



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

گزارش کارآموزی

(گرایش نرم افزار)

موضوع:

بهبود بخش «تولید کد از مدل» در خط تولید نرم افزار اعوان (WISE)

رسول ملکی نژاد

استاد کارآموزی:

دکتر عباس حیدرنوری

محل کارآموزی:

شرکت مشاوران نرم افزاری اعوان

تابستان ۱۳۹۳

۱- فهرست مطالب

۱-۲- مقدمه	۱
۱-۳- معرفی محل کارآموزی	۱
۱-۳-۱- اطلاعات تماس شرکت	۱
۲-۳- تعداد کارکنان	۲
۳-۳- محصولات مهم	۲
۳-۳-۱- خط تولید نرم افزار: WISE	۲
۳-۳-۲- مدیریت ارتباط با شهروند (ZRM)	۲
۳-۳-۳- پرتال درون سازمانی (WISE Portal)	۳
۳-۳-۴- شبیه سازی شبکه اجتماعی	۳
۳-۳-۵- سیستم مدیریت گزارش های پویا	۳
۳-۳-۶- سیستم مدیریت اسناد	۳
۳-۳-۷- ابزار نصب و به روز رسانی	۴
۴- خلاصه کارهای انجام شده	۴
۴-۱- هفته اول	۴
۴-۱-۱- آشنایی	۴
۴-۱-۲- تعریف وظایف	۴
۴-۱-۳- کار با EMF	۵
۴-۲- هفته دوم	۶
۴-۲-۱- تولید کد از مدل	۶
۴-۳- هفته سوم	۸
۴-۳-۱- ساخت افزونه Eclipse	۸
۴-۳-۲- مدیریت پیچیدگی تولید کد	۸
۴-۳-۳- xpand: check	۹

۹.....	xpand: protected region-۴-۳-۴
۱۰.....	آغاز فرایند اجرای خودکار تولیدگر کد۵-۳-۴
۱۱.....	هفته چهارم۴-۴
۱۱.....	آشنایی با افزونه‌های maven۱-۴-۴
۱۲.....	افزودن افزونه fornax-oaw-m2-plugin به ir.asta.wise.generator.plugin۲-۴-۴
۱۲.....	هفته پنجم۵-۴
۱۴.....	هفته ششم۶-۴
۱۵.....	هفته هفتم۷-۴
۱۵.....	دستاوردها۸-۴
۱۶.....	ارزیابی کارآموزی۵-۴
۱۶.....	گزارش فنی۶-۴
۱۷.....	نگاه کلی به پروژه تولید کد۱-۶
۱۷.....	مدلسازی با EMF Ecore۱-۱-۶
۱۸.....	تولید کد از مدل با ابزار Xpand۲-۱-۶
۱۹.....	استفاده از کتابخانه‌ی DOM4J برای پردازش فایل‌های XML۲-۶
۲۰.....	نتیجه‌گیری۷-۴
۲۱.....	مراجع۲۱-۴

۲- مقدمه

معماری مدل محور^۱ یک رویکرد رایج در مهندسی نرم افزار است. در این رویکرد، استفاده از مدل سازی، سطح انتزاع برنامه نویسی را زیاده تر می کند. تولید کد از مدل نیز، تولید نرم افزار را سریع تر، خطاهای انسانی در ساخت کد را کمتر و مطابقت با فناوری های نوظهور را سریع تر می کند. [۱]

شرکت اعوان نیز از این رویکرد در خط تولید نرم افزار خود موسوم به WISE بهره می برد. بهبود بخش ساخت کد از روی مدل در WISE، وظیفه اصلی دوره کارآموزی من بود که گزارش آن در این مستند بیان شده است. در این گزارش، پس از معرفی شرکت اعوان، ابتدا جزئیات مجموعه کارهای انجام شده در طول دوره برای بهبود بخش ساخت کد به صورت هفتگی آورده شده است. سپس به بیان دستاوردها و ارزیابی دوره کارآموزی پرداخته شده است. در انتها نیز در یک نگاه اجمالی، کارهای به انجام رسیده و مطالب آموخته شده، با عنوان گزارش فنی نوشته شده است.

