



(12)发明专利

(10)授权公告号 CN 104679807 B

(45)授权公告日 2018.06.05

(21)申请号 201410653054.8

(22)申请日 2009.06.30

(65)同一申请的已公布的文献号
申请公布号 CN 104679807 A

(43)申请公布日 2015.06.03

(30)优先权数据
61/077,022 2008.06.30 US

(62)分案原申请数据
200980133372.9 2009.06.30

(73)专利权人 起元技术有限责任公司
地址 美国马萨诸塞州

(72)发明人 斯科特·斯图德尔 乔尔·古尔德
戴维·菲利莫尔

(74)专利代理机构 北京林达刘知识产权代理事
务所(普通合伙) 11277

代理人 刘新宇

(51)Int.Cl.
G06F 17/30(2006.01)

(56)对比文件
CN 101208695 A,2008.06.25,
US 2007021995 A1,2007.01.25,
US 2008059436 A1,2008.03.06,
庄力可等.基于时间密度的Web日志用户浏
览行为分析.《计算机科学》.2004,

审查员 汪安

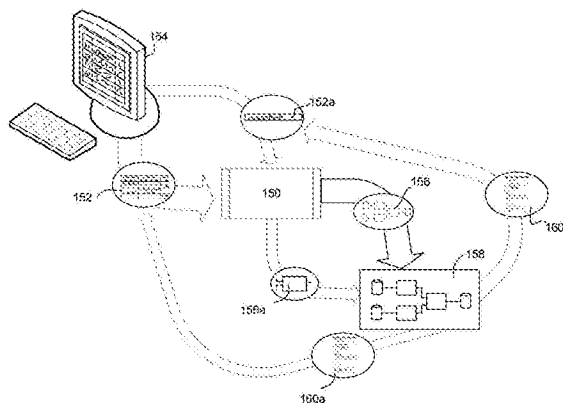
权利要求书4页 说明书24页 附图6页

(54)发明名称

基于图的计算中的数据日志记录

(57)摘要

提供一种方法和计算机系统。所述方法包括：接收针对基于图的计算的至少一个规则说明书，该基于图的计算具有通过链接表示数据流的元素而连接的数据处理组件，该规则说明书定义分别与一个或多个规则例关联的规则，所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则；基于接收到的规则说明书生成针对基于图的计算中的至少一个数据处理组件的变换，包括提供用于配置与所生成的变换关联的日志的特征的接口；以及使用所生成的变换来变换至少一个数据流，包括：在运行时跟踪基于图的计算中的数据流的处理组件的运行，根据所配置的日志特征，基于所跟踪的运行生成日志信息，以及存储或者输出所生成的日志信息。



1. 一种方法,包括:

接收针对一组数据处理组件的至少一个规则说明书,该规则说明书定义每一个都与一个或多个规则例相关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;

基于接收到的规则说明书生成针对至少一个数据处理组件的变换,包括提供用于配置与所生成的变换相关联的日志的特征的接口;以及

使用所生成的变换来变换至少一个数据流,所述至少一个数据流表示数据从一个数据处理组件到另一个数据处理组件的流动,包括:

在运行时跟踪数据处理组件的运行,

根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及

存储或者输出所生成的日志信息。

2. 如权利要求1所述的方法,其中,配置日志的特征包括选择至少一个事件,将针对该至少一个事件而生成日志信息。

3. 如权利要求2所述的方法,其中,根据与接收到的规则说明书相关联的规则,所述至少一个事件与数据流中的变换记录相关联。

4. 如权利要求3所述的方法,其中,所述至少一个事件包括识别将被变换的记录中的错误。

5. 如权利要求3所述的方法,其中,所述至少一个事件包括对于给定记录满足规则例的条件。

6. 如权利要求5所述的方法,其中,对于给定记录满足规则例的条件包括基于给定记录中的值满足逻辑表达式。

7. 如权利要求5所述的方法,其中,对于给定记录满足规则例的条件包括将记录中的值与关联于所述规则例的值相比较。

8. 如权利要求1所述的方法,其中,生成日志信息包括:

生成每一个都包括规则集的细节的一个或多个图例日志消息,该规则集包含由接收到的规则说明书定义的规则,以及

生成多个跟踪日志消息,其中,每个跟踪日志消息与图例记录相关联并且根据与接收到的规则说明书相关联的规则描述与数据流中的变换记录关联的至少一个事件。

9. 如权利要求8所述的方法,其中,描述事件的跟踪日志消息使用图例记录中的索引描述数据处理组件的至少一个输入或输出。

10. 如权利要求8所述的方法,其中,生成一个或多个图例消息包括每次运行计算系统生成一个图例消息。

11. 如权利要求1所述的方法,其中,使用生成的变换的数据处理组件根据与接收到的规则说明书相关联的第一规则对数据流中的输入记录进行变换。

12. 如权利要求11所述的方法,其中,基于所跟踪的运行生成日志信息包括针对指定准则被满足的第一规则的每个规则例生成日志消息。

13. 如权利要求11所述的方法,其中,基于所跟踪的运行生成日志信息包括针对由所述数据处理组件使用所生成的变换生成的经变换的记录中的每个值生成日志消息。

14. 如权利要求11所述的方法,其中,存储或者输出生成的日志信息包括从一个或多个

所述数据处理组件的日志端口输出日志消息。

15. 如权利要求14所述的方法,其中,存储或者输出生成的日志信息还包括从数据处理组件中的日志端口接收日志消息的数据流并且存储至少部分地源自所述日志消息的日志信息。

16. 如权利要求15所述的方法,还包括过滤接收到的日志消息的数据流并且存储源自日志消息的子集的日志信息。

17. 如权利要求14所述的方法,其中,存储或者输出所生成的日志信息还包括:

从数据处理组件中的日志端口接收指示指定准则被满足的、被触发的规则例的日志消息的数据流,

检查所述日志消息以确定每个规则的每个规则例提供至少一个日志消息的经缩减的输入记录集,所述规则通过由全部输入记录触发的规则说明书所定义,以及

存储经缩减的输入记录集。

18. 一种计算机系统,包括:

存储系统,存储针对一组数据处理组件的至少一个规则说明书,该规则说明书定义每一个都与一个或多个规则例关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;

生成器,被配置为基于接收到的规则说明书生成针对至少一个数据处理组件的变换,包括提供用于配置与所生成的变换关联的日志的特征的接口;以及

计算系统,被配置为使用所生成的变换来变换至少一个数据流,所述至少一个数据流表示数据从一个数据处理组件到另一个数据处理组件的流动,包括:

在运行时跟踪所述计算系统中的数据处理组件的运行,

根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及

存储或者输出所生成的日志信息。

19. 如权利要求18所述的计算机系统,其中,配置日志的特征包括选择至少一个事件,将针对该至少一个事件而生成日志信息。

20. 如权利要求19所述的计算机系统,其中,根据与接收到的规则说明书相关联的规则,所述至少一个事件与数据流中的变换记录相关联。

21. 如权利要求20所述的计算机系统,其中,所述至少一个事件包括识别将被变换的记录中的错误。

22. 如权利要求20所述的计算机系统,其中,所述至少一个事件包括对于给定记录满足规则例的条件。

23. 如权利要求22所述的计算机系统,其中,对于给定记录满足规则例的条件包括基于给定记录中的值满足逻辑表达式。

24. 如权利要求22所述的计算机系统,其中,对于给定记录满足规则例的条件包括将记录中的值与关联于所述规则例的值相比较。

25. 如权利要求18所述的计算机系统,其中,生成日志信息包括:

生成每一个都包括规则集的细节的一个或多个图例日志消息,该规则集包含由接收到的规则说明书定义的规则,以及

生成多个跟踪日志消息,其中,每个跟踪日志消息与图例记录相关联并且根据与接收

到的规则说明书相关联的规则描述与数据流中的变换记录关联的至少一个事件。

26. 如权利要求25所述的计算机系统,其中,描述事件的跟踪日志消息使用图例记录中的索引描述数据处理组件的至少一个输入或输出。

27. 如权利要求25所述的计算机系统,其中,生成一个或多个图例消息包括每次运行基于图的计算生成一个图例消息。

28. 如权利要求18所述的计算机系统,其中,使用生成的变换的数据处理组件根据与接收到的规则说明书相关联的第一规则对数据流中的输入记录进行变换。

29. 如权利要求28所述的计算机系统,其中,基于所跟踪的运行生成日志信息包括针对指定准则被满足的第一规则的每个规则例生成日志消息。

30. 如权利要求28所述的计算机系统,其中,基于所跟踪的运行生成日志信息包括针对由所述数据处理组件使用所生成的变换生成的经变换的记录中的每个值生成日志消息。

31. 如权利要求28所述的计算机系统,其中,存储或者输出生成的日志信息包括从一个或多个所述数据处理组件的日志端口输出日志消息。

32. 如权利要求31所述的计算机系统,其中,存储或者输出生成的日志信息还包括从数据处理组件中的日志端口接收日志消息的数据流并且存储至少部分地源自所述日志消息的日志信息。

33. 如权利要求32所述的计算机系统,其中,存储或者输出生成的日志信息还包括过滤接收到的日志消息的数据流并且存储源自日志消息的子集的日志信息。

34. 如权利要求31所述的计算机系统,其中,存储或者输出所生成的日志信息还包括:
从数据处理组件中的日志端口接收指示指定准则被满足的、被触发的规则例的日志消息的数据流,

检查所述日志消息以确定每个规则的每个规则例提供至少一个日志消息的经缩减的输入记录集,所述规则通过由全部输入记录触发的规则说明书所定义,以及
存储经缩减的输入记录集。

35. 一种计算机系统,包括:

用于接收针对一组数据处理组件的至少一个规则说明书的装置,该规则说明书定义每一个都与一个或多个规则例相关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;

用于基于接收到的规则说明书生成针对至少一个数据处理组件的变换的装置,包括提供用于配置与所生成的变换相关联的日志的特征的接口;以及

使用所生成的变换来变换至少一个数据流的装置,所述至少一个数据流表示数据从一个数据处理组件到另一个数据处理组件的流动,包括:

在运行时跟踪数据处理组件的运行,

根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及

存储或者输出所生成的日志信息。

36. 如权利要求35所述的计算机系统,其中,配置日志的特征包括选择至少一个事件,将针对该至少一个事件而生成日志信息。

37. 如权利要求36所述的计算机系统,其中,根据与接收到的规则说明书相关联的规则,所述至少一个事件与数据流中的变换记录相关联。

38. 如权利要求37所述的计算机系统,其中,所述至少一个事件包括识别将被变换的记录中的错误。

39. 如权利要求37所述的计算机系统,其中,所述至少一个事件包括对于给定记录满足规则例的条件。

40. 如权利要求39所述的计算机系统,其中,对于给定记录满足规则例的条件包括基于给定记录中的值满足逻辑表达式。

41. 如权利要求39所述的计算机系统,其中,对于给定记录满足规则例的条件包括将记录中的值与关联于所述规则例的值相比较。

42. 如权利要求35所述的计算机系统,其中,生成日志信息包括:

生成每一个都包括规则集的细节的一个或多个图例日志消息,该规则集包含由接收到的规则说明书定义的规则,以及

生成多个跟踪日志消息,其中,每个跟踪日志消息与图例记录相关联并且根据与接收到的规则说明书相关联的规则描述与数据流中的变换记录关联的至少一个事件。

43. 如权利要求42所述的计算机系统,其中,描述事件的跟踪日志消息使用图例记录中的索引描述数据处理组件的至少一个输入或输出。

44. 如权利要求42所述的计算机系统,其中,生成一个或多个图例消息包括每次运行基于图的计算生成一个图例消息。

45. 如权利要求35所述的计算机系统,其中,使用生成的变换的数据处理组件根据与接收到的规则说明书相关联的第一规则对数据流中的输入记录进行变换。

46. 如权利要求45所述的计算机系统,其中,基于所跟踪的运行生成日志信息包括针对指定准则被满足的第一规则的每个规则例生成日志消息。

47. 如权利要求45所述的计算机系统,其中,基于所跟踪的运行生成日志信息包括针对由所述数据处理组件使用所生成的变换生成的经变换的记录中的每个值生成日志消息。

48. 如权利要求45所述的计算机系统,其中,存储或者输出生成的日志信息包括从一个或多个所述数据处理组件的日志端口输出日志消息。

49. 如权利要求48所述的计算机系统,其中,存储或者输出生成的日志信息还包括从数据处理组件中的日志端口接收日志消息的数据流并且存储至少部分地源自所述日志消息的日志信息。

