特定领域建模方法在医药进销存系统中的研究和应用

李 辉[1]，汪 丰[2]

（1. 东南大学生物科学与医学工程学院，南京 210096;
2. 东南大学南京生物材料与医疗器械研究所，南京 211111）

摘要：医药关乎人们的生命健康，对于医药的监管必须严格把控。在深入分析医药进销存管理需求及相关医疗信息协议后，利用特定领域建模（domain-specific modeling，DSM）方法搭建医药进销存系统模型并加以实现。以医药进销存系统为例，归纳总结创建特定领域元模型的方法及原则，介绍使用建模工具 MetaEdit+进行领域建模的全过程。研究使用的 DSM 方法，最终可通过领域模型生成直接可运行的代码。

关键词：计算机软件；特定领域建模；医疗信息协议；MetaEdit+

中图分类号：TP311.5 **文献标识码：**A **文章编号：**1674-2850(2017)10-1075-09

Research and application of the medicine invoicing system based on domain-specific modeling method

LI Hui[1], WANG Feng[2]

(1. *School of Biological Science & Medical Engineering*, *Southeast University*, *Nanjing* 210096, *China*;
2. *Nanjing Institute of Biomaterials and Medical Devices*, *Southeast University*, *Nanjing* 211111, *China*)

Abstract: Medicine is related to people's life and health, and the regulation of medicine must be strictly controlled. After analyzing the management needs of the medicine invoicing system and related medical information protocol, this paper builds up the system model by using the domain-specific modeling (DSM) method, and makes it realizable. This paper summarizes the methods and principles of creating meta-models in specific fields, and introduces the whole process of domain modeling using MetaEdit+, a modeling tool. The DSM method used in this article can ultimately generate directly runable codes by the domain model.

Key words: computer software; domain-specific modeling; medical information protocol; MetaEdit+

0 引言

随着互联网技术的快速发展，与人们生活息息相关的医疗行业逐渐走向了医疗信息数字化的道路。互联网药房是互联网医疗的一个重要分支，也是医药网络营销交易模式的一个重要分支，主要进行网上药品零售和在线药学服务。截止到 2015 年 10 月，经国家食品药品监督管理总局批准，已有 460 家企业取得《互联网药品交易服务资格证书》。

药品进销存系统是互联网药房的核心子系统，是药品进出药房的首个门户。一个合格的药品进销存系统，应当做到对药品数量及质量的严格把控，确保药品在互联网零售过程中的可控性和安全性。然而，与普通商品不同，药品数据信息的流转必须符合医疗行业的相关协议，这无疑增加了系统开发的复杂性。

作者简介：李辉（1990—），男，硕士研究生，主要研究方向：生物信息学

通信联系人：汪丰，副教授，主要研究方向：生物医学传感技术、生理信号分析、无线传感网络、情感计算与生物反馈、远程医疗. E-mail: feng.wang@seu.edu.cn

对于一个普通的软件开发人员来说，学习门槛高，以致软件的生产效率得不到有效提升。

领域特定语言（domain-specific language，DSL）被公认为处理复杂软件系统开发和维护困难的一种有效解决方案[1~4]。DSL 对问题域进行抽象，提供一系列代码层面的符号和构造器。DSM 是一种设计和开发系统的软件工程方法学，通过建立特定领域建模语言（domain-specific modeling language，DSML），进一步提高领域的抽象程度[5]。DSML 对于问题域的抽象不再以抽象代码的形式呈现，取而代之的是可视化的图形符号[6]。因此，相较于 DSL，DSML 更具有表达性和易用性。目前，DSM 方法已经被众多公司和研究机构运用到生产实践当中，涉及医疗、电信、金融等领域，MUNNELLY 等[7~8]建立了医疗保健领域的建模语言（aspect language for pervasive healthcare，ALPH）；CHRISTIANSEN 等[9]建立了金融领域的精算建模语言（actulus modeling language，AML）；CHALLENGER 等[10]建立了多代理系统的建模语言（semantic web enabled agent modeling language，SEA_ML）。

本文分析医药进销存系统的业务需求以及医疗信息领域的相关信息协议，归纳总结医药进销存领域的相关概念、规则及约束，建立一套医药进销存领域的专用语言（medicine invoicing domain-specific modeling language，MIDSML），并设计相应的代码生成器。通过使用 MIDSML，构建药品进销存系统模型，生成应用系统。