۳- معرفی محل کارآموزی

شرکت مشاوران نرم افزاری اعوان در سال ۱۳۸۳ تاسیس شده است. این شرکت عضو شورای عالی انفورماتیک و سازمان نظام صنفی رایانه ای است. راهبرد اعوان، استفاده از فناوری های تخصصی مانند J2EE و Oracle و نخبگان علمی و فنی برای تولید نرم افزارهای سازمانی بزرگ (و نه متوسط و کوچک) است.

۳-۱- اطلاعات تماس شرکت

آدرس: خیابان آزادی، ابتدای خیابان توحید، جنب بانک گردشگری، پلاک ۳۷، طبقه سوم

شماره تلفن ها: ۶۶۹۰۴۸۱۷ - ۶۶۹۰۴۶۲۸

^۱ Model Driven Architecture (MDA)

شماره فکس: ۶۶۹۰۴۷۲۰

سایت اینترنتی: <http://asta.ir>

آدرس پست الکترونیکی: info@asta.ir

۳-۲- تعداد کارکنان

تعداد کارکنان اعوان، که در محیط رسمی و اداری فعال هستند حدوداً ۱۵ نفر است. این شرکت، هم‌چنین تعدادی دیگر از نیروهای خود را برای فاز نگهداری و پشتیبانی محصولات خود به کار گرفته‌است. نرم‌افزارهای اعوان بر بستر J2EE و فناوری‌های متن‌باز ساخته می‌شوند؛ از این رو تخصص در J2EE (و سایر فناوری‌های مرتبط با تولید نرم‌افزارهای سازمانی در این بستر مانند ساخت واسط کاربری و پایگاه داده)، Linux و Oracle از مهارت‌هایی است که اغلب کارکنان اعوان دارند.

۳-۳- محصولات مهم^۲

۳-۳-۱- خط تولید نرم‌افزار: WISE

WISE، راه حل اختصاصی این شرکت برای تولید سریع و بهینه نرم‌افزارهای سازمانی بزرگ است. این خط تولید با در نظر گرفتن ویژگی‌های مشترک این گونه پروژه‌ها، معماری لازم برای آن‌ها را در نظر گرفته و ابزارهای مختلفی از جمله تولید کد از مدل را فراهم کرده‌است. بخش زیادی از کار اعوان، مشاوره تولید نرم‌افزار است؛ به این معنا که اعوان، WISE را به شرکت‌های نرم‌افزاری می‌فروشد و نحوه‌ی استفاده و پشتیبانی‌های بعدی را در اختیارشان قرار می‌دهد تا در تولید نرم‌افزار از آن استفاده کنند.

۳-۳-۲- مدیریت ارتباط با شهروند (ZRM)

«مدیریت ارتباط با شهروند» یا citiZen Relationship Management - که به اختصار ZRM نامیده

^۲ گرفته شده از [۲]

می‌شود - یک سیستم نرم افزاری یکپارچه است که یک پل ارتباطی بهینه بین سازمان و مراجعین برقرار و رسیدگی را در داخل سازمان آسان می‌کند.

۳-۳-۳- پرتال درون سازمانی (WISE Portal)

پرتال سازمانی اعوان موسوم به WiSE Portal بستر مناسبی برای توسعه و تجمیع سامانه‌های مبتنی بر وب درون سازمانی فراهم می‌کند.

۳-۳-۴- شبیه سازی شبکه اجتماعی

شبیه‌ساز شبکه‌های اجتماعی نرم‌افزاری برای شبیه‌سازی کامپیوتری شبکه‌های اجتماعی است. تحلیل‌گران شبکه‌های اجتماعی می‌توانند با کمک این نرم‌افزار، شبکه مورد نظر خود را به صورت کامپیوتری شبیه‌سازی کنند، ویژگی‌های شبکه را تحلیل کنند و نتیجه برخی وقایع و تغییرات در این شبکه را پیش‌بینی کنند. این نرم‌افزار در حوزه‌هایی که دارای مفهوم شبکه اجتماعی هستند، کاربرد خواهد داشت.

