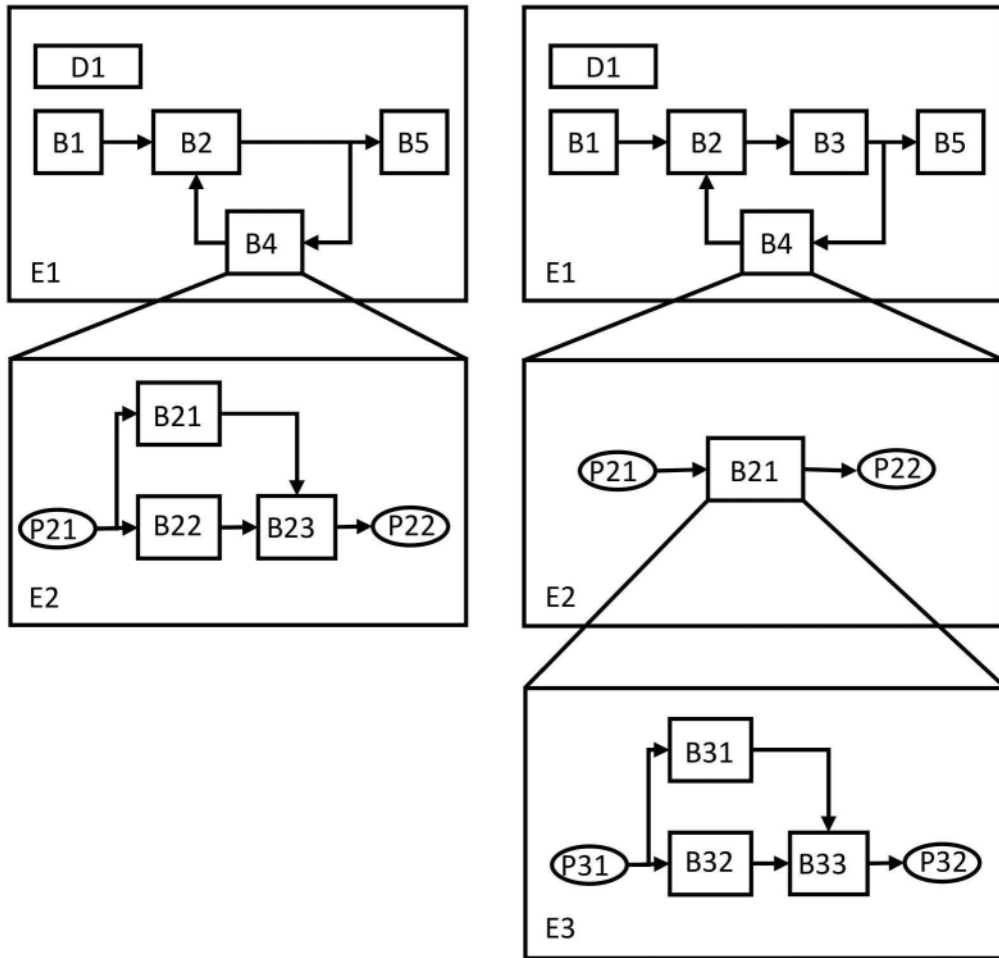


图7

过滤器		动作	
路径	<input type="text"/>	类型	<input type="text"/>
块类型	<input type="text"/>	名称	<input type="text"/>
		等级	<input type="text"/>
合理性			
参数	<input type="text"/>		
最小值	<input type="text"/>		
最大值	<input type="text"/>		

图8