1 医药进销存系统

医药行业的发展重点在于管理，随着信息技术的快速发展，各种信息管理软件不断被使用。医药进销存系统通过系统的各个模块使使用者了解药品管理、销售仓储管理等具体功能。系统实时监控药品的库存情况，动态调整，并依据库存变化调整采购计划。科学合理地管理医药仓储、有效地利用库存空间，可以降低企业成本[11]。

1.1 功能分析

根据对系统需求的归纳及总结，医药进销存系统的功能模块如图 1 所示。

可再细分为以下几部分：

1）基本信息管理，包括药品信息、供应商信息等基本信息的管理；

2）入库信息管理，包括入库信息管理、入库报表生成等；

3）销售信息管理，包括销售订单信息管理、销售报表等；

4）出库信息管理，包括出库信息管理、出库报表等；

5）库存信息管理，包括药品库存信息查询、生成补药清单等；

6）系统管理，包括信息管理权限等。

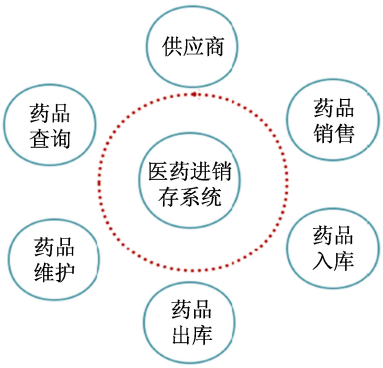


图 1 医药进销存系统的功能模块

Fig. 1 Functional modules of medicine invoicing system

1.2 业务场景分析

业务场景描述了各种数据是如何传递、流动及处理的。该医药进销存系统的各种业务场景简单阐述如下：

1）系统登录。操作员登录系统，输入用户名及密码，系统根据提交信息进行数据匹配，成功则登录系统，失败则给予提示。

2）药品入库。系统获取药品入库信息，产生入库操作。根据入库操作，系统在待检库中新增一条药品记录，同时产生入库明细记录，更新相关数据库。

3）药品检验。对待检库中的药品进行检验，合格的药品进入合格药品库中，不合格的药品则进入退

货药品库中。

4）药品销售。根据用户请求及库存信息，产生销售订单。根据用户支付、取药等情况更改订单状态。

5）药品领取。根据用户请求，生成领取操作记录。

2 医疗健康信息集成规范

医疗健康信息集成规范（Integrating the Healthcare Enterprise，IHE）是目前国际上广泛应用于解决医疗信息系统集成问题的规范。IHE 始于 1997 年，由北美放射学会（Radiology Society of North America，RSNA）和卫生保健信息管理系统学会（Healthcare Information Management Systems Society，HIMSS）联合发起，旨在促进影像和信息系统更高水平协同工作能力。

IHE 结合世界范围内众多领域专家以及医院实践经验的总结，制定出了实现信息系统协同工作以及信息共享的集成技术规范。IHE 技术规范以技术框架文档的形式表达。技术框架按照解决集成问题的医疗领域划分，IHE 最初旨在解决放射科的集成问题，随后逐步扩展，目前覆盖检验、病理、放疗等 11 个领域[12]。2009 年，IHE Pharmacy（医药领域）相关规范的制定由欧洲医院药剂师协会（European Association of Hospital Pharmacists，EAHP）和荷兰国家医疗保健 IT 研究所（Dutch National IT Institute for Healthcare，NICTIZ）联合发起。IHE Pharmacy 归纳总结了医药领域的角色、流程及相应的数据模型等。