۳-۳-۵- سیستم مدیریت گزارش‌های پویا

انعطاف‌پذیری در طراحی و تولید گزارشات یکی از دغدغه‌های اصلی مدیران سازمان‌هاست. این محصول موسوم به WiSE RMS، امکانات لازم را برای بهره‌برداری از گزارش‌های پویا در پایگاه داده دلخواه فراهم می‌کند. پس از طراحی گزارش در ابزارهایی همچون BIRT Designer، iReport یا Crystal Report Designer، می‌توان فایل گزارش را در این سیستم بارگذاری و از امکاناتی مثل تعریف پارامترهای گزارش‌گیری، تعریف سطح دسترسی گزارش‌گیری، زمان‌بندی گزارش‌گیری و ... استفاده کرد.

۳-۳-۶- سیستم مدیریت اسناد

سیستم مدیریت اسناد اعوان، DBFS، امکاناتی شبیه به Windows File Explorer را در محیط وب فراهم می‌کند. این سیستم با ذخیره اطلاعات فایلها در پایگاه داده جستجو، پشتیبان‌گیری، Replication و سایر امور مربوط، ذخیره و بازیابی فایل‌ها را آسان می‌کند.

۳-۳-۷- ابزار نصب و به روز رسانی

نصب و به‌روزرسانی سیستم‌های مبتنی بر Java EE از دغدغه‌ها و مشکلات راهبران این سیستم‌ها است. ابزار نصب و به‌روزرسانی اعوان موسوم به WiSE Installer شما را یاری می‌کند تا برنامه‌های Java EE را به سادگی طی چند دقیقه نصب و تنظیم کنید و نگران فراموش شدن بخش‌هایی از تنظیمات و اجرا نادرست قسمت‌هایی از عملیات نباشید.

۴- خلاصه کارهای انجام شده

در ادامه به خلاصه کارهای انجام شده در هر هفته اشاره خواهیم کرد. البته در اینجا منظور از هفته، ۳۲ ساعت حضور در شرکت است و ممکن است این ساعات، در یک هفته واقعی قرار نداشته باشند.

۴-۱- هفته اول

۴-۱-۱- آشنایی

هفته‌ی اول، هفته‌ی آشنایی با شرکت، آموزش‌های اولیه و شرح وظایف بود. حساب کاربری لازم برای دسترسی به شبکه و یکی داخلی شرکت و کارگزار svn داخلی برایم ایجاد شد. آقای مهندس پیشوایی، مدیر شرکت، پس از معرفی شرکت مستنداتی را برای آشنایی بیشتر برایم ارسال کردند.

اعوان برای تولید و مشاوره تولید نرم‌افزار، از خط تولید خود به نام WISE استفاده می‌کند. با استفاده از مستندات ارسالی و هم‌چنین شبکه و یکی داخلی، با محصولات شرکت، مخصوصاً WISE، آشناتر شدم.

مدیر شرکت، سپس تکنولوژی‌هایی را که در شرکت استفاده می‌شوند معرفی کرد و به بیان میزان اهمیت هرکدام پرداخت. پس باید برنامه‌ای برای یادگیری تکنولوژی‌های مهم و ضروری تنظیم می‌کردم.

۴-۱-۲- تعریف وظایف

در ادامه، مهندس پیشوایی، سرپرست کارآموزی، وظیفه اصلی دوره‌ی کارآموزی‌ام را شرح داد. به دلیل

آشنایی ام با UML و مدلسازی شیء گرا، تعیین شد که ادامه‌ی کار مجتبی ورمزیار، کارآموز تابستان ۹۲ را در پیش بگیریم که توسعه و بهبود بخش تولید خودکار کد از مدل WiSE بود. اعوان در WISE، برای تولید نرم‌افزار از یک متامدل میانی استفاده می‌کند. این متامدل در خود اعوان تعریف شده و صورت مفهومی^۳ آن وجود دارد. در واقع، بخش اصلی ساخت نرم‌افزار، ساخت مدل دلخواه از روی این متامدل و تبدیل آن مدل به کد است. کار اصلی این خواهد بود که این فرایند در چارچوب مدل‌سازی Eclipse^۴ (EMF) اجرا شود؛ به این ترتیب که ابتدا این متامدل با Ecore توصیف شود. اگر متامدلی با Ecore توصیف شود با ابزارهای EMF می‌توان از آن متامدل، مدل نمونه ساخت و آن را ویرایش کرد. گام نهایی، تولید کد این مدل است. ابزارهایی^۵ وجود دارد که امکان تولید کد از این مدل را فراهم می‌کند.