50. 如权利要求49所述的计算机系统,其中,存储或者输出生成的日志信息还包括过滤接收到的日志消息的数据流并且存储源自日志消息的子集的日志信息。

51. 如权利要求48所述的计算机系统,其中,存储或者输出所生成的日志信息还包括:

从数据处理组件中的日志端口接收指示指定准则被满足的、被触发的规则例的日志消息的数据流,

检查所述日志消息以确定每个规则的每个规则例提供至少一个日志消息的经缩减的输入记录集,所述规则通过由全部输入记录触发的规则说明书所定义,以及存储经缩减的输入记录集。

基于图的计算中的数据日志记录

[0001] 本申请是申请日为2009年6月30日、申请号为200980133372.9(国际申请号为PCT/US2009/049299)、发明名称为“基于图的计算中的数据日志记录”的发明专利申请的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请要求于2008年6月30日提交的美国临时申请No.61/077,022的优先权,通过引用的方式将其合并于此。

技术领域

[0004] 本说明书一般涉及基于图的计算中的数据日志记录(logging)。

背景技术

[0005] 复杂计算经常能够通过有向图被表达为数据流,有向图具有与图的顶点关联的计算的组件以及与图的连接(弧、边)相应的组件之间的数据流。在美国专利5,966,072(“Executing Computations Expressed As Graphs”)中描述了实现这样的基于图的计算的系统。在一些情况下,以可供人读的形式——被称为“business rules(商务规则)”——描述与顶点关联的计算。

发明内容

[0006] 一般说来,在一个方面,提供一种方法,包括:接收针对一组数据处理组件的至少一个规则说明书,该规则说明书定义每一个都与一个或多个规则例关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;基于接收到的规则说明书生成针对至少一个数据处理组件的变换,包括提供用于配置与所生成的变换关联的日志的特征的接口;以及使用所生成的变换来变换至少一个数据流,所述至少一个数据流表示数据从一个数据处理组件到另一个数据处理组件的流动,包括:在运行时跟踪数据处理组件的运行,根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及存储或者输出所生成的日志信息。

[0007] 多个方面可以包括一个或多个下列特征:

[0008] 配置日志的特征包括选择至少一个事件,将针对该至少一个事件而生成日志信息。

[0009] 根据与接收到的规则说明书关联的规则,所述至少一个事件与数据流中的变换记录关联。

[0010] 所述至少一个事件包括识别将被变换的记录中的错误。

[0011] 所述至少一个事件包括对于给定记录满足规则例的条件。。

[0012] 对于给定记录满足规则例的条件包括基于给定记录中的值满足逻辑表达式。

[0013] 对于规则例满足规则例的条件包括将记录中的值与关联于所述规则例的值相比较。

[0014] 生成日志信息包括:生成分别包括规则集的细节的一个或多个图例日志消息,该规则集包含由接收到的规则说明书定义的规则,以及生成多个跟踪日志消息,其中,每个跟踪日志消息与图例记录关联并且根据与接收到的规则说明书关联的规则描述与数据流中的变换记录关联的至少一个事件。

[0015] 描述事件的跟踪日志消息使用图例记录中的索引描述数据处理组件的至少一个输入或输出。

[0016] 生成一个或多个图例消息包括每次运行基于图的计算生成一个图例消息。

[0017] 使用生成的变换的数据处理组件根据与接收到的规则说明书关联的第一规则对数据流中的输入记录进行变换。

[0018] 基于所跟踪的运行生成日志信息包括针对指定准则被满足的、第一规则的各个规则例生成日志消息。

[0019] 基于所跟踪的运行生成日志信息包括针对由所述数据处理组件使用所生成的变换生成的变换生成的经变换的记录中的每个值生成日志消息。

[0020] 存储或者输出生成的日志信息包括从所述一个或多个数据处理组件的日志端口输出日志消息。

[0021] 存储或者输出生成的日志信息还包括从数据处理组件中的日志端口接收日志消息的数据流并且存储至少部分地源自所述日志消息的日志信息。

[0022] 该方法还包括:过滤接收到的日志消息的数据流并且存储源自日志消息的子集的日志信息。

[0023] 存储或者输出所生成的日志信息还包括:从指示指定准则被满足的、被触发的规则例的数据处理组件中的日志端口接收日志消息的数据流,检查所述日志消息以确定每个规则例提供至少一个日志消息的经缩减的输入记录集,所述规则通过由全部输入记录触发的规则说明书所定义,以及存储经缩减的输入记录集。

[0024] 一般说来,在另一方面,提供一种计算机系统,包括:存储系统,存储针对一组数据处理组件的至少一个规则说明书,该规则说明书定义每一个都与一个或多个规则例关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;生成器,被配置为基于接收到的规则说明书生成针对基于图的计算中的至少一个数据处理组件的变换,包括提供用于配置与所生成的变换关联的日志的特征的接口;以及计算系统,被配置为使用所生成的变换来变换至少一个数据流,所述至少一个数据流表示数据从一个数据处理组件到另一个数据处理组件的流动,包括:在运行时跟踪所述计算系统中的数据处理组件的运行,根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及存储或者输出所生成的日志信息。

[0025] 一般说来,在另一方面,提供一种计算机程序,存储在计算机可读介质上,该计算机程序包括导致计算机系统执行以下步骤的指令:接收针对基于图的计算的至少一个规则说明书,该基于图的计算具有通过链接表示数据流的元素而连接的数据处理组件,该规则说明书定义分别与一个或多个规则例关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;基于接收到的规则说明书生成针对基于图的计算中的至少一个数据处理组件的变换,包括提供用于配置与所生成的变换关联的日志的特征的接口;以及使用所生成的变换来变换至少一个数据流,包括:在运行时跟踪基于图的计算中的数

据处理组件的运行,根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及存储或者输出所生成的日志信息。

[0026] 一般说来,在另一方面,提供一种计算机系统,包括:用于接收针对一组数据处理组件的至少一个规则说明书的装置,该规则说明书定义每一个都与一个或多个规则例关联的规则,所述规则例指定用于确定取决于输入数据的一个或多个输出值的准则;用于基于接收到的规则说明书生成针对基于图的计算中的至少一个数据处理组件的变换的装置,包括提供用于配置与所生成的变换关联的日志的特征的接口;以及用于使用所生成的变换来变换至少一个数据流的装置,所述至少一个数据流表示数据从一个数据处理组件到另一个数据处理组件的流动,包括:在运行时跟踪数据处理组件的运行,根据所配置的日志特征,基于所跟踪的运行生成日志信息,以及存储或者输出所生成的日志信息。

[0027] 在附图以及下面的描述中给出本发明的一个或多个实施例的细节。本发明的其它特征、目的和优点将从说明书和附图以及权利要求中变得明显。

附图说明

[0028] 图1A示出在基于图的计算环境中将输入数据变换为输出数据。

[0029] 图1B示出在基于图的计算中的图的一个例子。

[0030] 图1C示出利用日志记录信息的变换生成的框图。

[0031] 图2A示出基于电子表格的规则条目(entry)的一个例子。

[0032] 图2B示出单个规则的一个例子。

[0033] 图3示出在基于图的计算环境中将数据记入日志的部分操作的流程图。

[0034] 图4是在基于图的计算环境中的日志记录结构的图形用户接口的图示。

具体实施方式

[0035] 用于数据日志记录和审核机制的示范性上下文(context)包括基于图的计算范例(paradigm),其将与基于图的计算关联的元数据存储于基于图的计算系统中。在此场景中,每个计算机程序使用计算图实现,也称作数据流图,或者简称为图。一个图包括表示数据处理组件的一个或多个节点或者顶点,它们由表示组件之间数据的流动的有向边联接(join)。图能够在并行处理环境下运行。系统跟踪图的开发过程中的变化,执行统计分析和相关性分析,并且管理与图的开发有关的元数据。与图有关的元数据的存储使得数据影响分析能够发生,给用户视觉感知:数据如何随其在图中的行进而改变,以及那些变化对其它图的影响。另外,系统提供的配置/变化管理能够当存在代码变化时存储图的多个版本,由此确保最新的代码和数据可用。

[0036] 商务规则,比如元数据的子集,存储在系统中。例如,在美国申请No. 11/733,434——通过引用的方式将其合并于此——中描述了各个方面的商务规则。每个商务规则都能够存储在单独的对象中。商务规则能够表达为用于将数据从一种格式转换到另一种格式、进行有关数据的判定或者基于一组输入数据生成新数据的一组准则。例如,在图1A中,航班预订系统中的记录102包括表示乘客的姓名104、今年已经飞行了多少英里106、他的票的等级108以及他所坐的行110的字段的价值。商务规则表示这样的乘客应该放进登机组1。商务规则一般容易为人理解,即,“头等舱乘客在组1”,但是在它可以用于操作数据以前可能

需要被翻译成计算机能够理解的语言。

[0037] 为了在基于图的计算环境中实现商务规则,生成变换(transform) 112,其从一个或多个数据源——例如输入数据集100——接收诸如记录102这样的输入记录,并且提供输出记录,例如,记录114,表示对于输出数据集120来说乘客姓名104以及他在哪个组118。在本例中,数据集被示出为显示一个示范性记录,但是通常数据集可以包括任意数量的记录。输入和输出数据集可以被处理为数据流,例如,像组成数据集的数据流入图或者流出图一样。

[0038] 然后,可以在基于图的计算中实现变换,该基于图的计算具有通过链接表示数据流的元素而连接的数据处理组件。例如,图1B的简单计算图130将输入视为两个数据集132、134(例如,经常飞行数据和航班预定数据),将每个集合中的数据在单独的格式化组件136、138中进行格式化从而它们能被一起使用,并且在联合组件140中将它们联合以产生输出数据集142。变换本身可以是基于图的计算,诸如图130中的计算,或者可以在图的组件内实现,诸如组成图130的独立组件136、138和140。

[0039] 为了为非专业用户简化变换的创建,为这些用户以他们熟悉的格式提供了一种工具用于输入一组商务规则,称为规则集,该工具告诉计算机系统他们需要变换做些什么。规则集是产生单个变换的一组规则。一个规则可以由一个或多个规则例(rule case)组成,这些规则例根据输入来确定用于规则的输出的不同值。一个规则也可以包括其它规则。一个规则集中的其它规则可以产生针对额外输出或者替换输出的值。规则集可以包含其它规则集,这称为“内含(included)”规则集。

[0040] 在图1C中示出利用日志记录信息(logging information)的变换生成系统的一般模型。商务规则环境(business rules environment, BRE)包括生成器150,其从编辑器154接收规则集152作为输入并且生成变换156。作为变换生成选项中的一个,可以随后通过定制图形用户接口中的各种日志记录(logging)事件和信息来激活日志记录。日志(log)是发生在组织的系统和网络内的事件的记录。日志由条目组成;每个条目都包含与已经在系统或者网络内发生的特定事件有关的信息。日志能够用于解决问题,以及用于提供许多功能,诸如优化系统和网络性能、记录用户的动作以及提供对调查异常活动有用的数据。日志能够包含与许多不同类型的事件有关的信息。根据系统的体系结构以及变换和商务规则的目的,所生成的变换156可以提供给基于图的计算系统158作为将用于图中的一个组件或者作为整个图本身。例如,生成器150可以是编译器、定制的程序或者另一个基于图的计算,被配置为使用标准工具以接收规则集152并输出变换156。

[0041] 生成器150还可以在规则集152被编辑时更新变换156。当规则集152被编辑时,编辑器154可以为生成器提供整个规则集或者它可以仅提供新的或者修改的规则或规则例152a。根据使用变换的系统的性能以及需要,生成器150可以生成全新的变换以替换初始变换156,或者它可以提供包含所述变换的组件156a。

[0042] 不需要一个单独的、专用的运行引擎用于在基于图的计算158期间执行日志记录操作。日志记录能够被配置为当其被运行时使用由图组件调用的函数而发生。针对不同的日志记录结构,能够提供针对规则运行的不同审核报告。例如,如图1C中虚线箭头所示(而非实线箭头示出的实际数据流),日志160a可以回溯到规则集152中的特定输入记录;而日志160b可以反映专用规则例152a在早先时间就已经被激发(fire)了。