图 2 展示了 IHE Pharmacy 对药品数据模型的部分定义。由图 2 可知，IHE 对药品数据模型的定义以 XML 的形式给出，每一种定义的数据模型都有其唯一的类型 ID（Template ID）. 药品数据模型包含药品 ID（National medicinal product code）、药品品牌名（Brand name）、用量（Pharmaceutical dose form）等信息。参照该规范，可以促进对问题域的深入理解。

```
<manufacturedProduct xmlns:pharm="urn:ihe:pharm:medication" classCode="MANU">
    <templateId root="1.3.6.1.4.1.19376.1.5.3.1.4.7.2"/>
    <templateId root="2.16.840.1.113883.10.20.1.53"/>
    <manufacturedMaterial classCode="MMAT" determinerCode="KIND">
        <templateId root="1.3.6.1.4.1.19376.1.9.1.3.1"/>
        <!-- National medicinal product code (brand-level) -->
        <code code=" " displayName=" " codeSystem=" " codeSystemName=" "/>
        <!-- Brand name -->
        <name>... </name>
        <!-- Pharmaceutical dose form -->
        <pharm:formCode code=" " displayName=" "
                        codeSystem=" " codeSystemName=" "/>
        <lotNumberText>...</lotNumberText>
        <pharm:expirationTime value=' '/>
        <!-- Container information -->
        <pharm:asContent classCode="CONT">
            <pharm:containerPackagedMedicine classCode="CONT" determinerCode="INSTANCE">
                <!-- Medicinal product code (package-level) -->
                <pharm:code code=" " displayName=" " codeSystem=" " codeSystemName=" "/>
                <!-- Brand name (package) -->
                <pharm:name>...</pharm:name>
                <pharm:formCode code=" " displayName="  "
                        codeSystem=" " codeSystemName="  "/>
                <pharm:capacityQuantity value=" " unit=" "/>
                <pharm:asSuperContent>
```

图 2 药品数据结构的部分定义

Fig. 2 Fragment definition of medicine data structure

3 特定领域建模开发流程

DSM 的开发流程主要分为两个部分：DSM 的定义及 DSM 的使用。图 3 展示了 DSM 开发的基本流程[13]。

1）领域专家定义 DSML，主要包括定义语言的概念、规则及约束等。

2）专门的软件开发人员定义代码生成器，即定义模型元素和代码之间的映射。

3）普通软件开发人员使用 DSML 创建系统模型。

4）使用领域框架，生成目标系统。

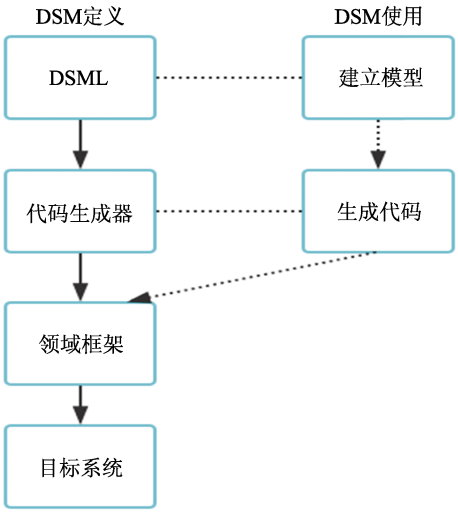


图 3 DSM 建模的主要流程[13]

Fig. 3 Main process of DSM modeling[13]

3.1 定义特定领域建模语言

定义 DSML 不是一件容易的事情。尽管许多工业书籍提供了有力的证据证明为何使用某种语言，许多学术研究从理论角度分析某种语言，但是很少有材料教语言开发者如何创建一门好的语言[14]。即便如此，在定义 DSML 时必须遵守一些必要的准则[15]：

1）DSML 必须包含该领域的所有知识（概念、关系、操作、功能等）。

2）在分析和设计阶段为领域概念提供一定的抽象，领域抽象非常重要。

3）DSML 必须具备可拓展性。

4）DSML 必须具有边界，在通用和专有之间寻求平衡。

3.2 定义代码生成器

由于目标环境的限制，需要定义代码生成器，将 DSML 中的模型元素转化为目标平台理解的代码。本文使用 MetaEdit+提供的一种专有脚本语言 MERL（MetaEdit+ reporting language）来定义代码生成器，其思想是遍历模型，提取模型中的信息，并将它转换为目标平台代码。

3.3 使用 MetaEdit+建模的主要步骤

MetaEdit+是 MetaCase 公司开发的一款特定领域建模工具，它提供了良好的可视化图形界面及专门的代码生成器定义语言 MERL. 使用 MetaEdit+建模的主要步骤如图 4 所示。

1）定义领域的概念及其属性，即 Object&Property.

2）定义规则，即 Rules.