در این هفته، هم‌چنین به مطالعه و اجرای مثال در مورد زبان برنامه‌سازی Groovy پرداختم که ممکن است در فرایند تولید کد به کار گرفته شود. گروهی یک زبان برنامه‌سازی چابک^۶ و پویا^۷ برای ماشین مجازی جاوا است.

۴-۱-۳- کار با EMF

ابتدا قرار شد یک بار فرایند کلی را به شکل آزمایشی از ابتدا تا انتها به پیش ببرم: تعریف یک متامدل فرضی، گرفتن مدل از آن و تولید کد آن مدل.

خودآموز [این پیوند](http://eclipsesource.com/blogs/tutorials/emf-tutorial) برای شروع کار مدلسازی با EMF مناسب بود.

(<http://eclipsesource.com/blogs/tutorials/emf-tutorial>)

این هفته، توانستم یک متامدل ساده را با Ecore توصیف کنم و یک مدل از روی آن بسازم. متامدل

^۳ Conceptual

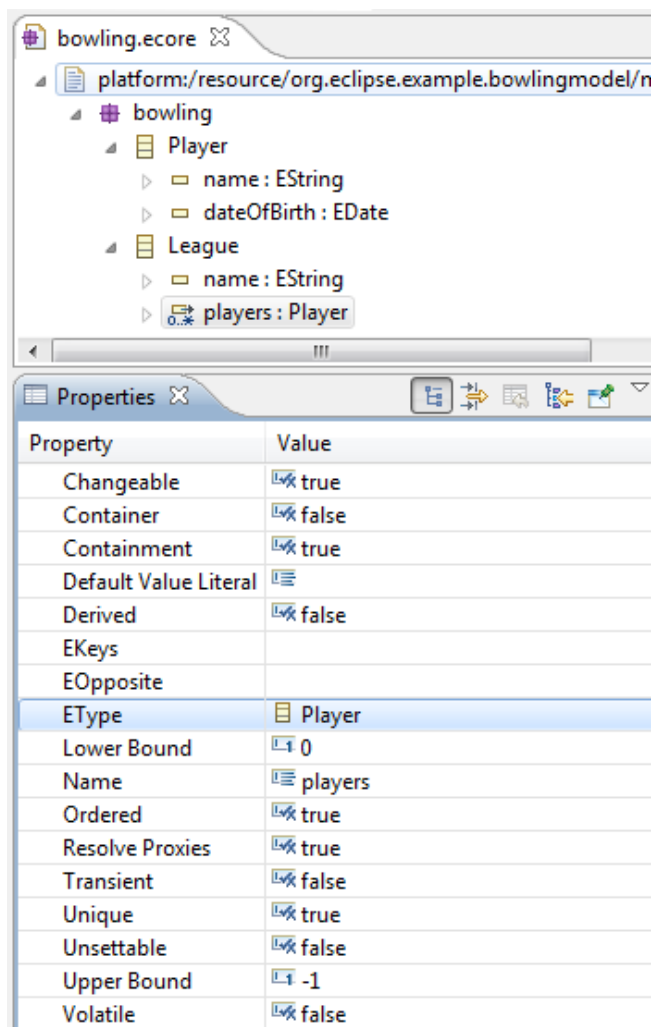
^۴ Eclipse Modeling Framework (EMF)

^۵ JET, Acceleo, Xpand, etc

^۶ Agile

^۷ Dynamic

توصیف شده در شکل دیده می‌شود.