[0043] 参考图2A, 在一些例子中, 可以将规则以电子表格格式输入。电子表格200中的触发列202、204、206、208相应于可用数据值, 以及行210a-h相应于规则例, 即, 与可用数据值有关的准则集合。如果对于在其中规则例具有准则的各个触发列来说, 给定记录(例如, 图1A中的102)的数据值满足触发准则, 那么规则例210n就应用于该记录。如果应用规则例210n, 则基于一个或多个输出列212生成输出。其所有的触发准则都得以满足的规则例可以称为“已触发(triggered)”。每个输出列212相应于一个潜在的输出变量, 可应用的行210n的相应单元格中的值为该变量确定输出——如果有的话。单元格可以包含分配给变量的值或者它可以包含必须被评估以生成输出值的表达式, 如下面将讨论的那样。虽然在图2A中仅示出一个输出列, 但是可以存在多于一个输出列。

[0044] 可以存在若干不同类型的触发列, 包括相应于变量的列、包含表达式但是对其计算一次然后将其当作变量对待的列、以及仅包含表达式的列。其它列类型包括仅包含数据的列, 以及基于仅包含数据的列指定对于每行进行评估的表达式列。仅包含表达式的列比那些与变量相应的列或者当作变量对待的列简单。

[0045] 在图2A的例子中, 第一行210a仅在一列202中具有准则, 这表示如果一个旅客的经常飞行英里的总数大于1,000,000, 那么不管任何其它列可能具有什么值都应用该规则例。在该情况下, 针对该用户的“乘机组(Boarding Group)”输出变量被设置为组1。同样地, 第二规则例210b表示任何在头等舱的乘机者都在组1中。在一些例子中, 规则被顺次评估, 因此一个超过1,000,000英里且持头等舱票的旅客将在组1中, 但是将仅触发第一规则例210a。一旦触发了规则例, 就不需要评估该规则中的其它规则例了。

[0046] 接下来的规则例210c基于两个输入值202和204——如果针对经常飞行英里总数和当年英里数二者定义的准则都满足的话, 那么该乘机者在组2中。在第四规则例210d中, 任何的商务舱客户也在组2中。其余的规则例210e-h包含与其它规则例相关的准则, 即, “其它(else)”和“相同的(same)”。“其它(Else)”表示在位于那一行之上且具有与左边相同准则的行中(即, 规则210b和210d)、那一列中没有一个准则得以满足, 而“相同的(same)”表示如果针对该列应用了在该规则例之上的规则例则应用该规则例。因此, 第五规则例210e应用于任一这样的记录: 其不匹配头两列202或204中的任一准则(因为它应当已经由规则例210a或210c处理过了), 在“座位等级”列中不具有“头等”或“商务”(列206中的“其它(else)”关键字), 并且具有小于或等于10的“座位行”值208。其余规则例210 f-h中的每一个都应用于这样的记录: 其也不匹配具有列202或204中的值的、任一较高的规则例, 在“座位等级”列中不具有“头等”或“商务舱”, 并且具有适当的“座位行”值。

[0047] 图2A的例子中的规则例210a-h也可以表示为独立的简单规则, 各自在它们自己的电子表格中, 如图2B所示。规则220a-d分别相应于图2A的行210a-d, 而规则220e具有相应于合起来的行210e-h的四个规则例。用户可以分别地创建这些独立规则, 而非生成图2A中示出的整个表格。每个规则例包含用于各个触发列的值以及用于各个输出列的值(该值可以为空, 即, 实际上被设置为“任意”)。当多个规则生成相同的输出时, 将这些规则排序并且对它们按次序进行考虑, 直到针对输入触发一个规则中的规则例并且生成输出为止。如果规则中没有规则例被触发, 那么就处理生成相同输出的下一个规则。如果在任一规则中都没有例对于某一输出触发, 那么就使用缺省值。

[0048] 在一些例子中, 编辑器接口154可以用图识别包含表达式的单元格。这将帮助用户

理解两个表达式之间的差异,一个表达式将独立地被评估为真或假,一个表达式返回与列变量相比较的值。当用户打字时,他可以例如通过在开头处打星号,来指示特定单元格将是一个表达式单元格。

[0049] 对于相应于输出变量的列,单元格可以包含下列中的一个:

[0050] • 值。该值将分配给输出变量。

[0051] • 表达式。该表达式的值将分配给输出变量。如果表达式评估为空 (NULL),那么该字段得到空值,除非输出字段不可为空。在输出字段不可为空但得到空值的情况下,生成错误。

[0052] • 关键字“空 (null)”。如果输出字段可为空,那么该字段将分配为空。否则,生成错误。

[0053] • 空字符串。如果输出字段具有缺省值,则分配该缺省值。否则,单元格被当作好像它包含关键字“空 (null)”一样对待。

[0054] • 关键字“相同的 (same)”。输出字段被赋以在上面的单元格中所计算的相同的值。

[0055] 如果可能的话,在检测到错误时就将其报告,即将“null”放入非可空字段的输出列中。然而,一些错误直到测试时间或运行时间才被报告。

[0056] 是创建为表格行还是独立规则,每个规则都具有特定的特征集合。规则集可以为它们包括的规则来确定这些特征。这些特征可以包括名称、规则类型、说明和注解字段、输出变量列表、输入变量列表、自变量列表、触发列列表、修改历史、测试数据集和错误处理行为。名称是不言自明的,并且用于列出规则集中的规则。在一些例子中,规则类型是规则集的属性。输出变量列表是由规则指定的值或生成的变量集合。这可以从规则集继承而来,并且可以存在一个或多个输出。输入变量列表识别规则用于评估记录所需要的所有变量,包括在列的最上面那些以及内部表达式使用的那些(例如,图2A中的规则210c所使用的“去年经常飞行英里数”值在表达式中使用但是不具有其自己的列)。

[0057] 规则可以是单发的 (single-fired) 或者多发的 (multi-fired)。例如,多个规则例可以用于针对一个或多个输出生成多个值。能够触发多个规则例的规则称为多发规则。多发规则是基于由规则计算的输出的类型而单独地识别的。如果由一个规则计算的(多个)输出是多个列表(在每个记录中可能具有多个值的输出),那么该规则就是一个多发规则。在多发规则中,一旦一个规则例被触发,相应的输出值就被附加到输出的值的列表中。然而,不同于单发规则,在多发规则中,评估即使在规则例被触发之后也继续进行。每个后续规则例也被评估,并且触发的各个规则例将导致其它值被附加到针对(多个)输出的值的列表中。

[0058] 在一些例子中,可以通过进行“与 (AND)”运算的行和进行“或 (OR)”运算的列中的规则例,以与上述相反的方式来评估规则。也就是说,只有当每行触发(对行进行“与”运算)但是仅一个单元格需要针对每行为真以触发(对列进行“或”运算),规则才产生输出。

[0059] 自变量列表仅对函数规则存在。它识别输入到规则的参数的名称和类型,并且可以是规则集的属性。触发列列表识别哪些列可以触发规则的应用。除了图2A和2B的例子中示出的输入变量之外,触发列可以相应于参数、查找变量、来自较早规则的输出变量、内含规则集的输出变量、对规则集的参数或者表达式。它们还可以包括来自函数规则的输入变

量,即,自变量。

[0060] 错误处理确定从规则集创建的变换如何处理当评估规则时发生的错误。对于处理触发表达式中的错误来说,选项是将允许错误或将忽略错误,在允许错误的情况下,变换拒绝导致错误的记录,忽略错误相当于假定触发表达式为假并且继续前进到下一个规则。对于输出表达式来说,可以通过以下方式来处理错误:允许错误且拒绝记录,忽略错误且将输出设置为空,或者忽略规则中的行且继续前进到下一行。

[0061] 如上所述,从规则集生成变换。规则集可以具有下列特征:

[0062] 名称、说明、以及注解——这些标识规则集。根据后端实施方式,规则集可以包括其在系统内位置的标识。在一些例子中,规则集的位置是项目中的路径。在一些例子中,规则集可以组织在关系数据库中并且通过名称定位。修改历史包括修改名称、日期以及登记注解(check-in comments)。

[0063] 变换类型——确定从规则集中生成什么类型的变换。可能值包括重新格式化、联合、上卷(rollup)以及过滤,如下面讨论的那样。

[0064] 输入数据集——提供字段列表以及经命名的常数以供编辑。在一些例子中,当生成变换时,缺省地,将假定输入数据集中的一个的记录格式。可以存在多个输入数据集,允许规则集生成针对不同环境的变换。还允许多个逻辑到物理映射的集合,即,不同的物理名称集合。在一些例子中,存在具有一个或多个数据集的输入映射表。在一些例子中,联合组件可以具有多个输入映射表,并且每个可以具有多个数据集。

[0065] 输出数据集——提供输出字段名称列表。缺省地,当生成变换时将假定输出数据集中的一个的记录格式。输出数据集可以与输入数据集相同。内含规则集将不具有输出数据集。在一些例子中,因为具有输入数据集,所以存在多个输出数据集,允许规则集生成针对不同环境的变换。

[0066] 内含规则集列表——一个规则集可以使用由另一个规则集计算的输出字段(明确列出的输出字段,而非输出记录格式的字段)。基于对从包含规则集中可见的内含规则集中输出的变量的集合进行定义的内含规则集映射表,内含规则集中的输出变量可以用作包含规则集中的变量。

[0067] 内含变换文件列表——可以选择性地包括对在处理规则集时将使用的变换进行指定的一个或多个文件。

[0068] 列出变量和常量的一系列映射表——这些表与输入和输出数据集交织在一起。它们使得变量列表为编辑器所知并且将商务名称与技术名称之间的映射记入文档。每个变量具有一个商务名称、技术名称(其可以使用表达式计算)和基本类型(字串、数字、日期或者日期时间)。与每个变量关联的变量表是将商务名称与技术名称之间的映射记入文档的常量的可选列表。下面更详细地描述变量表。

[0069] 对外部测试数据文件的参考——测试文件用于测试规则,类似于上面讨论的嵌入式测试数据集。

[0070] 不拒绝标志——如果该标志被设定的话,那么由规则集生成的变换将不拒绝记录(抛弃错误)。可以使用该标志以使得抛弃错误的规则将被忽略,好像那些规则从未触发一样。

[0071] 部署表——列出一个或多个部署,其(间接地)指示哪些规则应该包括在各个创建

中。稍后更详细地描述部署表。

[0072] 可选关键字——使用户能够指定专用输入字段的商务名称,该商务名称代表用于联合类型和上卷类型规则集的关键字。在一些例子中,关键字实际上通过关键字的类型被实现为输入变量表中的一个条目。

[0073] 查找文件的可选列表——提供商务名称、关键字信息以及输入变量和常量的完整表,每个查找文件一个表。下面更详细地描述查找文件支持。

[0074] 参数表——列出多个变量,这些变量的值来自环境或者来自运行时参数集。

[0075] 规则集与若干不同表关联:

[0076] 1. 输入变量和常量表。对于变换类型的规则集,该表包含规则中将被引用的输入记录格式的字段。不是每个记录格式的字段都需要列出,但是它们通常是如此。对于联合类型的规则集,将存在多个输入表,每个表代表用于联合操作的一个输入数据集。

[0077] 2. 用于所有内含规则集的输入变量和常量表。在使用内含规则集时,每个内含规则集都具有它自己的输入变量和常量表。当建立一个变换时,由内含规则集使用的输入变量被映射到完成该包含的规则集的上下文中的实际输入。因此,该列表被提升到包含规则集中。如果多个内含规则集被包括,那么每个输入变量表都被提升。(如果内含规则集本身包括一个规则集,那么就不提升第二级变量。)从内含规则集提升的输入变量和常量不可用于包含规则集中。该表被包括,因此可以在到内含规则集的输入与到包含规则集的输入之间建立映射。更多细节参见下文。

[0078] 3. 用于所有内含规则集的输出变量和常量表。当规则集已经被内含时,那些内含规则集的输出变为到包含规则集的输入。该表列出所有那些变量。它一开始被直接填充以所有内含规则集中的输出变量和常量表;然而,可以改变商务名称以避免名称冲突。对于该表来说,技术名称是该内含规则集内部的商务名称。

[0079] 4. 输出变量和常量表。对于变换类型的规则集来说,该表包含将被规则集计算的输出记录格式的字段。不进行计算的输出变量也可以被包括并且将被规则集忽略。(所生成的变换具有用于将输入复制到输出的通配符规则。另外,输出可以包括缺省值在其中。)

[0080] 输出变量也可以被用作中间变量,意指由一个规则产生的输出的值可以在稍后的规则中被引用。有时,通过这样的方式使用输出而不将其直接包括在来自变换的输出记录中。