3）定义具象的符号表示，即 Object 和 Rules 自定义标签。

4）设计代码生成器。

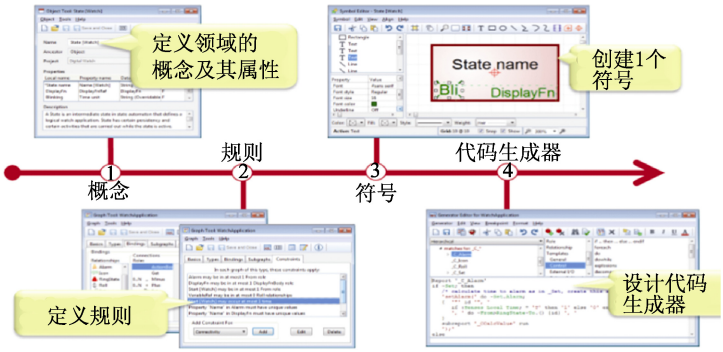


图 4 利用 MetaEdit+建模的主要步骤

Fig. 4 Main steps of modeling used by MetaEdit+

4 医药进销存系统设计与实现

4.1 定义 MIDSML

定义 MIDSML 的首要工作是提取医药进销存领域的相关概念。表 1 展示了该领域的部分概念及其符号表示。

表 1 医药进销存领域的部分概念及其符号表示

Tab. 1 Some concepts and their notations of medicine invoicing domain

概念	符号表示	概念	符号表示	概念	符号表示
供应商		药品		数据库	
全站文件		人物角色		中央系统	
入库记录		出库记录		订单记录	
取药记录		配置文件		库存子系统	
订单子系统		药品库子系统			

同时需要提炼概念之间的关联关系，本设计定义了注册、购买、登录、入库等一系列关联。关联关系限定了概念之间信息流的流转方式，这种限定方式一方面使得模型结构更加清晰易懂，同时也可在关联关系上绑定动作，以验证数据流向的正确性。本设计关联关系的部分定义如图 5 所示。