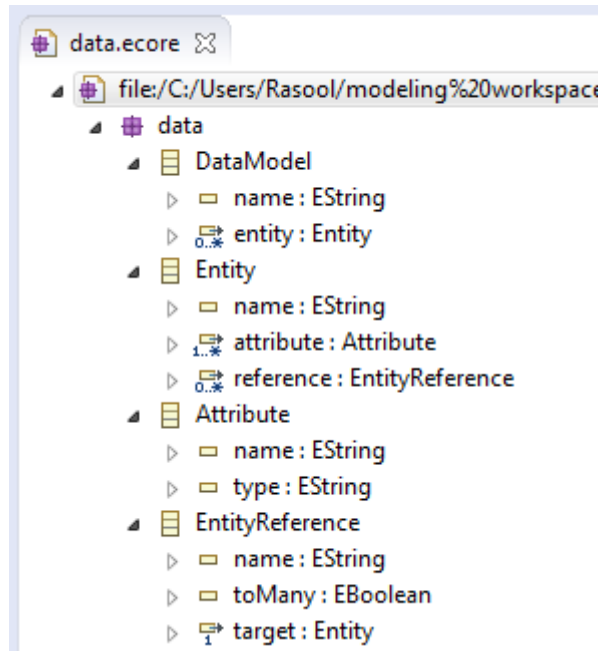


تصویر ۱ - متامدل فرضی توصیف شده با Ecore

۴-۲- هفته دوم

۴-۲-۱- تولید کد از مدل

در این هفته متامدل ساده دیگری را توصیف کردم که در شکل آورده شده است.



تصویر ۲- متامدل فرضی برای استفاده در تولید کد - توصیف شده با Ecore

اولین فرایند تولید کد از روی مدل، در این هفته به انجام رسید.

Eclipse امکاناتی برای تبدیل مدل به متن دارد که در پروژه M2T^۸ جمع شده‌اند. Acceleo و Xpand دو عنصر مهم در این پروژه هستند. با Acceleo کار کردم و کد یک متامدل با آن تولید شد. سپس به این نتیجه رسیدم که این ابزار، انعطاف کمتری نسبت به Xpand در تولید کد دارد و انگار فقط قالب‌های از پیش تعریف شده‌ای را برای مدل‌های ورودی می‌پذیرد:

Is my UML Case tool compatible with Acceleo?

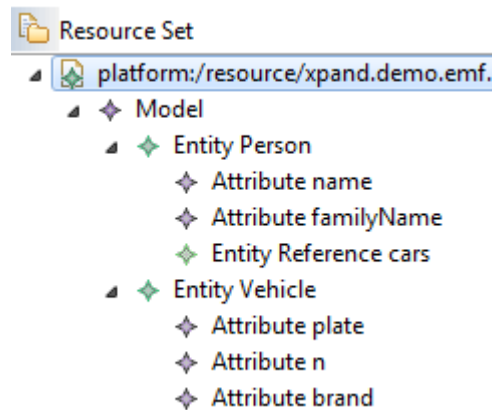
Acceleo may read any model produced by any UML Case tool. But all possible version combinations (UML + XMI + modeler specificities) are not yet supported. (see [compatibility list](#)). If you want to use Acceleo with an UML Case tool which isn't supported, please contact our support to realize this specific import.^۹

بنابر این، Xpand را برای تولید کد انتخاب کردم. بعد از امتحان کردن منابع و خودآموزهای اینترنتی مختلف خودآموز [این پیوند](#) را مناسب‌تر یافته، مراحل آن را به انجام رساندم. مدل نمونه ساخته شده در

^۸ Model to Text

^۹ گرفته شده از <http://www.acceleo.org/pages/faq/en>

تصویر دیده می شوند.



تصویر ۳ - مدل ساخته شده از متامدل فرضی توصیفی با EMF

۴-۳- هفته سوم

۴-۳-۱- ساخت افزونه Eclipse

در این هفته، متامدلی را که با ecore توصیف شده بود، به صورت یک افزونه^{۱۰} eclipse ساختم. برای این کار، پروژه افزونه را به صورت فایل jar، export کرده و در پوشه‌ی plugins در eclipse کپی می‌کنیم. خودآموز زیر می‌تواند در این زمینه مفید باشد:

<http://www.vogella.com/tutorials/EclipsePlugIn/article.html>

۴-۳-۲- مدیریت پیچیدگی تولید کد

تولید کد را با مدیریت پیچیدگی و تقسیم تولیدگر کد به چند قسمت کوچکتر و تعریف کلاس‌های utility ادامه دادم.

