

基于与或逻辑的界面关系模型表示方法

梁伟晟 李 磊

(中山大学软件研究所 广州 510275)

摘 要 作为用户与系统直接交互的途径,界面设计的好坏将影响用户对系统的第一印象,因此界面设计在软件工程设计中十分重要。传统有限状态机在表达复杂的界面状态关系时存在困难。我们引入与或图表示方法来表达初始界面到最终界面的业务流程问题求解过程,建立基于与或逻辑的界面关系模型,并给出模型的逻辑表示,讨论了利用逻辑推理和演绎实现模型语法和语义检查的方法。

关键词 界面关系模型,与或图,有限状态机

Representation of User Interface Relation Model Based on And/Or Logic

LIANG Wei-Sheng LI Lei

(Software Institute, Zhongshan University, Guangzhou 510275)

Abstract As a way contacting with user, user interface influences the appraisalment of user to system. So the user interface design is important to system in software engineering. FSM encounters difficulty in express complicated user interface relation. We introduce AND/OR graph to express solution of business flow problem which works from initial interface to final interface. We build the user interface relation model based on AND/OR logic, introduce the logic representation of model, discuss the way of syntax and semantic validation by reasoning and deduction.

Keywords User-interface relation model, And/Or graph, FSM

1 研究动机

在业务系统的开发中,界面作为用户与系统直接交互的途径,界面设计的好坏将影响用户对系统的第一印象,从而影响到用户对系统的评价。研究统计表明^[1],平均 48% 的系统设计开发工作量在用户界面部分。可视化的界面设计工具较好地支持了界面布局设计,但却不能很好地支持界面之间的交互操作。在业务系统中,界面之间的交互操作顺序蕴含了业务流程,因此界面关系模型的设计就显得尤其重要了。

界面关系模型描述业务系统的界面之间的关系。所有软件都可以使用有限状态机来进行描述和建模^[2],将界面看成系统所处的状态。当用户或者系统操作触发界面跳转时,系统状态发生改变,因此界面关系模型可以用有限状态机来描述。文[3]给出了基于有限状态机的界面关系模型 $R = \{s_0, S, TS, F\}$, 其中 s_0 是初始界面, F 是终止界面集, S 是界面集合, $TS: S \Rightarrow S$ 是转换集合,表示业务操作引起界面之间调用和跳转。不仅如此,基于有限状态机的界面模型建模还可以用于产生测试用例,从而验证界面的正确性。在此方法中,有限状态机的一条路径表示一个测试用例,其优点是有限状态机一旦建立,测试用例就可以自动生成。文[3]描述了一种基于有限状态机界面关系模型的反向推导验证方法,文[4]描述了一种基于扩展有限状态机测试的测试数据自动选取方法。然而,由于界面输入和操作的复杂性,有限状态机表示的界面关系模型存在状态空间较大的问题,给模型的维护带来很大的工作量。当需求变化时,相应的模型调整需要界面设计人员具备较多的领域知识。对于一些复杂的界面关系,例如界面 I 的下一步操作,必须既调用界面 I_1 又调用界面 I_2 ; 又例如界面 I 的下一步操作只能选择界面 I_1 和 I_2 之一,使

用有限状态机显式地表达这些关系是比较困难的。另外,出于优化业务流程的需要,要对界面状态进行裁剪和合并,有限状态机在这方面也存在局限性。

与或图是人工智能中用于问题规约和求解的常用方法^[6],将复杂问题逐步细化分解为简单的子问题。在业务系统中,业务处理流程是从某个界面开始,经过一系列操作步骤以及相应的界面转换,最终在某个界面结束。从某种意义上说,初始界面开始到最终界面结束的过程实际上是业务流程的问题求解过程,因此可以采用类似与或图的表示形式去扩展基于有限状态机的界面关系模型。同时,利用与、或这两种基本逻辑关系组合,可以表达复杂的界面关系。结合商品订购系统的例子,本文首先讨论了业务系统界面之间的与或逻辑关系,然后给出了基于与或关系的界面关系模型以及模型的一些性质,最后给出模型的逻辑表示,并且探讨了利用逻辑推理和演绎实现模型语法和语义检查的方法。