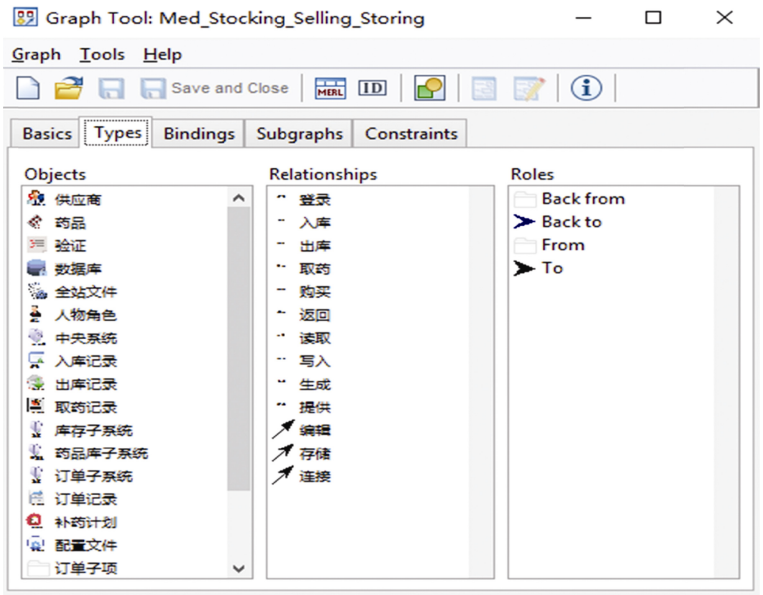


图 5 医药进销存领域概念间的关联设计

Fig. 5 Relationship design of concepts in medicine invoicing domain

医药进销存系统的模型，由 MIDSML 设计生成，本文所设计的模型结构如图 6 所示。

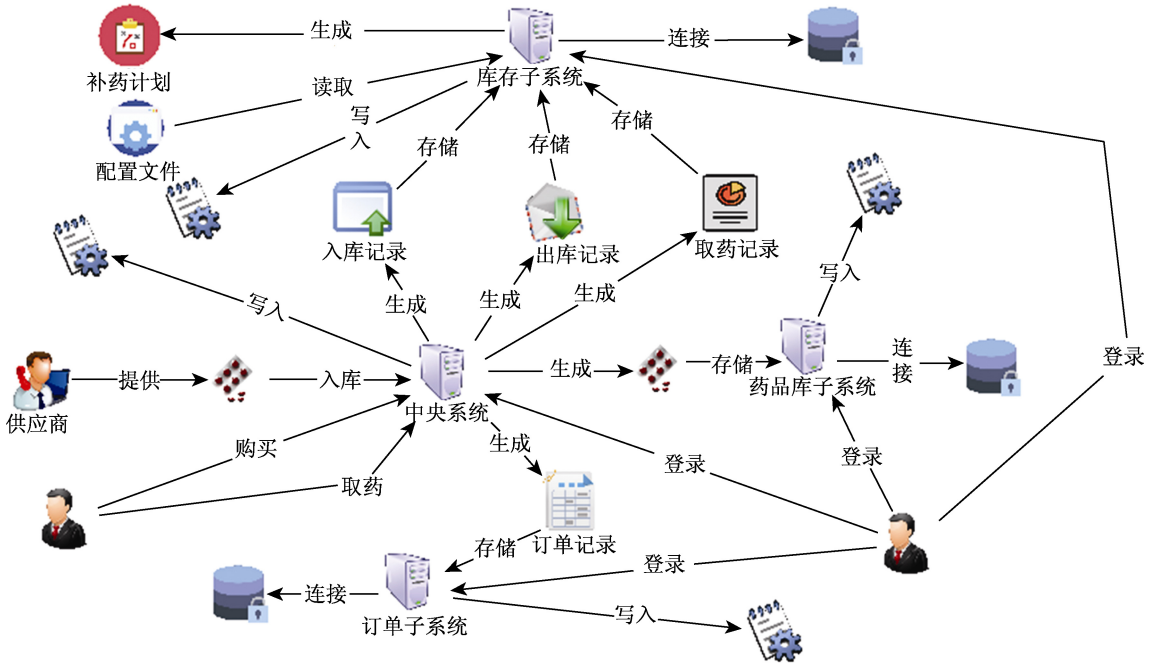


图 6 医药进销存系统模型

Fig. 6 Model of medicine invoicing system

模型中包含 1 个中央系统和 3 个子系统——库存子系统、药品库子系统及订单子系统。请求通过中央系统分别传送给各子系统模型进行处理。库存子系统用于处理与库存相关的业务请求，如入库、出库及库存统计等。药品库子系统用于处理与药品相关的业务请求，如药品信息编辑、统计等。订单子系统则用于处理与订单相关的业务，如用户购买等。在上述各系统中，其内部逻辑又构成了一个子模型。图 7 展示了中央系统的模型结构。用户的请求传送到消息路由器，消息路由器对请求信息进行过滤和转发，各处理器对请求进行相应的处理，生成最终的数据体存储在数据库中。处理器并非一个独立的实体，一