^{۱۰} plugin

xpand: check -۳-۳-۴

xpand امکان check را در حین تولید کد به ما می‌دهد؛ به این معنا که می‌توان قواعدی^{۱۱} تعریف کرد که در حین تولید کد از مدل بررسی شود و در صورت برقرار نبودن آن قواعد، هشدار^{۱۲} یا خطا^{۱۳} اعلام شود. مثلاً تعریف کردم که اگر اسم صفتی^{۱۴} در مدل، یک کاراکتر بود هشدار داده شود، یا اگر اسم دو صفت در یک موجودیت^{۱۵} مدل، تکراری بود خطا داده شود. نمونه این تعریف قواعد در تصویر دیده می‌شود.

```
import data;
context data::Entity ERROR
  "Names of Entity attributes must be unique":
    attribute.forAll(a1| attribute.notExists(a2| a1 != a2 && a1.name == a2.name ) );

context Attribute WARNING
  "Name of '" + name + "' too short." :
    name.length > 1;
```

تصویر ۴- تعریف قواعد در تولید کد با xpand

xpand: protected region -۴-۳-۴

امکان دیگر xpand برای تولید کد، تعریف محدوده‌ی حفاظت شده^{۱۶} است؛ به این معنا که می‌توان قسمت‌هایی را تعریف نمود که کد آن فقط در همان یک بار اول، ایجاد شود و در دفعات بعدی اجرای تولیدگر کد، دست نخورد. مثلاً می‌توان قسمت‌های داخل توابع را به عنوان محدوده حفاظت شده تعریف کرد که اگر برنامه‌نویس کد خود را در داخل آن نوشت، با تولید دوباره کد، آن قسمت پاک نشود. این امکان نیز در تولیدگر کد به انجام رسید. نمونه کد تولید شده نهایی را در تصویر می‌بینید.

^{۱۱} rules

^{۱۲} warning

^{۱۳} error

^{۱۴} attribute

^{۱۵} entity

^{۱۶} protected region

```

/*PROTECTED REGION ID(Person) ENABLED START*/
// generated at 2014/08/04 17:37
/*PROTECTED REGION END*/
public abstract class PersonImplementation {

    private String name;
    public void setName( String value ) {
        this.name = value;
    }
    public String getName() {
        return this.name;
    }

    private String familyName;
    public void setFamilyName( String value ) {
        this.familyName = value;
    }
    public String getFamilyName() {
        return this.familyName;
    }
}

```

تصویر ۵ - نمونه کد تولید شده از مدل فرضی

۴-۳-۵- آغاز فرایند اجرای خودکار تولیدگر کد

فرایند تولید کد از روی مدل به انجام رسید ولی این کافی نبود؛ نیاز بود که این فرایند، به صورت خودکار اجرا شود، و Maven اولین نامی است که در زمینه ساخت خودکار به ذهن می‌رسد. Maven وابستگی^{۱۷}ها را مدیریت کرده و به کمک افزونه‌های فراوانش، فرایندهای تولید نرم‌افزار را به صورت خودکار انجام می‌دهد. Maven استفاده زیادی در پروژه‌های اعوان دارد. در زمینه تولید کد نیز نیاز به یک افزونه بود که جریان کاری^{۱۸} تولید کد را صدا بزند. این جریان‌های کاری فرایند تولید کد را تعریف می‌کنند که نمونه‌ای از آن در تصویر دیده می‌شود. در جست و جو برای این افزونه، افزونه‌ای با نام -Fornax Model Workflow- M2-Plugin را پیدا کردم.