[0081] 5. 参数表。规则可以包括对参数的引用。在运行时在图的参数集的上下文中解析参数。类似于其它变量,在规则集中参数具有商务名称、技术名称(例如,\$RUNDATE)和类型。

[0082] 6. 针对每个查找文件的变量映射表。这些类似于输入表,但是映射到针对查找文件的记录格式的字段。

[0083] 不共享规则集(被设计用于生成变换)通常受输入和输出数据集两者约束。输入数据集是输入变量的源。输出数据集是输出变量的源。有时一个规则集将具有多个输入数据集和/或多个输出数据集。在那种情况下,每个输入数据集和输出数据集都是变换的可能输入或输出。可以仅存在输入变量的一个集合(除联合操作外),但是对于不同的数据集可以存在不同的商务名称与技术名称之间的映射。在某些情况下,输入变量可以由规则集使用并且存在于一个输入数据集中而不存在于第二输入数据集中。在那种情况下,表达式被指定为第二输入数据集中的缺失变量的技术名称。如果规则集不使用输入变量,那么不必针

对每个输入数据集提供技术名称。

[0084] 内含规则集被稍有不同地对待。内含规则集可以不具有关联的输入和输出数据集。而是,它们具有输入变量和输出变量。包括内含规则集的规则集负责映射输入和输出。

[0085] 变量

[0086] 变量可以具有下列属性,并且可以以表格形式呈现给用户:

[0087] 1. 商务名称(逻辑名称)。商务名称是规则中使用的名称。在一些例子中,施加限制以使得两个输入变量不可以具有相同的名称,两个输出变量不可以具有相同的名称,来自内含规则集的两个输出不可以具有相同的名称,并且同一个查找文件中两个查找变量不可以具有相同的名称。输入变量可以具有与输出变量相同的名称。在这种情况下,用户接口可以基于上下文或者通过使用诸如在输出变量名称的前面的“out.”这样的前缀消除输入与输出的歧义。不同查找文件中的查找变量可以具有相同的名称。因此,使用诸如查找文件本身名称这样的前缀将消除它们的歧义。

[0088] 2. 简单类型。在一些例子中,可以支持四个基本类型——字符串、数字、日期和日期时间。这些相应于类型字符串(int)、十进制(20)、日期(“YYYY-MM-DD”)和日期时间(“YYYY-MM-DD HH24:MI:SS.nnnnnn”)。在变换中使用的在基本类型与实际类型之间的转换将与商务规则的编辑分开进行处理,例如,通过所生成的变换组件。

[0089] 3. 缺省值。缺省值仅为输出变量所需要。它是一个在以下情况时使用的值:(1)在规则中的输出列中对于该输出存在空单元格,或者(2)当没有规则触发以计算该输出的值时。只要输出变量可为空,缺省值就可以是NULL(空)(空单元格被解释为NULL)。

[0090] 缺省值是表达式,就如在规则表达式表中的输出列中所使用的那样。这意味着缺省值可以指代输入变量或输出常量或者包含表达式。缺省值也可以指代其它输出,只要没有引入迂回(circularity)。

[0091] 4. 技术名称(物理名称)或表达式。它是指定变量的表达式。有可能使用表达式而非字段名称用于输入和包括的变量(在一些例子中,不允许对于输出变量使用表达式)。在向量的情况中,表达式应该是充分证明资格的(qualified)。

[0092] 当处理提示的(prompted)变量以及来自内含规则集的输入和输出变量时,与变量关联的技术名称实际上只是共享的规则集内部所使用的商务名称。当处理仅内部使用的输出变量时(在一个规则中计算并且在后来的规则中使用的中间变量),技术名称可以为空。

[0093] 5. 可选说明和注解。

[0094] 常量

[0095] 各种变量表包括对于常量以及变量的映射。常量相应于C++中的enums。系统可以支持最初来自有效值和无效值的常量值,以及最初来自有效范围和无效范围的常量范围。另外,可以创建代表区别性值和/或范围的集合的常量。

[0096] 常量与变量关联。这意味着常量的商务名称不一定在整个规则集上是唯一的。编辑器将基于在规则中常量出现在哪一列而正常知道对于任一常量的上下文;然而,对于用户来说有可能选择属于表达式中不同变量的常量。在那种情况下,常量将通过变量名称来证明资格(例如,“航线等级.商务”)。

[0097] 当计算输出变量时,仅仅使用单个值常量(为输出字段分配范围没有意义)。

[0098] 常量具有下列属性,并且将以表格形式呈现给用户(变量和常量可以混合,类似于

将一个表嵌入到另一个表内部)。

[0099] 1. 变量名称。(多个) 常量应用于一个变量。变量名称实际上是关联变量本身的一部分。

[0100] 2. 商务名称。商务名称是规则中使用的名称。该名称不需要一定是值标识符,具体地说,允许内部间隔和标点。在某些情况下,用于常量的商务名称仅在它们所应用的变量内是唯一的。

[0101] 3. 常量类型。值、范围或集合中的一个。如前所述,范围和集合在用于对比(输入)而非赋值(输出)时是合法的。

[0102] 4. 对于值来说:实际值。在本例中,字符串被引用而数字则不。在缺省形式下引用日期和日期-时间(例如,“YYYY-MM-DD”)。也允许使用表达式,只要该表达式返回能够被自动转换为变量类型的简单类型。

[0103] 当常量是对于内含规则集的输入或输出表的一部分时,不存在值。而是,值是对于相应输入或输出变量的关联常量的商务名称。

[0104] 5. 对于范围来说:最小值和最大值。两者都是常量或表达式,正如上面文件的实际值那样。范围被用作在规则中进行比较的简略的表达方式。对于范围来说只允许等同性比较,并且系统将范围翻译为“变量 \geq 最小值(variable \geq minimum)和变量 \leq 最大值(variable \leq maximum)”。如果没有指定最小值,那么该比较的一部分将被跳过。同样地用于最大值。该范围通过逗号隔开最小值和最大值来存储。

[0105] 6. 对于集合来说:以逗号隔开的值列表。列表的每个元素都是一个常量或表达式,类似于上面文件中的实际值。对于集合只允许等同性比较,并且系统将集合翻译为以“[值的向量列表]的变量成员(variable member of[vector list of values])”形式的表达式。

[0106] 7. 可选说明和注解。

[0107] 当处理来自共享规则集的提示变量时,也提示常量。在示出对于共享规则集的输入和输出变量的表中,也示出与那些变量关联的常量。对于那些常量的缺省映射是提示的信息的一部分,但是用户能够推翻(override)常量值。

[0108] 系统将检测因为不匹配常量而在使用变量中何时存在可能的冲突。具体地说,如果(1)任一变量的值被复制到另一个变量,(2)如果两个变量都定义有常量,以及(3)常量的集合在名称和值两个方面都不一样,那么将在用户需要将一个变量的值翻译为其它值时生成错误。源变量包括输入变量、查找变量、来自内含规则集的输出以及用作输入的输出变量。目标变量包括输出变量以及到内含规则集的输入。在规则表达式中或者在变量表中可能发生赋值。

[0109] 变量的排序

[0110] 为了避免迂回逻辑,系统强制变量和规则的严格排序。全局排序的例子如下:

[0111] 输入变量和参数。

[0112] 第一内含规则集的输入映射。

[0113] 第一内含规则集的输出值。

[0114] ...

[0115] 第n个内含规则集的输入映射。

[0116] 第n个的内含规则集的输出值。

[0117] 第一查找文件的缺省关键字值。

[0118] 第一查找文件的输出字段。

[0119] ...

[0120] 第n个查找文件的缺省关键字值。

[0121] 第n个查找文件的输出字段。

[0122] 所有输出变量的缺省值。

[0123] 每个项目的计算都使用在前步骤中计算的值。这意味着,例如,第一内含规则可以参照它的映射表中的输入变量和参数。然而,第二内含规则可以将它的输入映射到从第一内含规则计算得到的输出。类似地,对于每个输出变量的缺省值在任一规则之前被计算,因此它们基于输入变量、参数、查找文件或者来自任一内含规则的输出的值。当到了实际上计算规则的输出的时间时,按次序评估规则以使得较晚的规则可以使用从较早的规则计算得到的值。

[0124] 将数据集链接到变量

[0125] 在一些例子中,输入变量表直接来自输入数据集记录格式,而商务名称来自有关输入数据集的元数据。然而,在一些例子中,在规则集内部保留该映射的副本有好处。首先,在规则集中保留有变量映射表的副本使得可以在产生环境的上下文之外编辑规则集。规则集以及关联的规则可以被串行而成为一个沙箱(sandbox)并且被编辑为沙箱项目的一部分。其次,保留有输入变量映射表的副本使用户可以解决冲突或相反推翻现有的元数据。例如,如果输入数据集中的两个字段映射到相同的商务名称,那么就可以在输入变量表中改变那些商务名称中的一个。

[0126] 当一个规则集被首次创建时,输入变量表是空的。一旦用户识别出输入数据集,输入变量表就被输入数据集的元数据自动填充。(相同的逻辑应用于输出变量和输出数据集,但是为简单起见,本讨论的余下部分将集中在输入数据集上。)

[0127] 为简单起见,在本说明书中使用单数形式的术语“输入数据集”。存在能够链接到输入变量的零个或多个输入数据集,以及能够链接到输出数据集的零个或多个输入数据集的单独的集合。具体地说,输入变量表具有用于商务名称的一列,用于类型的一列等等,用于技术名称的许多列,每个输入数据集一个技术名称。一旦指定单个输入数据集,就能够使用类似的技术添加第二输入数据集。然而,在第二数据集或者后续数据集的情况中,技术名称和商务名称之间的映射可以是不那么完整的,特别是因为系统可能无法算出第二数据集和后续数据集中每个字段映射到哪一变量。在这些例子中,用户能够手动地校正任一缺失的信息。

[0128] 当最初从输入数据集创建输入表时,输入数据集中的每个字段将导致创建一个输入变量。用于输入变量的技术名称将成为该字段的名称。将基于字段类型分配类型。空缺(void)将被当作字符串,实数(real)将被当作数字。子记录将不具有相应的输入变量,虽然子记录内部的字段将具有相应的输入变量。联合体(union)将导致用于该联合体的每个分支的输入变量。如果元素是一个向量,那么相应的输入变量的技术名称将假定该向量的第一元素(“in.vect[0]”)。用户能够将其推翻。例如,对于一个多输出变换来说,用户可以将技术名称改变为in.vect[index]。或者,用户可以创建与该向量的其它元素相应的附加输入

变量,如果该向量长度固定的话。联合体和向量在输出数据集中可能得不到支持(不会针对它们创建输出变量)。在一些例子中,多输出组件的变化可以输出一个输出向量而非多个输出记录。

[0129] 在一些例子中,商务名称由元数据计算得来。用于确定一个字段的商务名称的逻辑的例子如下:

[0130] 如果字段(物理元素Physical Element)具有display_name(显示名称),那么该字段的display_name被用作商务名称。

[0131] 否则,如果字段具有逻辑元素(Logical Element)并且逻辑元素具有display_name,那么该逻辑元素的display_name被用作商务名称。

[0132] 否则,如果字段具有逻辑元素,那么该逻辑元素的名称被用作商务名称。

[0133] 否则,商务名称从技术名称计算得到。

[0134] 如果存在冲突(重名),那么只有一个商务名称将被分配。其它字段将不分配以任何商务名称。

[0135] 在一些例子中,在规则集和数据集元数据之间不存在动态链接。如果用户改变元数据数据(例如,重新命名逻辑元素),那么那些改变不被系统自动获得。在一些例子中,数据之间的双向关系可以用于使这些改变能够被检测到。

[0136] 如果用户添加第二数据集到规则集,那么系统将设法使用如上所列出的、相同的物理到逻辑映射规则,来填充用于每一个商务名称的字段。如果变量无法被映射,那么那些变量的技术名词对于添加的数据集将变为空并且用户将必须手动地填写字段名称或者表达式。可用字段将被列在用户接口中的下拉菜单中。

[0137] 在从数据集元数据创建输入变量表的同时,可以将常量添加到输入变量表,也可从数据集元数据添加。系统将创建用于所有有效值和无效值以及所有与认证说明书(Validation Spec)关联的有效范围和无效范围的常量,该认证说明书与每个逻辑或物理元素关联。

[0138] 用于确定常量的商务名称的逻辑的例子如下:

[0139] 如果有效值(有效范围,等等)具有display_name(显示名称),那么display_name将被用作商务名称。

[0140] 否则,如果有效值(有效范围,等等)具有描述,那么该描述将被用作商务名称。

[0141] 否则,常量将被包括在变量表中而无需商务名称。

[0142] 没有必要从数据集开始创建变量。创建输入变量列表的第二种方法是标识基础系统中的逻辑实体(Logical Entity)。如果选择了一个逻辑实体,那么系统将针对逻辑实体中的每个逻辑组件来创建具有一个变量的变量表。变量的商务名称将是逻辑组件的display_name。如果逻辑组件具有认证说明书,那么也将使用在前的文件规则创建常量。

[0143] 最后,输入和输出变量能够被手动地添加——要么通过将它们添加到变量表,要么通过在编辑规则的同时创建它们。例如,当用户添加一列到规则中时,他选择哪个输入变量应该用于该列。但是他也能够选择“new... (新的)”并即时地创建输入变量。然后系统将提示用户数据类型和可选注解。直到晚些时候才需要填入技术名称。

[0144] 系统需要变量列表以便使规则能够被编辑。然而,直到晚些时候才必须完成商务名称和技术名称之间的映射。可以仅当用户准备相对于外部测试文件测试整个规则或者准

备实际从规则集创建变换时需要映射。

[0145] 内含规则集(included rule set)

[0146] 在一些例子中,规则集能够被共享。具体地说,内含规则集被设计为包括在另一个规则集之内以使得其逻辑变为包含规则集所生成的变换的一部分。

[0147] 虽然内含规则集通常被特定设计为共享的,但是也能够独立使用内含规则集以创建变换。例如,用户能够创建针对过滤类型(filter-type)的变换计算布尔输出的规则集。但是同时,规则集能够被包括在另一个变换内部并且布尔输出(共享规则集的输出变量,在该包含规则集中可用)可用于计算更复杂的输出。

[0148] 内含规则集类似于其它类型的规则集。它们具有输入变量和输出变量。并且它们本身能够包括其它内含规则集。但是内含规则集中的输入和输出变量的处理不同于变换类型(transform-type)的规则集。在变换类型的规则集中,输入和输出变量被映射到技术名称,因此能够生成变换。但是在内含规则集中,不需要将输入和输出变量映射到技术名称。(如果规则集既被共享又用于生成变换,那么输入和输出变量将被映射到技术名称用于生成变换的部署。)当用户将内含规则集包括在另一个规则集中时,包含规则集需要具有用于映射内含规则集的输入和输出的变量映射表。在包含规则集的上下文中,只有共享规则集的输入变量和输出变量是可见的。包括在共享规则集中的任何规则集的任一变量都不对包含规则集公开。

[0149] 在包含规则集的上下文中,共享规则集的输入变量需要被映射到包含规则集的变量,或者使用那些变量的表达式。共享规则集的商务名称将被列在变量映射表中,但是那些名称将不可用于包含规则集中的规则中。而是,包含规则集可以只需要将共享规则集的每个输入变量(通过商务名称)与包含规则集中的表达式匹配。

[0150] 由于考虑在输入变量、参数和查找之前评估内含规则集,因此内含规则集的输出能够被用作查找的关键字。在一些例子中,评估次序更灵活,并且能够基于相关性分析而自动地确定查找相对评估内含规则集的次序。因为在计算任一输出变量之前评估内含规则集,所以包含规则集中没有输出变量能够被映射到内含规则集中的输入。如果到内含规则集输入的映射不能通过简单输入变量完成,那么可以代之以使用表达式。

[0151] 到内含规则集输入变量的映射可以是NULL(空),只要内含规则集中的输入变量可为空。映射也可以留白。如果映射留白,那么当且仅当在包含规则集的输出的计算中需要该输入变量时,将在变换产生时间报告出错。在一些例子中,假定一切都是可为空的,这将简化用户接口。

[0152] 在包含规则集的上下文中,共享规则集的输入变量也需要映射到包含规则集中的商务名称。该映射表是上面所述一个映射表的反向。当映射共享规则集的输入变量时,表将共享规则集的输入变量的商务名称映射到包含规则集中的现有变量。但是当映射共享规则集的输出变量时,包含规则集具有指定用于共享规则集的输出的商务名称的表——将包含规则集中的名称映射到共享规则集中相应的名称。

[0153] 需要输出变量映射来解决潜在的命名冲突。缺省映射是简单地在包含规则集和共享规则集中都使用相同的商务名称。但是共享规则集中的输出变量的名称可能与包含规则集中已经定义的变量的商务名称冲突,因此可以改变包含规则集中所映射的名称。

[0154] 并不是来自共享规则集的每个输出都需要被映射。如果一个输出未映射,那么该

输出就不能用于包含规则集中,并且将忽略来自共享规则集的相应逻辑。另一方面,来自共享规则集的所有输入都可以被映射,虽然如果规则集设计者确信将不需要它们,那么它们就可能被映射到不感兴趣的变量。在一些例子中,系统本身可以确定哪些输入真的需要被映射。

[0155] 在一些例子中,映射表通过商务名称而非通过引用完成。当共享规则集包括在另一个规则集中时,包含规则集得到来自共享规则集的输入和输出的副本。这些名称与映射信息一起被存储在包含规则集中。可以对共享规则集进行改变,导致添加、删除或者重新命名某些输入或者输出。

[0156] 包含规则集和内含规则集之间的引用完整性问题可以在从系统加载规则集时由包含规则集处理。从包含规则集中删除从共享规则集消失的输入变量。添加到共享规则集的输入变量被添加到包含规则集中的映射表,但是保持不被映射。同样地,添加到共享规则集的输出变量得以添加到包含规则集中的映射表,但是保持不被映射。如果输出变量从共享规则集中被删除,并且在包含规则集中没有使用它,那么就只是从映射表中删除它,但是如果在包含规则集中使用它,那么用户得到一个该变量不再可用的出错消息。

[0157] 包含规则集实际上存留来自共享规则集的冗余信息。具体地说,在输入和输出变量映射表中,包含规则集可以只需要将共享规则集中的商务名称列表与包含规则集中的相应名称一起维持。为了有效率,包含规则集也存留类型、缺省值、说明和注解,它们全部都是从共享规则集中复制而来。这些值在编辑包含规则集时只可读,但是为了生成报告和其它分析的效率而被包括在其中。

[0158] 共享规则集映射表在内含规则集中也具有一个附加条目,该附加条目是一个附加注解。这使用户能够添加其它注解到所映射的值。

[0159] 查找文件

[0160] 规则集可以选择性地具有一个或多个查找文件。规则集中的每个查找文件都包括下列信息:

[0161] 1. 查找文件的商务名称。

[0162] 2. 可选说明和注解。

[0163] 3. 用于组成关键字的商务名称列表。在查找文件被添加到表达式中使用这些名称,因此用户看到类似这样的:lookup(My Lookup File,<customer name key>,<account type key>) (查找(我的查找文件,<客户名称关键字>,<账户类型关键字>))。

[0164] 4. 用于每一个关键字的缺省表达式列表。

[0165] 5. 查找文件的技术名称。在一些例子中,可以在部署中推翻它们。

[0166] 6. 一个或多个查找数据集。每个查找文件被不严格地约束在系统内的数据集上,正如规则集被输入数据集约束一样。缺省地,存在与规则集中的每个查找文件关联的一个查找数据集,但是可以存在更多查找数据集以便用于替换的部署。

[0167] 7. 输入变量和常量表。除了对每个查找文件存在一个表之外,它类似于用于规则集的输入变量和常量表。通过输入变量,用于查找文件的输入变量和常量表可以具有与每个关联的查找数据集相应的多个技术名称。

[0168] 除了可能存在多于一个查找文件之外,类似于输入变量那样对查找文件进行处理。每个查找文件在一个页面上被编辑,在商务名称和技术名称之间具有映射表并且可以

与多个数据集关联。它们也具有与每个字段关联的常量。可以通过从输入数据集加载输入变量的元数据的方式读取查找数据集的元数据,来初始化查找文件的映射。

[0169] 如果用户使用查找字段变量,并且在查找中没有发现关键字,那么该字段的值假定为空。如果该字段为空,除非规则例被专门触发,否则规则例就将评估为假并且被跳过。在这种情况下,不生成出错消息。如果用户使用查找文件变量(查找文件本身而非字段),那么就假定函数lookup_match,因此查找文件变量评估为真或假。两种情况都应用于或者输入列或者输出列的规则表达式。如果用户使用查找字段变量作为缺省输出变量,那么不能找出查找就被译成NULL(空)。

[0170] 与正常处理过程相比,查找操作较慢,在查找操作中关键字被用来从参考文件中检索一个或多个数据记录。BRE包含代码,该代码被设计为通过针对每个记录高速缓存查找结果,来限制花费高的查找操作的数目。每当规则引用查找变量(应当由查找操作返回的值中的一个)时,变换生成过程就将查找操作变为子程序调用。子程序包含在每个记录一开始的时候被初始化为假的全局布尔变量,其指示子程序是否已经被当前记录调用。查找子程序第一次被调用,布尔变量将为假。在本例下,布尔变量被设置为真。然后执行实际查找操作并且由查找调用返回的记录被高速缓存到一个变量中。最后,在启用测试时,查找操作的结果被添加到事件日志中。

[0171] 在处理相同记录期间的任一后续查找操作将调用相同的子程序。然而,对于后续子程序调用来说,布尔变量将变为真。这改变了子程序的行为,以使得可以返回先前读取并且高速缓存的数据而非进行多余的查找操作(并且避免生成额外的日志事件)。

[0172] 为简单起见,高速缓存只作用于单个关键字值。如果对相同的查找文件使用不同的关键字值(在相同的记录中)进行多个引用,那么那些查找结果中只有一个将被高速缓存。全部其它查找子程序调用将翻译成实际查找操作。也就是说,熟练的从业者将看到,如何能够通过使用用于经高速缓存的结果的哈希表而非简单变量,将该查找扩展为支持具有不同关键字的多个查找。

[0173] 参数

[0174] 规则集可以参照参数。在一些例子中,每个规则集具有一个可选参数表,就像变量表一样,可选参数表将参数的商务名称映射到它们的技术名称。参数表中的每个条目都具有下列特征:

[0175] 1. 商务名称。这是当参数将出现在规则正文中时它的名称。通常,无论输入变量在何处使用都能够使用参数。

[0176] 2. 技术名称。这是在开发环境中参数的名称。

[0177] 3. 参数的类型(字串、十进制、日期或者日期时间)。在生成的变换中,当需要时将参数转换为其它类型。

[0178] 4. 可选说明和注解。

[0179] 除了参数的值在整个输入文件上是常量,并且在处理开始时它们的值是外部指定的之外,参数与变量相仿。

[0180] 测试规则和日志记录

[0181] 生成变换的一部分包括测试它将对应的规则。规则也经认证,也就是说,经语法和语义一致性检查。与认证相对,测试包括规则的运行并且由用户确定正确性,例如通过提供

期望的输出或者将输出与期望值进行手工比较。

[0182] 系统支持两个级别的测试。如先前所述,每个规则可以具有一个关联的测试数据集,以值和期望结果的嵌入式表的形式。这被称为单元测试。在编辑一个规则时,可以针对测试数据的每行重新评估规则的输出。实际结果和期望结果之间的任一不匹配或者不能产生有效结果都将突出显示以待解决。

[0183] 在一些例子中,对使用标准机制的服务器过程可存取外部输入测试文件。使用外部文件的测试称为文件测试。测试文件具有与用于规则集的输入数据集匹配的记录格式。在一些例子中,可以提供替换的记录格式。选择性地,用户能够标识包含期望结果的数据集。系统针对测试数据集运行规则集并且显示产生了什么输出以及为什么生成。如果期望结果被包括,那么系统就将实际结果与期望结果进行比较并且列出所有不同的记录。在一些例子中,接口能够扩展以使用户能够增量地重新训练个别值。

[0184] 单元测试与文件测试之间的一些差别包括:

[0185] 1. 对于查找文件来说:在单元测试模式下,对于每个测试例(test case),每个查找变量的值被定义为该测试的一部分。不指定关键字;当测试运行时,对于每个测试例、对于每个查找变量假定相同的值。测试数据集包含多个测试例,每个测试例能够为每个查找变量指定不同的值。在文件测试模式下,使用真实的查找文件。这意味着不同的关键字将返回不同的值,但是它也意味着针对特定关键字的、用于任一给定查找变量的值在测试期间不能被改变。