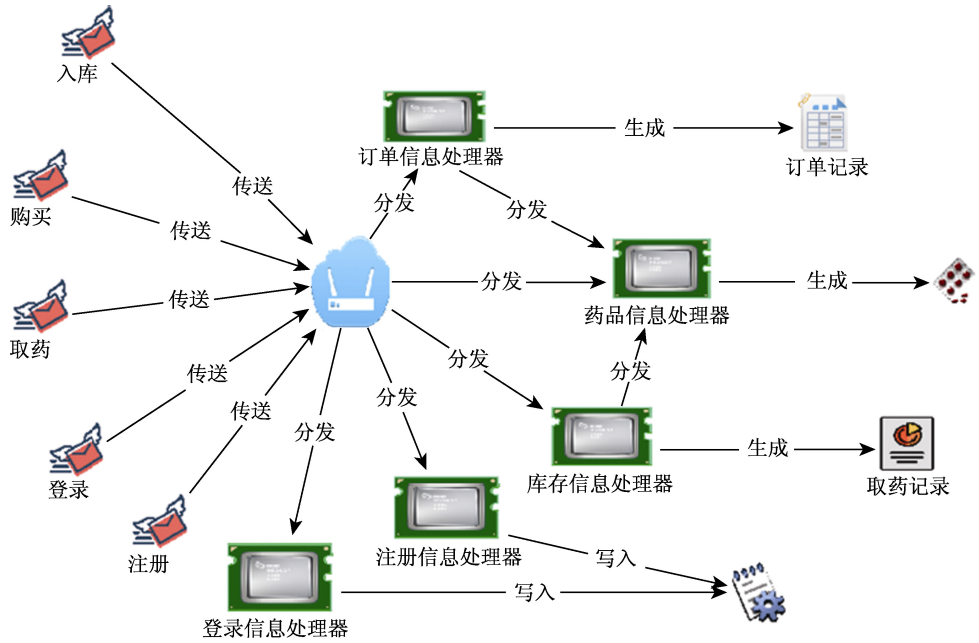


图 7 中央系统模型

Fig. 7 Model of central system

些情况下会将部分请求分发至其他处理器，以达到业务功能的完整实现。

其他子系统的模型与中央系统类似。本设计的模型框架分为两层，外层是静态结构的展示，内层是动态业务的实现。这种设计将模型中的静态信息和动态信息分割开，使得模型简单明了，易于理解。

4.2 定义代码生成器

通过遍历模型，捕获模型中所有的静态和动态信息，代码生成器可以产生可执行的输出代码。本设计将采用 Go 语言进行开发，利用 Beego 框架[16]进行实现。医药进销存系统的代码生成器结构图如图 8 所示。顶部是主生成器 Go Script，程序的入口函数以及一些初始化的操作将在此处进行定义。其下层的子生成器分别对应 Beego 框架中的 4 个组成部分——配置、模型、路由及控制器的代码生成，其中配置和模型部分的脚本对应模型中静态结构的代码实现，路由和控制器部分的脚本对应模型中动态业务的代码实现。最底层的生成器则分别对应 GO 文件中头部导入模块及函数模块的代码生成。