^{۱۷} dependency

^{۱۸} workflow

```

<?xml version="1.0" encoding="windows-1252"?>
<workflow>
<property file="workflow.properties"/>
<bean class="org.eclipse.emf.mwe.utils.StandaloneSetup" >
<platformUri value="./.."/>
<registerGeneratedEPackage value="data.DataPackage"/>
</bean>
<component class="org.eclipse.emf.mwe.utils.Reader">
  <uri value="${modelFile}" />
  <modelSlot value="model" />
</component>
<component
  class="org.eclipse.xtend.check.CheckComponent">
<metaModel id="mm"
  class="org.eclipse.xtend.typesystem.emf.EmfRegistryMetaModel">
</metaModel>
<checkFile value="${checkFile}"/>
<emfAllChildrenSlot value="model"/>
</component>

<component class="org.eclipse.xpand2.Generator">
<metaModel id="mm" class="org.eclipse.xtend.typesystem.emf.EmfRegistryMetaModel">
</metaModel>
<expand value="templates::Root::Root FOR model"/>
<outlet path="${srcGenPath}"/>
<postprocessor
  class="org.eclipse.xpand2.output.JavaBeautifier"/>
</outlet>
<prSrcPaths value="${srcGenPath}"/>
<prDefaultExcludes value="false"/>
<prExcludes value="*.xml"/>
</component>
</workflow>

```

تصویر ۶- نمونه جریان کاری در تولید کد با Xpand

۴-۴- هفته چهارم

۴-۴-۱- آشنایی با افزونه‌های maven

هر افزونه Maven مجموعه‌ای از یک یا چند MOJO^{۱۹} است. در واقع هر MOJO، یک *goal* اجرایی

در maven است. برای استفاده از یک افزونه، باید مشخصات^{۲۰} آن افزونه را در قسمت build در pom.xml

^{۱۹} Maven Old Java Object

^{۲۰} groupId, artifactId, version (optional)

پروژه خودمان وارد کنیم. در همانجا پیکربندی^{۲۱} مربوطه به افزونه را می توان وارد کرد و نیز تعیین کرد که کدام هدف^{۲۲} از افزونه در کدام فاز از چرخه ساخت maven اجرا شود.^{۲۳} در پیوند زیر اطلاعات بسیار خوبی در مورد نحوه ی ساخت افزونه های maven هست:

<http://maven.apache.org/plugin-developers/index.html>

۴-۲- افزودن افزونه `fornax-oaw-m2-plugin` به `ir.asta.wise.generator.plugin`

پیش تر گفتیم که `fornax-oaw-m2-plugin` یک افزونه maven برای اجرای جریان های کاری مدل (oAW 4.x / Eclipse MWE) است. بعد از کار با این افزونه و مواجه شدن با مشکلات مربوط به دریافت وابستگی ها و رفع تداخل های آنها، به توصیه مهندس پیشوایی، قرار بر این شد که کد منبع آن را گرفته و در پروژه ی `ir.asta.wise.generator.plugin` وارد کنم. این پروژه که اسمش `wise maven plugin` است، در واقع یک افزونه است که از قبل برای کارهای مربوط به اجرای خودکار فرایندهای تولید کد در WISE ساخته شده و استفاده می شود. به عبارت دیگر، حالا می بایست MOJO های افزونه `fornax` را در `wise maven plugin` وارد کنم.

۴-۵- هفته پنجم

در این هفته، افزودن افزونه `fornax-oaw-m2-plugin` را به پروژه `wise maven plugin` افزودم. حالا برنامه ساز می تواند با افزودن افزونه `wise maven plugin` به پروژه خود (در `pom.xml`) از `run-workflow` MOJO در فرایند ساخت پروژه خود استفاده کند. این استفاده را در شکل می بینید.

^{۲۱} configuration

^{۲۲} goal

^{۲۳} آشنایی با فازهای مختلف چرخه های maven: <http://maven.apache.org/ref/3.2.2/maven-core/lifecycles.html>

```

<plugin>
  <groupId>ir.asta.wise</groupId>
  <artifactId>wise-maven-plugin</artifactId>
  <!-- <version>1.0-SNAPSHOT</version> -->
  <configuration>
    <force>true</force>
    <workflowEngine>mwe</workflowEngine>
    <workflowDescriptor>xpand/workflow.mwe</workflowDescriptor>
  </configuration>
  <executions>
    <execution>
      <phase>generate-sources</phase>
      <goals>
        <goal>run-workflow</goal>
      </goals>
    </execution>
  </executions>
</plugin>