[0186] 2. 对于内含规则集来说:在单元测试模式下,不运行内含规则集并且甚至不必要是完整的。而是,对于来自每个内含规则集的每个输出在测试数据集中指定一个值。在文件测试模式下,内含规则集以它们将在产品中运行的方式运行。这暗含着内含规则集所需要的任一查找文件或者参数也必须在测试时间指定。

[0187] 3. 对于参数来说:在单元测试模式下,可以对于针对每个测试例的每个参数设置不同的值。在文件测试模式下,每个参数的值对于整个测试都是常量。

[0188] 4. 对于当前日期来说:当测试时,如果该规则参照当前日期或者时间,用户指定应该为当前日期和时间假定的值。在单元测试模式下,日期与时间对于每个测试例可以是不同的。在文件测试模式下,针对整个测试设置单个日期和时间值(该值可以是与运行测试的时候机器的日期和时间不同的值)。

[0189] 5. 对于记录格式和映射来说:不需要对于单元测试指定映射;测试完全基于变量的商务名称完成。对于文件测试来说,所有变量被映射到技术名称并且指定用于输入、输出和查找的记录格式。

[0190] 通过定制的日志记录特征来测试和认证规则集,如图3中所示的流程图300中所示。接收一个或多个规则集和相应的测试数据作为输入(302)。生成器基于规则集生成(304)变换并且使用它计算测试数据中的每个记录的输出值。可选地,用户能够通过跟踪(trace)包括所生成的变换的、基于图的计算的运行(308),来配置(306)将生成的日志的特性。在基于图的计算中,相关组件的“log(日志)”端口用于测试输出。“log”端口是用于组件的带外通信信道。它是从变换得到附加输出元数据的一种方法,其不需要改变实际输出数据的记录格式。日志记录信息从日志端口输出(310)作为集中到日志文件中日志消息,该日志文件存储在运行期间可访问的存储系统中。可选地,可以过滤日志消息并且将其子集存

储在日志文件中。例如,因为输出记录未改变(例如,如果对于给定输入记录,没有规则例被触发),所以包括所生成的变换的组件可以使大多数的输入记录通过。可能期望只存储与已触发的规则例相应的日志消息,这些已触发的规则例改变输入记录的字段中的一个或多个值。图中的一个或多个组件可以过滤来自日志端口的记录以使得只有那些日志消息被写入日志文件中。

[0191] 针对测试生成的变换可以与将针对正常运行生成的变换稍有不同。例如,在正常运行下,使用变换的组件可以在给定输出端口生成给定类型的输出记录。在测试时间,与输出有关的测试被发送到日志端口,而输出记录保持不变。

[0192] 从第一测试例开始起,测试例的输入被输入变换中,并且输出连同哪个规则生成它的指示一起,被写入到输出阵列。该过程对每行重复直到评估了最后一行。然后,输出阵列可以用来生成如上所讨论的结果表。可以评估输出阵列以确定规则集是否有效。输出值可以包括在测试数据中,因为从一个测试生成的输出值可以与在前的测试中生成的值相比较。从输出阵列的第一行开始起,将生成的输出与来自测试数据或者在前的测试结果的期望输出相比较。如果任一输出不匹配,那么将该不匹配记入日志。对每一行重复该过程。在一些例子中,评估步骤被集成到输出-生成步骤中,并且每个所生成的输出都在它们被生成时与相应的期望输出进行比较。任一不匹配或者在处理测试数据结果过程中的其它错误都被记入日志。如上所述,一个规则集的输出可以是到另一个规则集的输入,在该情况下内含规则集被评估为包含规则集的一部分。

[0193] 用户可以通过输出字段,或者通过可能使用输入字段的表达式,限制被测试的规则。在一些例子中,用户可以在测试期间选择禁用某一规则。在一些例子中,用户不必等待整个测试文件被处理;一旦头几个记录得到输出,测试结果就可用。

[0194] 除了测试数据本身之外,可以跟踪(308)下列任一信息以便进行文件测试和日志记录:

[0195] 1. 输入数据集的物理位置(Physical Location)。其被存储在用于每个输入数据集的输入变量表中的规则集中。对于联合类型的数据集,需要所有物理位置。每当需要物理位置时,都可以使用数据库中的表名。

[0196] 2. 输入数据集的记录格式。缺省地,从针对输入数据集的数据集定义中取得该记录格式。输入变量中存在一个位置,用于用经核对而记入沙箱中的不同记录格式来推翻该记录格式。对于联合类型的变换,需要所有记录格式。

[0197] 3. 使用哪个部署。

[0198] 4. 所有查找文件的物理位置。被存储在查找文件表中。

[0199] 5. 用于每个查找文件的记录格式。从与每个查找文件关联的数据集定义中或者从推翻的记录格式文件取得。

[0200] 6. 每个参数的值。其将在测试参数会话中设置。

[0201] 7. 输出文件的物理位置。仅当完成回归(比较)(regression(comparison))测试时需要。它存储在输出变量表中。

[0202] 8. 输出文件的记录格式。再一次地,仅当完成回归测试时需要,并且从输出数据集定义或者从可选的推翻记录格式文件中取得。

[0203] 9. 项目沙箱的位置。必须在主机上的沙箱之外完成测试。沙箱应该是包含规则集

的项目的经核对的副本。所有的记录格式文件将从沙箱中取得。

[0204] 10. 当一个规则参照“now (现在)”、“today (今天)”或者类似的值时用于日期和时间的值。

[0205] 在该例子中,缺省地,变换不将单元格状态写入日志,然而,该功能可以在用户接口400中被激活,如图4中所示。也就是说,允许用户配置(306)日志特征,诸如打开和关闭各种特定准则的日志记录。接口400包括用于指定日志消息(称作“日志记录”)何时被创建(包括针对每个输入记录,何时发生错误或者指定的表达式何时为真)的部分410,和用于指定每个日志消息中包括些什么(例如,规则例激发、输入值、输出值、查找值、包括输出值表格的规则集、参数值和/或评估每个单元格的结果)的部分420。日志记录输入、输出等等使运行变慢但仅仅是很少量的。将单元格状态记入日志显著地减慢运行,或许减慢如量级阶次(order of magnitude)那么多。将执行测试并且依照指定结构显示检索到的结果。

[0206] 当不测试和/或进行日志记录时,生成器仍然可以生成将输入、输出等等记入日志的变换,并且在产生过程中使用那个变换。增强的变换生成相同的输出并且生成可以被分析、后运行的一系列日志消息以确定运行了哪些规则。如果用户将日志文件中生成的日志消息保存(310),那么,即使没有使用任何测试输入,也可以在产生过程中重放规则集的运行这一事实之后使用BRE。该运行被称作回放并且它对审核有用。日志文件包含输入、输出、查找、触发的规则例等等的细节,如所讨论的那样。它还具有一个头,其准确地记录哪个版本的规则集(名称和版本)被用来生成创建日志文件的变换。当正在产生过程中运行时,客户应当将日志端口的输出保存为一个文件(可选地,压缩文件)。为了回放一个会话,以供审核,客户将BRE投入应用然后开始回放会话。客户标识包含会话日志的文件。然后BRE读取头,打开所指示的规则集和版本,然后处理其余的日志文件,好像它在文件或单元测试模式下运行一样。在回放期间的显示与用户在运行文件或单元测试之后看到的显示相同,除了诸如以下几点之外:(1)没有相对于回归数据的比较;(2)像单元格状态那样的一些信息或许将从日志中遗失,因此它将不被示出,和(3)因为正在示出的规则集的版本可能不是当前版本,所以在回放期间规则集将是只读的。

[0207] 当启用日志记录功能时,生成的变换被修改为包括日志记录声明。可以通过调用write_to_log()生成日志消息,其是一个输出任意字符串到日志端口的函数。当使用write_to_log时,指定格式的数据被写入日志端口。例如,示范性日志端口格式如下

```
[0208] record
[0209] string('|') node;
[0210] string('|') timestamp;
[0211] string('|') component;
[0212] string('|') subcomponent;
[0213] string('|') event_type;
[0214] string('|\\n') event_text;
[0215] end;
```

[0216] 日志记录信息全部存储在event_text字段,其包含在对write_to_log()的调用中指定的字符串。日志端口记录中的其它字段被自动地通过组件填充,并且被BRE忽略。

[0217] event_text(专用于日志消息)的格式的例子如下:

```
[0218] record
[0219] decimal('|') count;//事件数目
[0220] record
[0221] decimal('|') recNum;//记录编号,1是第一个记录,0是图例(legend)
[0222] string(1) opCode;//见下
[0223] decimal('|') rule set;//哪个内含规则集,0是主规则集
[0224] decimal('|') group;//哪个输入组,哪个规则,等等
[0225] decimal('|') index;//输入索引,输出索引,规则例,等等
[0226] string(1) nullFlag;//空白或者用于空值的星号("-")
[0227] string('\x01') value;//输入值,输出值,等等
[0228] end events[count];
[0229] end;
[0230] 下面是示范性操作码(opCode):
[0231] "I" is an input variable.//输入变量
[0232] "O" is an output variable.//输出变量
[0233] "K" is a lookup key.//查找关键字
[0234] "L" is a lookup variable.//查找变量
[0235] "R" is a triggered rule case,for non-legend records.For legend records
it is a rule.//被触发的规则例,用于非图例记录。对于图例记录,其是一个规则
[0236] "C" is a rule cell;it is only used when testing-level tracing is
enabled,except for the legend where it is always used.//规则单元,其仅在测试级
跟踪被启用时使用,除了一直使用它的图例记录之外。
[0237] "P" is a parameter.//参数
[0238] "H" is a column heading.//列标题
[0239] "N" is a rule set name;it is only used in legend records.//规则集名称,
仅在图例记录中使用
[0240] "V" is a version number;it is only used in legend records.//版本号,仅
在图例记录中使用
[0241] "D" is a difference record;it is only used when testing.//差别记录,仅
在测试时使用
[0242] 在该例子中,每个输入记录将有一个日志消息。日志消息将包含一串事件。对于每
个输入记录的一大串事件,将对每个输入、每个输出、每个查找字段、每个运行的规则例等
等制作一个事件。
[0243] 在本例中,日志事件中的值字段不是二进制的。其不包含专用字符hex01或换行
符,因为它们将不正确地触发事件或日志记录之间的分隔符。而是,所有值都被转换为可印
刷的字串。不可打印的字符被转换为十六进制字串(例如,换行符->"\x0a")。二进制变量被
转换为十进制,等等。
[0244] 变换中的任何优化(像使用内部查找文件一样)在必要时也都可以被禁用,以确保
准确地日志记录。
```


[0245] 当看到第一记录时,日志消息被创建,其包含相应规则集的细节。这就是所谓的“图例(legend)”日志消息,并且对于给定规则集每个图运行就生成其一次。图例日志消息的第一部分包括日志格式版本以及规则集位置和版本号(回放所需)。其后将是将各个输入、输出、规则、查找文件、查找字段、参数等等的名称记入文件的信息。这些名称与索引关联(1是第一输入变量,2是第二输入变量等等)。这使得后续与事件关联的“跟踪”日志消息通过索引而非名称来引用变量和规则,以节省日志文件中的空白。

[0246] 在图例日志消息写入到日志文件之后,对于由正在记入日志的图处理的数据流的第一记录以及每个后续记录来说,任何后续日志记录事件都可能发生,每个都与相应的跟踪日志消息关联。

[0247] (1) 如果输入记录被记入日志,那么就对于针对每个输入记录将那些输入变量的值记入文件的每个输入变量创建一个日志消息。

[0248] (2) 如果参数被记入日志,那么就对于将这些参数的值记入文件的每个输入变量创建一个日志消息。

[0249] (3) 如果存在内含规则集,那么运行内含规则集中的规则。那些规则将依据该逻辑递归地生成日志消息。

[0250] (4) 如果例状态被记入日志,那么就计算每个规则中对于每个例的值并且记入日志。

[0251] (5) 使用链在一起的if-then-else(如果一则一否则)逻辑或switch(跳转)声明或者内部查找来运行实际的规则逻辑。一旦得知哪个规则例触发,就针对触发的规则例创建日志消息。

[0252] (6) 同样,一旦得知哪个规则例触发,就将值写入规则的输出变量并且同时,针对将分配的值记入文件的每个输出(例如,经变换的记录中的值)创建日志消息。

[0253] (7) 在评估这些规则由子例程操控的同时查找参考完成。子例程逻辑将执行查找,然后针对从查找文件读取的每个变量创建日志消息,所述变量将用作关键字的值和找到的所有查找变量的值记入文件。然后,子例程将返回查找值到主逻辑。查找子例程将保留一个表示其是否已经被调用的布尔变量,以避免对于同一个关键字生成重复的日志事件。