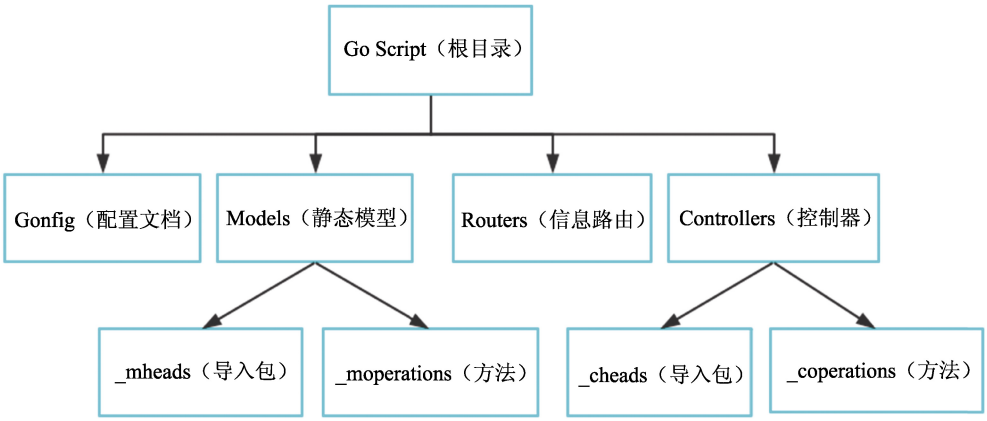


图 8 代码生成器结构图

Fig. 8 Structure diagram of code generator

路由部分的代码生成器脚本如图 9 所示。

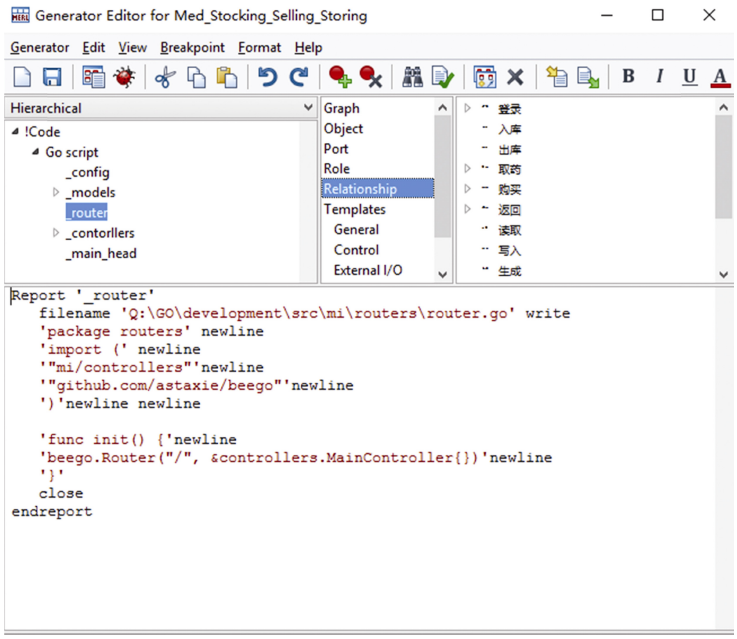
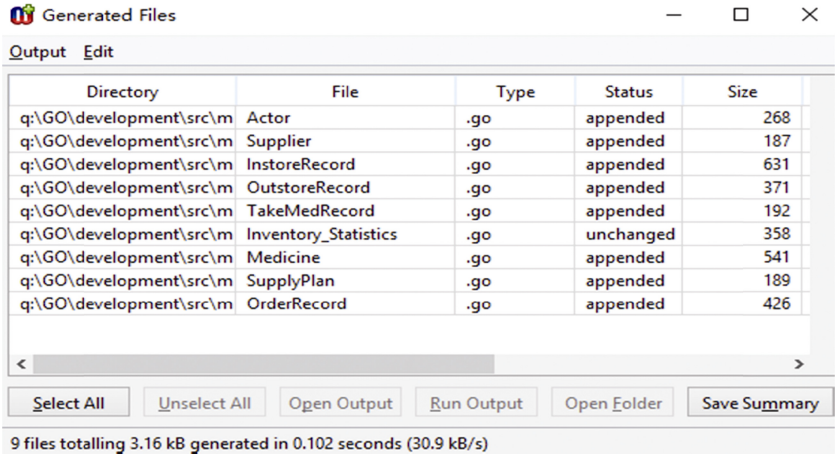


图 9 代码生成器部分脚本

Fig. 9 Part of code generator scrip

执行代码生成器，工具将医药进销存系统的模型转换为可编译执行的代码。部分 GO 文件输出如图 10 所示。



Directory	File	Type	Status	Size
q:\GO\development\src\m	Actor	.go	appended	268
q:\GO\development\src\m	Supplier	.go	appended	187
q:\GO\development\src\m	InstoreRecord	.go	appended	631
q:\GO\development\src\m	OutstoreRecord	.go	appended	371
q:\GO\development\src\m	TakeMedRecord	.go	appended	192
q:\GO\development\src\m	Inventory_Statistics	.go	unchanged	358
q:\GO\development\src\m	Medicine	.go	appended	541
q:\GO\development\src\m	SupplyPlan	.go	appended	189
q:\GO\development\src\m	OrderRecord	.go	appended	426

图 10　代码生成器的部分文件输出

Fig. 10　Some output files of code generator

4.3　模型评估与分析

模型的优劣关乎系统最终的质量，而影响模型质量的一个决定性因素则是领域元模型，即 MIDSML，因为模型中的元素及元素之间的关联关系均在元模型中定义，模型是元模型的拼装。本文将对 MIDSML 进行评估与分析，以验证 MIDSML 可以提升软件开发的速度，降低错误率。

现阶段关于 DSML 系统评测标准的研究较少[17]。DSML 评测方法学的探索也尚处于起步阶段[18]。CHALLENGER 等[19]提出了一套评测框架对多代理系统（multi-agent systems，MASs）的建模语言进行评估，本文采用同样的方式对 MIDSML 进行评估分析。表 2 展示了本文所设计的医药进销存系统模型、自动生成文件及手动补充文件的细则。由表 2 可以看出，DSM 方法可以自动生成系统的绝大部分代码，代码生成率为 89.8%.