2 界面与或关系

在一个商品订购系统中,客户选择商品,填写商品订购单后,系统要形成商品发货单以及购物发票。在这个业务流程中,商品发货单和发票单据界面是必须都被执行的,这种界面之间的关系是“与”关系。客户下订单之后,他可以选择银行划账支付费用,也可以选择商品到货之后付款,前者需要调用界面填写银行账号信息,后者则需要填写付款人联系地址和联系电话。在这个业务流程中,填写银行账号信息的界面和填写付款人联系方式的界面是选择性执行的,这种界面之间的关系是“或”关系。采用类似与或图的形式,可以表示为图 1 和图 2。

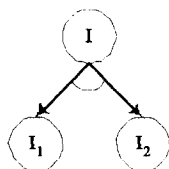


图1 “与”关系

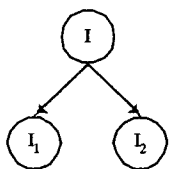


图2 “或”关系

其中 I 是业务流程中 I_1 和 I_2 的前一个界面。在图 1 中, I_1 和 I_2 之间是“与”关系;在图 2 中, I_1 和 I_2 之间是“或”关系, 并且称 I 是“与(或)”节点。对于 I 只有一个后继界面的情形, 仍然称 I 是“与”节点。

3 界面关系模型

定义 1 界面关系模型 $R = \{s_0, S, F, TS, A, O\}$, 其中 s_0 是初始界面状态, F 是终止界面状态集, S 是界面状态集合, $s_0 \in S, F \subseteq S$; $TS: S \rightarrow S$ 是状态转换集合, 表示业务操作引起界面之间调用和跳转, 用连接状态节点的有向边表示, 有向边的权值记为 $cost(TS)$; A 是具有与关系的状态节点组合的集合; O 是具有或关系的状态节点组合的集合。

界面关系模型是扩展的有限状态机, 其表示形式为包含与或逻辑关系的有向图。

定义 2 界面关系模型 $R = \{s_0, S, F, TS, O, A\}$ 和界面关系模型 $R' = \{s'_0, S', F', TS', O', A'\}$, 如果

- 1) $S' \subseteq S, F' \subseteq F$.
- 2) $TS' \subseteq TS$, 并且 $\forall (s'_1 \Rightarrow s'_2) \in TS'$, 有 $s'_1 \in S', s'_2 \in S'$, 即 TS' 中的边所关联的节点都在 S' 中。
- 3) $\forall a' \in A', \exists a \in A$, 使得 $a' = a$, 并且 $\forall o' \in O', \exists o \in O$, 使得 $o' \subseteq o$, 即对于与关系, A' 必须包含 A 中对应的整个状态节点组合, 对于或关系, O' 可以是 O 中对应的状态节点组合的一部分。
- 4) $\forall s' \in S', \forall t \in \{t | (s' \Rightarrow t) \in TS\}$, 若 $\exists a' \in A'$, 使得 $t \in a'$, 则 $(s' \Rightarrow t) \in TS'$; 若 $\exists o' \in O'$, 使得 $t \in o'$, 则 $(s' \Rightarrow t) \in TS'$, 即父节点和与或关系节点之间的边包含在 TS' 中则模型 R' 是模型 R 的子模型。

业务系统的界面关系模型蕴含了业务流程, 因此如果 R' 是 R 的子模型, R' 对应的业务流程集合是 R 对应的业务流程集合的子集。

界面关系模型 R' 是 R 的子模型, 如果 $\forall (s_1 \Rightarrow s_2) \in TS$, 有 $s_2 \notin S'$ 或者 $s_2 = s'_0$, 即在 R' 中只有 s'_0 与 R 的其余节点关联, 则 R' 是 R 的封闭子模型。

R' 是 R 的封闭子模型, 那么 R' 对应的业务流程是相对独立的业务子流程, 在系统设计的时候可以作为一个独立的模块。特别地, 如果 R' 是 R 的封闭子模型, 并且 $O' = \emptyset$, 即业务流程的每一步都是确定, 没有选择性执行步骤, 则 R' 是 R 的确定的封闭子模型。