[0254] (8) 如果没有规则例触发,那么就将运行最后的else分句(或者用于跳转声明的缺省分句)以分配缺省输出值。同时,将针对将分配的值记入文件的每个输出创建日志消息。

[0255] 可以通过各种方法使用从日志端口提供的日志记录信息。例如,在某些情况下,对于给定场景(scenario),与用作测试数据集的数据集关联的输入记录的数目可以大于所期望的数目,诸如在测试某个图中的数据处理组件的规则集中使用的测试数据集。来自日志端口的日志记录信息可以在图的初始运行期间由图的组件处理。这些组件可以检查日志及记录信息以确定对于规则集的规则中的每个规则例,提供至少一个跟踪日志消息所需要的输入记录的最小集。例如,这些组件识别第一输入记录,在每个规则中的各个例针对该第一输入记录被触发。然后将那些被标识的记录与缩减的测试数据集相关联地存储,在该缩减的测试数据集中已经过滤出所有其它输入记录。经缩减的测试数据集仍然使规则集中相同规则和规则例的能够测试,但是可以小得多并且更加有效地用于测试。

[0256] 交叠分析

[0257] 在一些例子中,如上所述,当用户相对于规则集运行测试数据集时,可以跟踪每个

被触发的规则,也就是说,如果较高优先级的规则例的未使其所有输入条件得到满足,则使其所有输入条件都得到满足并且已经产生输出的规则例。在处理了测试数据之后,系统可以对测试输出数据进行后处理并且生成未由任何一个测试例触发的每个规则或者规则例的列表。该信息可以叠加在编辑器中规则的显示上以快速地示出哪些规则被触发或者未被触发。通过该信息,用户能够查找被其它规则模糊掉的可能规则,也就是说,重叠的规则。也可以针对各个规则例示出计数。计数可能只是在得知规则例是否被触发上有用,特别是对于调节值以实现输出的期望分布以及对于识别对调整性能最可能的规则例。

[0258] 变换生成过程

[0259] 通过将各个规则集转换为变换来在应用(图)中评估商务规则。然后,变换附着于图中的某一组件。这些组件可以是包含以特定方式链接以便运行变换的一组标准组件的子图。然后,这些子图可以通过附加组件被使用,例如,对于联合和上卷使用关键字。

[0260] 可以以多种方式从商务规则中生成变换码。在变换的内部没有被设计为用户编辑型的例子中,生成过程可能导致难以理解但是比逐个应用规则更加有效地实现规则的变换。在某些情况下,专用的查找文件或者其它技术可以用于提高所生成的变换的性能。如何生成变换的一些细节可以存储在一个部署中。可以在多个图中使用的规则集可以具有针对其不同的可能用户的多个部署。规则集也可以包含规则超集(super set of rules)——仅其中的一些为各个部署所需要——其具有识别当生成变换时使用哪些规则的部署。如果规则具有许多常数值,极少量表达式(如果有的话),那么不使用if then else逻辑,而可以使用查找表。在这种情况下,查找表是规则的一部分(没有分别地保存)。例如,考虑该规则:

[0261]

| 出发城市 | 目的地城市 | 服务等级 | 表达式 | 经常飞行英里数 |
|------|-------|------|-------------------------------|---------|
| BOS | LAX | 头等 | | 6000 |
| BOS | LAX | 商务 | | 3000 |
| BOS | LAX | 二等 | 使用英里数 (Is Using Miles) = 是? | 0 |
| BOS | LAX | 二等 | 其它 | 3000 |
| BOS | CHI | 头等 | | 2000 |
| BOS | CHI | 二等 | | 1000 |
| BOS | NYC | 头等 | | 1500 |
| BOS | NYC | 商务 | | 1000 |
| BOS | NYC | 二等 | | 500 |

[0262] 通过建立一个具有下列信息的存储器内查找表来操控该规则:

[0263]

| 从 | 到 | 等级 | 表达式 | 英里数 |
|---|---|----|-----|-----|
|---|---|----|-----|-----|

| | | | | |
|-----|-----|---|---|------|
| BOS | LAX | 1 | 0 | 6000 |
| BOS | LAX | 2 | 0 | 3000 |
| BOS | LAX | 3 | 1 | 3000 |
| BOS | CHI | 1 | 0 | 2000 |
| BOS | CHI | 2 | 0 | 1000 |
| BOS | NYC | 1 | 0 | 1500 |
| BOS | NYC | 2 | 0 | 1000 |
| BOS | NYC | 3 | 0 | 500 |

[0264] 然后,变换大约像这样:

```
[0265] int expr=lookup(from,dest,class).expr
[0266] int miles=lookup(from,dest,class).miles
[0267] if(expr==1 and is_using_miles)miles=0
[0268] return miles;
```

[0269] 每个规则集都具有部署表,其被适配为将名称映射到有关该部署的配置的细节。参考图4,希望将特定运行信息记入日志的用户能够根据每个规则集的部署表中定义的属性将每个条目输入到图形用户接口中:

[0270] 1. 部署名称。一个在规则集内必须是唯一的任意字串。

[0271] 2. 输入数据集名称。如果在输入变量表中列出多个输入数据集,那么部署表中的每个条目都指示哪个输入数据集用于该部署。

[0272] 3. 输出数据集名称。如果在输出变量表中列出多个输出数据集,那么部署表中的每个条目都指示哪个输出数据集用于该部署。

[0273] 4. 用于每个内含规则集的部署名称。对于每个内含规则集来说,我们需要指示哪个部署应当用于包含规则集的各个相应部署。

[0274] 5. 将要生成的组件和变换文件的目标位置。

[0275] 在一些例子中,总存在有至少一个部署,被命名为缺省部署。它是一个在没有指定其它部署时使用的部署。

[0276] 这里,在一个示范性实施例中,是变换生成的基本过程。首先,对于规则集中计算的输出生成多个规则。将通过变换中的通配符规则来处理所有其它输出。通常,仅在内部使用的输出变量导致将在所生成的变换中创建局部变量。也就是说,根据需要,生成的变换可以包括更多局部变量,以便避免重复的计算(例如,如果优化是针对速度优先于空间)。

[0277] 根据变换类型存在一些变换差异:

[0278] • 重新格式化——输入被称作“in0”,输入变量可以具有像“in.field”这样的技术名称。输出被称作“out”,输出变量可以具有像“out.field”这样的技术名称。

[0279] • 联合——两个输入被称作“in0”和“in1”。输出被称作“out”以及通配符规则假定in0被复制到“out”。当生成组件时可以设置所有参数。规则集将具有多个输入集,一个输入集针对用于联合的各个输入。规则集也将指定联合类型、输入是否应当被重复数据删除(de-dupe)、以及被用作用于联合的关键字的字段的商务名称(在一些例子中,其必须存在于各个输入集中)。而且,可以允许用户指定针对被用作输入过滤器的每个输入的表达式。

[0280] • 上卷——输入被称作“in0”,输出被称作“out”。在上卷类型的规则集的情况下,

允许用户使用聚集函数(在其它变换类型中不被支持)。如果用户创建一个技术名称是“input_select”或者“output_select”的输出变量,那么“input_select”和/或“output_select”函数就通过计算那些输出的规则的逻辑被添加到变换。两个函数的输入都被称作“in0”(即使output_select通常将其参数命名为“out”)。像在联合类型中一样,可以在生成组件时设置所有参数。

[0281] • 过滤——两个预定义的常量中的一个被输出。用于过滤类型(Filter-type)的变换的唯一输出变量是类型组件“选择”,其是非零的且非空的以通过输出。在一些例子中,其被实现为子图中的重新格式化组件。

[0282] 也可以实现附加的变换类型:

[0283] • 扫描——对于扫描类型的规则集来说,用户能够指定在记录之间被记住的一个或多个输出的值。将针对每个记录正常计算这些输出的值。然而,将针对那些包含来自最后一个记录的输出的值的输出中的每一个创建附加的内置输入。这使得用户能够例如,计算跨越多个记录之上的字段的总和——通过将部分和存储在输出变量中,该输出变量的状态随后将可用作下一个记录中的输入。

[0284] 此外,对于扫描类型(scan-type)的规则集,用户能够指定可选关键字。关键字是用于成组记录的一个或多个字段。当针对扫描类型的规则集指定关键字时,记录之间记住的所有输出的状态对于关键字的每个唯一值都是不同的。例如,如果关键字是一个客户编号,并且一个输出用来计算每个客户的所有交易的总和,那么在多个范围(cover)下,将对每个唯一的客户编号保存一个部分和,因此能够针对每个客户计算不同的总和。

[0285] • 分类——一个规则具有N个输出并且变换决定对于每个记录应当使用哪个输出。对于该组件来说,系统创建一个out::classify(in)函数。输出是单个整数值,表示应当使用哪个输出端口(可以存在多于一个)。输出为0意味着零端口,输出为1意味着第一端口,等等。

[0286] 用于分类类型(Classification-type)的变换的唯一输出变量是类型组件“选择”,其将是输出端口的索引(基于零的)。这类似于一个过滤器,除了针对输出是使用N个值而非两个值之外。

[0287] • 函数——函数类型(function-type)的规则集可以被转化为变换文件,然而并非作为组件变换。而是,当函数类型的规则集被转化为变换时,所构造的变换文件被设计为包括在其它变换中。每个输出变量被转化为一个函数。那些函数的输入取决于规则的类型。针对函数类型的规则,输入是针对以列在表中的次序的规则输入。针对非函数类型的规则,每个输出函数取得一个输入,命名为,具有与输入变量相应的所有字段的记录。

[0288] 当规则集用来生成变换——其将依次成为图的一部分——时,图组件包括规则集和部署的名称。然后,图的开发者可以编辑规则集而非组件中所生成的变换。对规则集的改变将导致重新生成变换。在一些例子中,用户能够移双击(shift-double click)由BRE生成的组件。移双击导致将BRE投入使用的图形开发环境(GDE),通过EME名称、规则集名称和部署。在一个例子中,命令行接口可用于将每个BRE过程投入使用;然而,也可以采用不同的内处理通信机制。

[0289] 可以使用在计算机系统上运行的软件实现上述的日志记录方法。例如,在运行在经编程或可编程计算机系统(可以具有各种架构,诸如分布式、客户机/服务器或者网格式)

上的一个或多个计算机程序中的软件形式的过程,所述计算机系统每个包括至少一个处理器、至少一个数据存储系统(包括易失和非易失性存储器和/或存储元件)、至少一个输入设备或端口以及至少一个输出装置或端口。软件可以形成更大程序的一个或多个模块,例如,提供与计算图的设计和配置有关的其它业务。图的节点和元素可以实现为存储在计算机可读介质中的数据结构或者符合存储在数据库中的数据模型的其它结构化数据。

[0290] 软件可以提供在诸如CD-ROM这样的存储介质上,可由通用或专用可编程计算机读取或者通过网络通信介质传递(以传播信号形式编码)到运行其的计算机。可以在专用计算机上或者使用诸如微处理机这样的专用硬件执行所有这些功能。软件可以以分布式实现,其中由软件指定的不同计算部分由不同的计算机执行。每个这样的计算机程序都优选地存储或下载在可由通用或专用可编程计算机读取的存储介质或器件中(例如,固态存储器或介质,或者磁介质或光介质),以供计算机系统读取存储介质或器件来执行其中所描述的过程时对计算机配置和操作。也可以考虑将本发明的系统实现为计算机可读存储介质,被配置为具有计算机程序,其中存储介质被配置为使得计算机系统以专门的预定义方式操作以执行这里描述的功能。

[0291] 已经描述了本发明的许多实施例。然而,应当理解,在不脱离本发明的精神和范围的条件下可以进行各种修改。例如,上面所述的一些步骤可以独立地排序,因此可以以与所述次序不同的次序来执行。

[0292] 应当理解,前述说明意在说明而非限制本发明的范围,本发明的范围由所附权利要求来限定。例如,上述的许多功能步骤可以以不同的次序来执行而基本上不影响总体处理。其它实施例在后面的权利要求的范围内。