表 2　模型、产生文件、补充文件详情

Tab. 2　Model, generated artifacts and added artifacts details

评测条目	输入模型				生成文件数目			手动添加文件数目		
评测细则	模型数目	模型中的元素	模型中的关联	逻辑错误数	文件数目	代码行数	函数数目	文件数目	代码行数	函数数目
数目/个	1	21	13	0	15	2 534	56	0	258	7

5　结论

本文介绍了利用建模工具 MetaEdit+开发医药进销存系统的完整过程，建立了一套医药进销存领域的特定领域语言 MIDSML. 同时，利用 MetaEdit+工具提供的 MERL 脚本语言，设计了元模型的代码生成器，可以将模型转化为 Go 语言代码。经测试，本文所设计的医药进销存系统，可以通过模型直接得到目标产品，代码自动生成率达 89.8%. DSM 方法降低了普通开发人员的学习成本，提升了软件的开发效率。

[参考文献] (References)

[1]　van DEURSEN A, KLINT P, VISSER J. Domain-specific languages: an annotated bibliography[J]. ACM Sigplan Notices,

2000, 35(6): 26-36.

[2] MERNIK M, HEERING J, SLOANE A M. When and how to develop domain-specific languages[J]. ACM Computing Surveys (CSUR), 2005, 37(4): 316-344.

[3] PEREIRA M J V, MERNIK M, da CRUZ D C, et al. Program comprehension for domain-specific languages[J]. Computer Science and Information Systems, 2008, 5(2): 1-17.

[4] FOWLER M. Domain-specific languages[R]. Boston: Addison Wesley, 2011.

[5] TOLVANEN J P. Domain-specific modeling: welcome to the next generation of software modeling[EB/OL]. [2005-10-28]. http://www.devx.com/enterprise/Article/29619.

[6] SCHMIDT D C. Guest editor's introduction: model-driven engineering[J]. Computer, 2006, 39(2): 25-31.

[7] MUNNELLY J, CLARKE S. A domain-specific language for ubiquitous healthcare[C]//International Conference on Pervasive Computing and Applications. New York: IEEE, 2008: 757-762.

[8] MUNNELLY J. ALPH: a domain-specific language for crosscutting pervasive healthcare concerns[C]//Proc. of the 2nd Workshop on Domain Specific Aspect Languages. New York: ACM, 2007: 4.

[9] CHRISTIANSEN D R, GRUE K, NISS H, et al. An actuarial programming language for life insurance and pensions[C]//Proceedings of 30th International Congress of Actuaries. Washington DC: International Actuarial Association, 2013: 11-C.

[10] CHALLENGER M, DEMIRKOL S, GETIR S, et al. On the use of a domain-specific modeling language in the development of multiagent systems[J]. Engineering Applications of Artificial Intelligence, 2014, 28: 111-141.

[11] 于长慧. 医药进销存信息管理系统的设计与开发[D]. 成都：电子科技大学，2014.
YU C H. Design and development of purchase-sale-stock information management system[D]. Chengdu: University of Electronic Science and Technology of China, 2014. (in Chinese)

[12] 吕旭东. IHE 技术框架与医疗工作流集成[J]. 中国医疗器械信息，2004，10（5）：26-31.
LÜ X D. IHE technical framework and workflow integration in healthcare[J]. China Medical Devices Information, 2004, 10(5): 26-31. (in Chinese)

[13] KELLY S, TOLVANEN J P. Domain-specific modeling: enabling full code generation[M]. New York: IEEE, 2008.

[14] KELLY S, POHJONEN R. Worst practices for domain-specific modeling[J]. IEEE Software, 2009, 26(4): 22-29.

[15] GRANT E S. A meta-model approach to defining UML-based domain-specific modeling language[J]. Lecture Notes in Engineering & Computer Science, 2012, 2195(1): 780-785.

[16] 金婧，陈家铨，王靖天，等. 基于 Go 语言及 Beego 等开源技术的公务车辆管理系统的设计与实现[J]. 信息通信，2015（2）：140-141.
JIN J, CHEN J Q, WANG J T, et al. Design and implementation open source technologies such as official vehicles management system based on the Go language and Beego[J]. Information & Communications, 2015(2): 140-141. (in Chinese)

[17] MOHAGHEGHI P, DEHLEN V. Where is the proof?-A review of experiences from applying MDE in industry[C]//European Conference on Model Driven Architecture-Foundations and Applications. Heidelberg: Springer-Verlag, 2008: 432-443.

[18] KAHLAOUI A, ABRAN A, LEFEBVRE E. DSML success factors and their assessment criteria[J]. Metrics News, 2008, 13(1): 43-51.

[19] CHALLENGER M, KARDAS G, TEKINERDOGAN B. A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems[J]. Software Quality Journal, 2016, 24(3): 755-795.