在本文讨论的业务系统界面关系模型的意义下, 从初始界面状态 s_0 按照业务流程跳转到某个界面状态 s , s_0 到任意

一个 s 是可达的, 不考虑游离于业务流程之外的界面表示, s_0 是 R 的有向图表示形式的根。

假设从界面 s_1 跳转到界面 s_2 的成本为 $cost(s_1, s_2)$, 成本可以定义为业务人员的操作耗时、系统运行时间等。将 $cost(s_1, s_2)$ 作为有向边 $s_1 \Rightarrow s_2$ 的权值, 那么确定效率最优业务流程即是寻找业务流程的初始界面到终止界面的最优界面跳转路径, 算法如下。

输入: 界面关系模型 $R = \{s_0, S, F, TS, A, O\}$
 输出: s_0 到 F 终止界面的最优成本 (用节点的赋值表示)
 1) checked = $\{s_0\}$; unchecked = $S - \{s_0\}$; edge = $\{\}$; $s = s_0$
 2) MaxCost = $\text{Max}\{s_1, \text{cost}\}$, 对 s_0 的每一个子节点 s_1
 3) 对 $\forall s_1 \in S$, 执行 4) - 7)
 4) If s_1 是 s 的子节点 Then
 5) If s 是“与”节点 Then $s_1, \text{cost} = \text{MaxCost}$ Else $s_1, \text{cost} = \text{cost}(s, s_1)$;
 6) $s_1, \text{pre} = s$;
 7) Else $s_1, \text{cost} = +\infty$; $s_1, \text{pre} = \text{nil}$
 8) while checked $\neq S$ do, 执行 9) - 16)
 9) MinCost = $\text{Min}\{s_1, \text{cost}\}$; sel = $\{s_1 | \text{Min}(s_1, \text{cost})\}$, 对 $\forall s_1 \in \text{unchecked}$
 10) If sel.pre 是“与”节点 Then
 11) brother = $\{s_1 | \text{sel 的与关系兄弟节点 } s_1\}$; sel = sel + brother
 12) checked = checked + $\{sel\}$; unchecked = unchecked - $\{sel\}$;
 13) edge = edge + $\{s_1, \text{pre} \Rightarrow s_1\}$, 对 $\forall s_1 \in \text{sel}$
 14) 对 $\forall s_1 \in \text{sel}$ 并且 $\forall s_2 \in \text{unchecked}$, 执行 15) - 16)
 15) If $s_2, \text{cost} > s_1, \text{cost} + \text{cost}(s_1, s_2)$ then
 16) $s_2, \text{cost} = s_1, \text{cost} + \text{cost}(s_1, s_2)$; $s_2, \text{pre} = s_1$

4 界面关系模型的逻辑表示

界面关系模型中的与或关系可以用命题逻辑公式表示。对于前面讨论的节点与或关系, 用逻辑公式表示为:

与: $I \rightarrow I_1 \wedge I_2$

或: $I \rightarrow (I_1 \wedge \neg I_2) \vee (\neg I_1 \wedge I_2)$

对于 I 只有一个后继界面 I_1 的情形, 把 I 看作“与”节点, 表示为 $I \rightarrow I_1$ 。例如, 界面关系模型 $R = \{s_0, \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}, \{s_1, s_4, s_5, s_6, s_7\}, \{s_0 \Rightarrow s_1, s_0 \Rightarrow s_2, s_0 \Rightarrow s_3, s_2 \Rightarrow s_4, s_2 \Rightarrow s_5, s_3 \Rightarrow s_6, s_3 \Rightarrow s_7\}, \{(s_4, s_5)\}, \{(s_1, s_2, s_3), (s_6, s_7)\}\}$, 其逻辑表示为:

1) $s_0 \rightarrow (s_1 \wedge \neg s_2 \wedge \neg s_3) \vee (\neg s_1 \wedge s_2 \wedge \neg s_3) \vee (\neg s_1 \wedge \neg s_2 \wedge s_3)$

2) $s_2 \rightarrow s_4 \wedge s_5$

3) $s_3 \rightarrow (s_6 \wedge \neg s_7) \vee (\neg s_6 \wedge s_7)$

转换为范式:

$\{ \neg s_0 \vee s_1 \vee s_2 \vee s_3, \neg s_0 \vee \neg s_1 \vee \neg s_2, \neg s_0 \vee \neg s_1 \vee \neg s_3, \neg s_0 \vee \neg s_2 \vee \neg s_3, \neg s_2 \vee s_4, \neg s_2 \vee s_5, \neg s_3 \vee s_6 \vee s_7, \neg s_3 \vee \neg s_6 \vee \neg s_7 \}$

与或图常用于逻辑公式的归结, 通过与或逻辑关系的命题公式实现模型的逻辑表示, 可以利用演绎和推理等方法, 实现对模型的语法和语义自动检验。

4.1 游离界面检查

在实际应用意义下, 从初始界面 s_0 到任意一个界面 s 是可达的, 需要检查界面关系模型中是否存在游离于业务流程之外的界面表示。假设 S 是模型 R 的界面状态集合, G 是模型 R 逻辑表示的析取范式集合, 检查游离界面的算法为:

1) $D = \{\}$

2) 对 $\forall s \in S$

3) If 不存在 $g \in G, g \vee \neg s \equiv \text{True}$ Then $D = D + \{s\}$

算法执行结束, 如果模型存在游离界面, 则 D 中包含了这些游离界面。

4.2 与或关系一致性检查

s 是模型 $R = \{s_0, S, F, TS, A, O\}$ 中的界面状态, $s_1, s_2, \dots, s_n$

(下转第 210 页)

求解模型,并提出了基于启发式搜索的灵活规划算法框架。在算法提出的过程中,考虑到本算法的实际情况,对启发式代价做了相应的规定,同时对于启发式函数也做了相应改进。经过测试,该方法在处理很多灵活规划问题上都可以得到较好的效果,具有较高的效率。

本文的后续工作如下:进一步完善基于启发式搜索的灵活规划的相关算法,将该算法运用到实际的智能系统中。

参考文献

- 1 丁德路,姜云飞.智能规划及其应用的研究.计算机科学,2002,29(2):100~103
- 2 Blum A L, Furst M L. Fast Planning Through Planning Graph Analysis [C]. In: Proceeding of the International Joint Conference on Artificial Intelligence, 1995. 1636~1642
- 3 Kautz H, Selman B. Planning as Satisfiability [C]. In: Proceedings of the European Conference on Artificial Intelligence (ECA92). Vienna, Austria, August 1992
- 4 Bonet B, Geffer H. Planning as heuristic search: New results [C]. In: Biundo S, Fox M, eds. Recent Advances in AI Planning.

(上接第 204 页)

是 s 的子状态节点。如果 A 和 O 均存在 $(s_1, s_2 \dots s_n)$ 的组合,则 $s_1, s_2 \dots s_n$ 的与或关系是不一致的。

假设 $(s_i \dots s_j)$ 是 A 和 O 中均存在的组合,按照前面描述的逻辑表示,在析取范式集合 G 中必然存在 $\{\neg s \vee s_i, \dots, \neg s \vee s_j, \neg s \vee \neg s_i \vee \dots \vee \neg s_j\}$ 析取式子集,依据逻辑推理方法,

$s, \{\neg s \vee s_i, \dots, \neg s \vee s_j, \neg s \vee \neg s_i \vee \dots \vee \neg s_j\} \vdash s, \{\neg s \vee (s_i \wedge \dots \wedge s_j), \neg s \vee \neg s_i \vee \dots \vee \neg s_j\}$

$\vdash s, \{\neg s \vee (s_i \wedge \dots \wedge s_j \wedge (\neg s_i \vee \dots \vee \neg s_j))\}$

$\vdash s, \{\neg s \vee \text{false}\}$

$\vdash s \wedge \neg s$

$\vdash \text{false}$

由此,可以证明 $s_i, \dots, s_j$ 的与或关系是不一致的。

4.3 语义检验

G 是模型 R 逻辑表示的析取范式集合。假设 X 是用户界面需求相关的领域公理集合, E 是用户需求规定的允许的界面跳转流程示例逻辑表示集合, $\bar{E}$ 是用户需求规定的不允许的界面跳转流程示例逻辑表示集合,采用人工智能诊断的方法,如果

