$$Mg-s=Ms\times Mg=\begin{bmatrix}1&0&0&0\\1&0&0&0\\0&1&1&0\\0&0&1&0\\0&0&1&0\\0&0&1&0\\0&0&0&1\\0&0&0&1\\0&0&0&1\end{bmatrix}\times\begin{bmatrix}1&0&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=\begin{bmatrix}1&0&0\\1&0&0\\1&1&0\\0&1&0\\0&1&0\\0&1&0\\0&0&1\\0&0&1\end{bmatrix} \tag{4}$$

$$Ms-a=Ma\times Ms=\begin{bmatrix}1&0&0&1&0&1&0&0\\0&1&1&0&1&0&0&0\end{bmatrix}\times\begin{bmatrix}1&0&0&0\\1&0&0&0\\0&1&1&0\\0&0&1&0\\0&0&1&0\\0&0&1&0\\0&0&0&1\\0&0&0&1\end{bmatrix}=\begin{bmatrix}1&0&2&0\\1&1&2&0\end{bmatrix} \tag{5}$$

$$Ma-c=Mc\times Ma\begin{bmatrix}1&0\\0&1\end{bmatrix}\times\begin{bmatrix}1&0&0&1&0&1&0&0\\0&1&1&0&1&0&0&0\end{bmatrix}=\begin{bmatrix}1&0&0&1&0&1&0&0\\0&1&1&0&1&0&0&0\end{bmatrix} \tag{6}$$

$$Mg-s-a-c=Mc\times Mg-s-a=\begin{bmatrix}1&0\\0&1\end{bmatrix}\times\begin{bmatrix}1&2&0\\2&2&0\end{bmatrix}=\begin{bmatrix}1&2&0\\2&2&0\end{bmatrix} \tag{7}$$

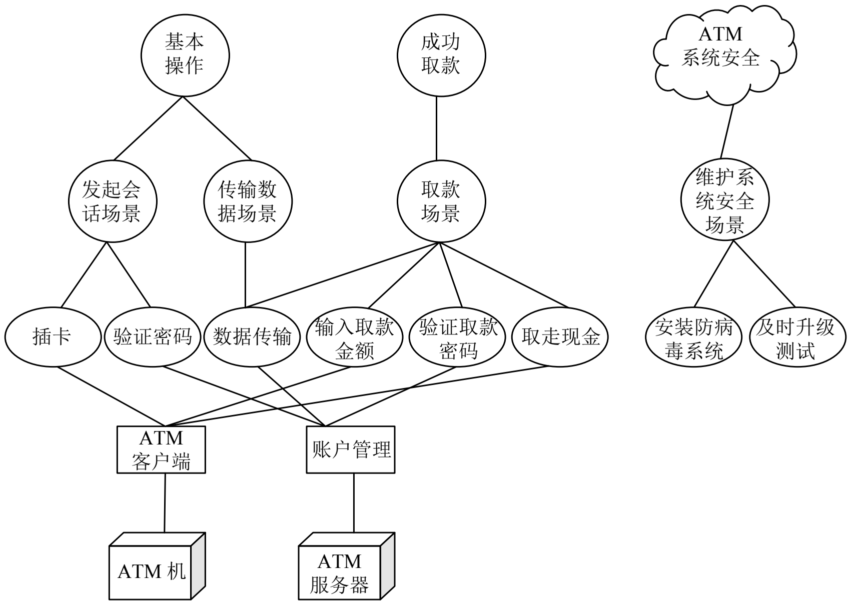

图 8　基于 ATM 系统的需求传播途径

以“成功取款”目标为例, 验证基于 ATM 系统的目标元模型层和场景元模型层间, 目标元素与动作元素追踪关系的准确性. 根据图 8 可知, “成功取款”目标更改影响到的动作是与“取款”场景相关的动作一致. 其中, “取款”场景由一系列“数据传输”、“输入取款金额”、“验证取款密码”、“取走现金”系统或人为的动作来完成. 这意味着 g2 与 a3、a4、a5、a6 间具有关联关系. 显然, 这与图 9 的第二列一致. 同理, 可知图 9 的其他列也是正确的.

	g1	g2	g3
s1	1	0	0
s2	1	0	0
s3	1	1	0
s4	0	1	0
s5	0	1	0
s6	0	1	0
s7	0	0	1
s8	0	0	0

图 9　ATM 系统中目标层与场景层间追踪关系

	g1	g2	g3
n1	1	2	0
n2	2	2	0

图 10　ATM 系统中目标层与软件体系结构层间追踪关系

再以“成功取款”目标为例, 验证基于 ATM 系统的目标元模型层和软件体系结构元模型层间, 目标元素和节点元素追踪关系的准确性. 根据图 8 可知, “成功取款”目标的更改影响到的构件是与“数据传输”、“输入取款金额”、“验证取款密码”以及“取走现金”动作相关的构件一致, 其中“数据传输”和“验证取款密码”与“账户管理”构件关联, “输入取款金额”和“取走现金”则与“ATM 客户端”构件关联, 而这两个构件又分别部署在“ATM 服务器”和“ATM 机”节点上. 这意味着 g2 与 n1、n2 间具有关联关系. 显然, 这与图 10 的第二列一致. 同理, 可知图 10 的其他列也是正确的.

因此, 通过矩阵运算, 在 ATM 系统多视点元模型间建立了可追踪关系, 一方面实现了在不同视点下元模型的统一, 另一方面可有效管理需求变更.