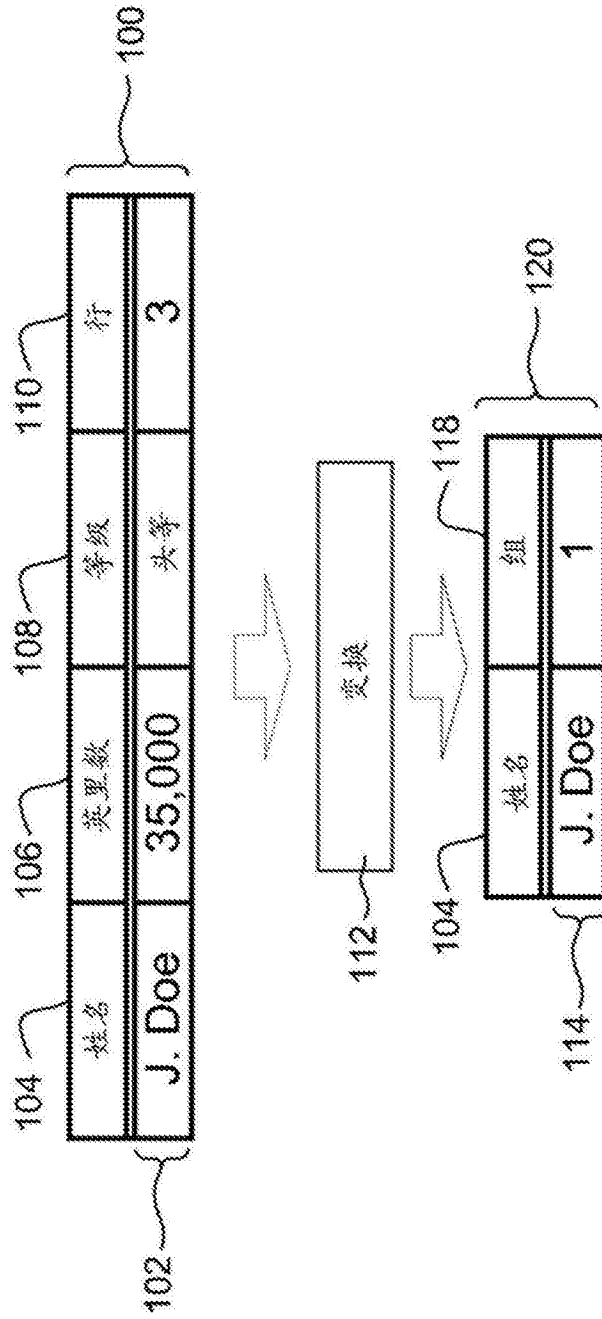


图1A

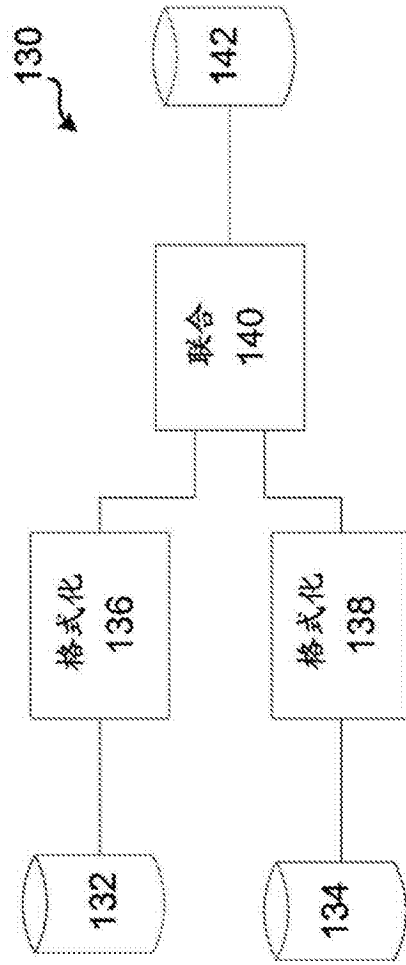


图1B

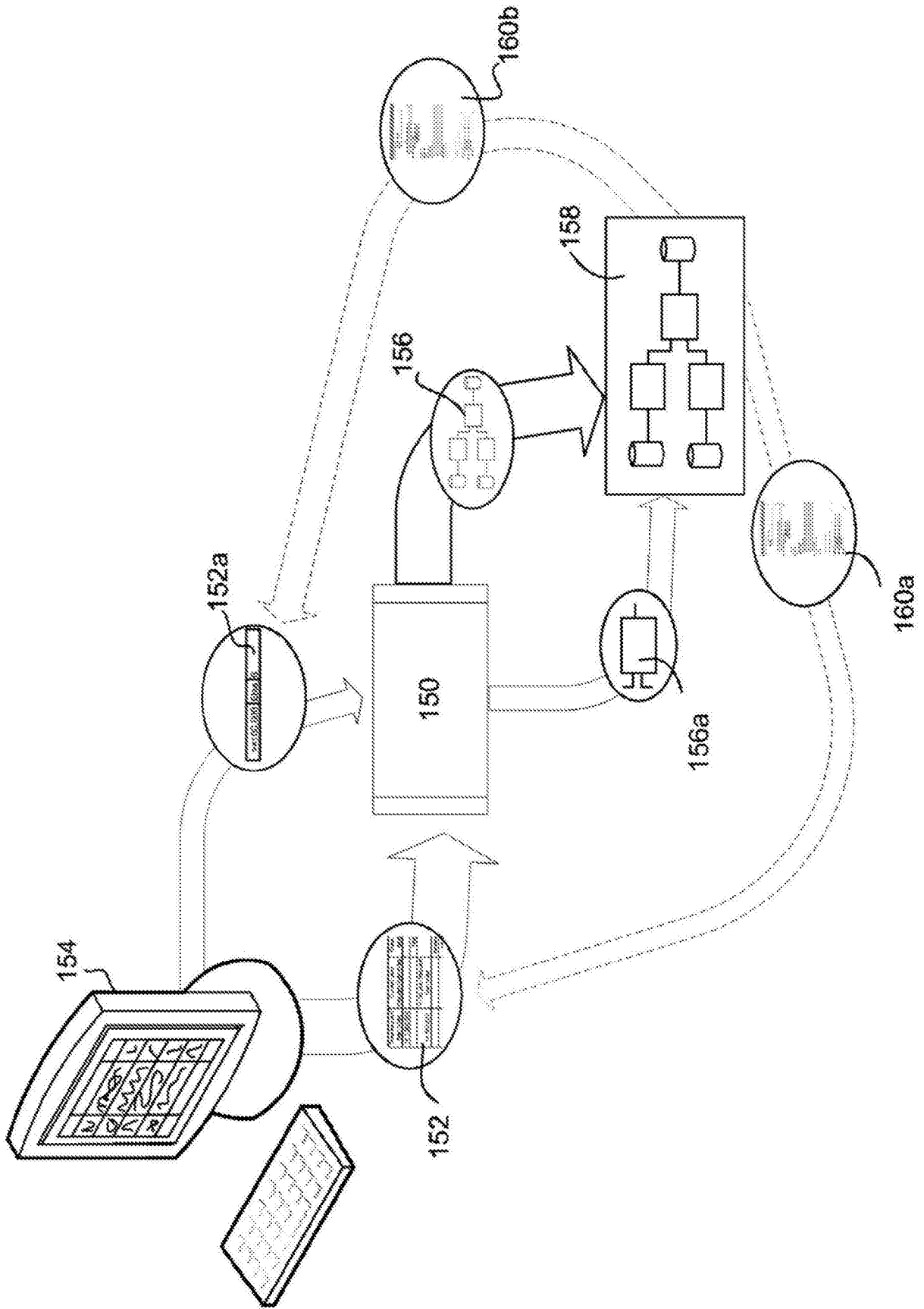


图1C

| | 202 经常飞行 总英里数 | 204 当年经常飞行 英里数 | 206 座位等级 | 208 座位行 | 212 乘机组 |
|------|---------------------|----------------------|-------------|------------|------------|
| 210a | $\geq 1,000,000$ | | | | 1 |
| 210b | | | 头等 | | 1 |
| 210c | $\geq 100,000$ | \geq 去年经常飞行 英里数 | | | 2 |
| 210d | | | 商务 | | 2 |
| 210e | | | 其它 | ≤ 10 | 2 |
| 210f | | | 相同的 | ≤ 40 | 3 |
| 210g | | | 相同的 | ≤ 50 | 4 |
| 210h | | | 相同的 | 其它 | 5 |

200

图2A

| 220a | 表达式 | 乘机组 |
|------|------------------------------|-----|
| | 经常飞行总英里数 $\geq 1,000,000$ | 1 |

| 220b | 表达式 | 乘机组 |
|------|------------|-----|
| | 座位等级 == 头等 | 1 |

| 220c | 表达式 | 乘机组 |
|------|--|-----|
| | 经常飞行总英里数 $\geq 1,000,000$ 且当年经常飞行英里数 \geq 去年经常 飞行英里数 | 2 |

| 220d | 表达式 | 乘机组 |
|------|------------|-----|
| | 座位等级 == 商务 | 2 |

| 220e | 座位行 | 乘机组 |
|------|-----------|-----|
| | ≤ 10 | 2 |
| | ≤ 40 | 3 |
| | ≤ 50 | 4 |
| | 其它 | 5 |

250

图2B

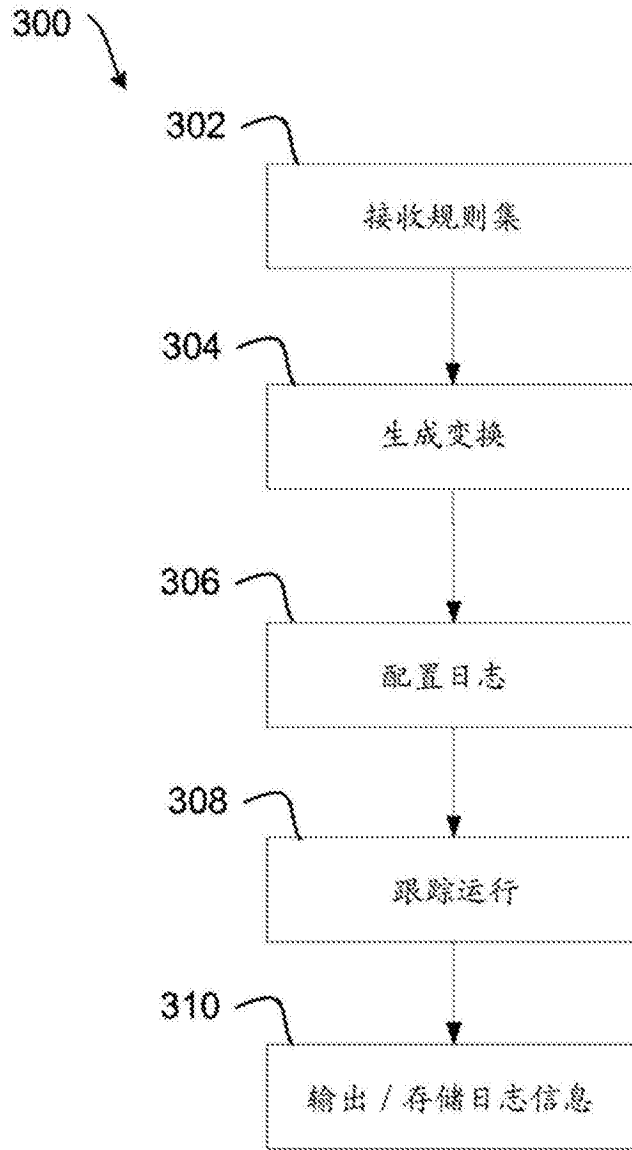


图3

400

Deployment – “Default deployment”

GENERATE COMPONENT SETTINGS

Name:

Description:

Comment:

Transform EME Path: ...

Component EME Path: ...

Target Co>Operating Systems®: ▾

Datasets:

| | Type | Name | Dataset |
|---|--------|---------|-----------|
| 1 | Input | Input 1 | Dataset 1 |
| 2 | Output | Output | Dataset 1 |

LOGGING SETTINGS

Create a log record:

Never

For each input record

When an error occurs

When this expression is True:

Include in each log record:

Rule case firings (recommended)

Input values

Output values

Lookup values

Output values from included rulesets

Parameter values

Result of evaluating each cell (not recommended)

LOGGING SETTINGS

Input Data:

| | Name | KInd | Location |
|---|---------|------|----------|
| 1 | Input 1 | File | |

Baseline Data:

| | Name | KInd | Location |
|---|----------|------|----------|
| 1 | <Output> | File | |

Sandbox: ...

410

420

图4