1) $x \in X, x \wedge G$ 是逻辑为真的

2) $\forall e \in E, e \wedge G$ 是逻辑为真的

3) $\forall \bar{e} \in \bar{E}, \neg \bar{e} \wedge G$ 是逻辑为真的

则 G 在 $X, E, \bar{E}$ 规定的语义环境下是正确的。

更进一步,将模型 R 的析取范式集合 G 改写为命题函数,即谓词形式的公式集合 H ,求解是否存在解释 I ,使得 H 是可满足的。如果 I 满足 H ,则 I 是界面调转流程的例。反过来说,如果 I 是用户需求规定允许的界面跳转流程的例, I 不满足 H ,则 H 在 I 定义的需求下是不正确的。由于当解释的域是有限集合时,一阶逻辑公式可满足问题等价于 SAT 问

5th European Conference on Planning (ECP'99) Durham, UK. Springer-Verlag, 1999

- 5 Miguel I, Shen Q, Jarvis P. Efficient Flexible Planning via Dynamic Flexible Constraint Satisfaction. Engineering Applications of Artificial Intelligence, 2001, 14(3): 301~327
- 6 Miguel I, Jarvis P, Shen Q. Flexible Graphplan. In: Proceedings of the Fourteenth European Conference on Artificial Intelligence, 2000. 506~510
- 7 Li Yang, Gu Wen-xiang, Yin Ming-hao, et al. Planning system based on heuristic. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 2005
- 8 谷文祥,李杨,殷明浩.带标记的反向搜索的规划识别算法.计算机科学,2004,31(10):243~244
- 9 彭祥云,姜云飞.基于资源约束和局部启发搜索的规划系统.控制工程,2006,13(2):185~189
- 10 杨宁,田蔚风,金志华.一种求解 DGPS 动态整周模糊度问题的交叉禁忌搜索.光学精密工程,2006,14(5):891~895
- 11 杨照华,祁振强,房建成.空中目标的免疫识别算法.光学精密工程,2006,14(5):922~927

题,因此可以利用 SAT 求解器对模型进行检验。

主要工作总结 通过引入与或逻辑,丰富了界面关系模型的表达能力,可以表达传统有限状态机所不能表示的复杂界面关系。我们的主要工作是,与用户需求的业务流程相结合,建立体现业务流程的界面关系模型;其次,在模型中引入与或逻辑关系,通过与或逻辑的组合表达实际应用中存在的复杂界面关系;第三,讨论了模型的性质,给出求解效率最优业务流程的算法;另外,给出模型的逻辑表示,利用命题逻辑的推理和演绎,探讨了模型的语法和语义检查问题。

参考文献

- 1 Schlunbaum E. Model-based User Interface Software Tools: Current state of declarative models[R]. GIT-GVU-96-30, Graphics, Visualization & Usability Center, Georgia Institute of Technology, Atlanta, November 1996
- 2 邵维忠,杨芙清.面向对象的系统分析[M].清华大学出版社、广西科学技术出版社,1998
- 3 梁伟晨,李磊.基于表单的业务系统界面逻辑模型获取的研究[J].计算机工程,2007,33(5):56~68
- 4 张涌,钱乐秋,王渊峰.基于扩展有限状态机测试中测试输入数据自动选取的研究[J].计算机学报,2003,26(10):1295~1303
- 5 Memon A M. A Comprehensive Framework for Testing Graphical User Interfaces[D]:[doctoral thesis]. the Graduate Faculty of Arts and Sciences, University of Pittsburgh, 2001
- 6 Nilsson N J. Principles of Artificial Intelligence[M]. North Holland: Tioga Publishing, 1980
- 7 Lee D, Yannakakis M. Principles and Methods of Testing Finite State Machines A survey[J]. In: Proc. of the IEEE, 1996, 84(8):1090~1123
- 8 李磊.机器学习在需求工程中的应用[C].见:第八届中国机器学习学术会议报告,2002