4　总结与展望

本文主要研究了一种在多视点的元模型间进行需求追踪的方法. 一方面把面向目标的 I*模型和软件体系结构模型引入到传统的基于 UML (Unified Modeling Language) 的面向对象的建模阶段中, 旨在设计一个基于多视点的建模过程框架. 合理的建模框架使我们对需求的理解更加完整. 另一方面在多视点间建立了追踪元模型, 并且通过矩阵运算实现了多视点下元模型间的追踪. 这不仅解决了视点分散性问题, 实现了多视点元模型间的统一, 还有效解决了需求变更困难的问题. 然而, 本文在场景元模型和软件体系结构元模型间主要针对功能需求进行了追踪并验证, 缺乏对非功能需求追踪的研究. 接下来的研究方向可从非功能需求在多视点元模型间的追踪继续展开.

参考文献

1 Booch G, Rumbaugh G, Jacobsen I. The unified modeling language user guide. Massachusetts: Addison-Wesley, 1999: 15–17.

2 France R, Bieman JM. Multi-view software evolution: A uml-based framework for evolving object-oriented software. Proceedings of IEEE International Conference on Software Maintenance. Florence, Italy. 2001. 386–395.

3 Leite JCSP, Freeman PA. Requirements validation through viewpoint resolution. IEEE Transactions on Software Engineering, 1991, 17(12): 1253–1269. [doi: 10.1109/32.106986]

4 Kruchten P. Architecture blueprints-the “4+1” view model of software architecture. Proceedings of Tutorial Proceedings on TRI-Ada ’91: Ada’s Role in Global Markets: Solutions for A Changing Complex World. Anaheim, CA, USA. 1995.

5 Finkelstein A, Kramer B, Nuseibeh B, *et al*. Viewpoints: A framework for integrating multiple perspectives in system development. Software Engineering and Knowledge Engineering, 1992, 2(1): 31–57. [doi: 10.1142/S0218194092000038]

6 Cimatti A, Roveri M, Susi A, *et al*. Validation of requirements for hybrid systems: A formal approach. ACM Transactions on Software Engineering and Methodology, 2013, 21(4): Article No.22.

7 Goknil A, Kurtev I, van den Berg K, *et al*. Change impact analysis for requirements: A metamodeling approach. Information and Software Technology, 2014, 56(8): 950–972. [doi: 10.1016/j.infsof.2014.03.002]

8 Tekinerdogan B, Erata F. Modeling Traceability in System of Systems. Proceedings of the Symposium on Applied Computing. Marrakech, Morocco. 2017. 1799–1802.

9 朱雪峰, 金芝. 关于软件需求中的不一致性管理. 软件学报, 2005, 16(7): 1221–1231.

10 王映辉, 王立福, 张世琨, 等. 一种软件需求变化追踪方法. 电子学报, 2006, 34(8): 1428–1432. [doi: 10.3321/j.issn:0372-2112.2006.08.015]

11 Zubcoff JJ, Garrigos I, Casteleyn S, *et al*. Evaluating the use of pareto efficiency to optimize non-functional requirements

satisfaction in i* modeling. IEEE Latin America Transactions, 2016, 14(1): 331–338. [doi: 10.1109/TLA.2016.7430098]
12 周莹新, 艾波. 软件体系结构建模研究. 软件学报, 1998, 9(11): 866–872.
13 Mendonça DF, Rodrigues GN, Ali R, *et al*. GODA: A goal-oriented requirements engineering framework for runtime dependability analysis. Information and Software Technology, 2016, 80: 245–264. [doi: 10.1016/j.infsof.2016.09.005]
14 Roy JF, Kealey J, Amyot D. Towards integrated tool support for the user requirements notation. Proceedings of the 5th International Conference on System Analysis and Modeling: Language Profiles. Kaiserslautern, Germany. 2006.
15 Georg G, Mussbacher G, Amyot D, *et al*. Synergy between activity theory and goal/scenario modeling for requirements elicitation, analysis, and evolution. Information and Software Technology, 2015, 59: 109–135. [doi: 10.1016/j.infsof.2014.11.003]
16 Mylopoulos J, Chung L, Liao S, *et al*. Exploring alternatives during requirements analysis. IEEE Software, 2001, 18(1): 92–96. [doi: 10.1109/52.903174]
17 Gnaho C, Semmak F, Laleau R. Modeling the impact of non-functional requirements on functional requirements. Proceedings of International Conference on Conceptual Modeling. Hong Kong, China. 2017. 59–67.
18 Wei BY, Delugach HS. A framework for requirements knowledge acquisition using UML and conceptual graphs. Lee R. Software Engineering Research, Management and Applications. Cham: Springer, 2016.
19 孙昌爱, 金茂忠, 刘超. 软件体系结构研究综述. 软件学报, 2002, 13(7): 1228–1237.
20 梅宏, 申峻嵘. 软件体系结构研究进展. 软件学报, 2006, 17(6): 1257–1275.
21 Tibermacine C, Sadou S, That MTT, *et al*. Software architecture constraint reuse-by-composition. Future Generation Computer Systems, 2016, 61: 37–53. [doi: 10.1016/j.future.2016.02.006]
22 李潇, 魏长江. 基于追踪矩阵获取完整性需求的研究. 计算机科学, 2019, 46(6): 188–195.
23 王映辉, 张世琨, 刘瑜, 等. 基于可达矩阵的软件体系结构演化波及效应分析. 软件学报, 2004, 15(8): 1107–1115.