```

تصویر ۷- استفاده از هدف `run-workflow` در پروژه با افزودن `wise maven plugin`

در اینجا، یک فرایند آزمایشی توصیف یک متامدل فرضی ساده با EMF، گرفتن یک مدل نمونه از آن و تولید کد آن مدل با استفاده از maven و xpand به انجام می‌رسد. گام بعدی، اجرای این فرایند بر روی متامدل میانی اعوان در WISE است؛ یعنی در آغاز باید متامدل میانی شرکت، با EMF توصیف شود. با بررسی فایل‌های xml موجود برای مدل‌ها و پرسش از مدیر فنی شرکت، سعی کردم این متامدل را استخراج و با Ecore توصیف کنم. تصویر متامدل کامل‌شده را در شکل می‌بینید.

- ✚ metadata
 - ▷ 📄 RootDataModel
 - ▷ 📄 Package -> NamedElement
 - ▷ 📄 NamedElement
 - ▷ 📄 Entity -> NamedElement
 - ▷ 📄 Enumeration -> NamedElement
 - ▷ 📄 Constant -> NamedElement
 - ▷ 📄 Attribute -> NamedElement
 - ▷ 📄 Identifier -> Attribute
 - ▷ 📄 Property -> Attribute
 - ▷ 📄 Association
 - ▷ 📄 PropertyValidator
 - ▷ 📄 EmailValidator -> PropertyValidator
 - ▷ 📄 NonnumericValidator -> PropertyValidator
 - ▷ 📄 LengthValidator -> PropertyValidator
 - ▷ 📄 LongValidator -> PropertyValidator
 - ▷ 📄 DoubleValidator -> PropertyValidator
 - ▷ 📄 PatternValidator -> PropertyValidator
 - ▷ 📄 EditorType
 - ▷ 📄 GeneratorClass
 - ▷ 📄 SearchMethod
 - ▷ 📄 RelationType
 - ▷ 📄 Multiplicity

تصویر ۸- متامدل میانی WISE توصیف شده با EMF

۴-۶- هفته ششم

در این هفته، کار تبدیل مدل میانی توصیف شده با فایل های gen.xml را به مدل های EMF، که متامدل آن هفته پیش ساخته شد، آغاز شد. این کار نیازمند کار با فایل های xml بود. با توصیه آقای پیشوایی از کتابخانه dom4j برای پردازش این فایل ها و همزمان از JUnit برای آزمون برنامه استفاده کردم. این برنامه نیازمند بررسی حالت های مختلف عناصر موجود در فایل های xml بود که برای پرهیز از عبارت های switch-case ناخوانا و تودرتو در طراحی خود از چندریختی^{۲۴} استفاده کردم. بدین ترتیب، مدیریت پیچیدگی برنامه آسان تر شد.

^{۲۴} polymorphism

۴-۷- هفته هفتم

در این هفته نیز کار هفته پیش را ادامه دادم و در پایان دوره، ساخت برنامه تبدیل مدل gen.xml به مدل EMF به پایان رسید.

۴-۸- دستاوردها

کارآموزی در اعوان، یک شروع خوب برای ورود به صنعت حرفه‌ای نرم‌افزار برای من بود. در این قسمت به صورت موردی به دستاوردها و آموخته‌هایم اشاره می‌کنم.

- کسب تجربه عملی استفاده از رویکردهای نظری مهندسی نرم‌افزار در پروژه‌های سازمانی بزرگ
- کسب تجربه کار گروهی در تولید نرم‌افزار
- افزایش کیفیت قسمت ساخت کد از مدل در خط تولید نرم‌افزار اعوان
- آشنایی با maven و افزونه‌های آن
- یادگیری مدل‌سازی در چارچوب مدل‌سازی Eclipse با استفاده از Ecore
- یادگیری و کار با ابزار ساخت کد از مدل Acceleo
- یادگیری و کار با استفاده از ابزار Xpand در ساخت کد از مدل
- کار عملی با تست نرم‌افزار (JUnit)
- یادگیری برنامه‌نویسی با زبان Groovy

۵- ارزیابی کارآموزی

من با کارآموزی در اعوان، با رویکرد مهندسی در تولید نرم افزار، آن هم نرم افزارهای سازمانی بزرگ آشنا شدم و برای نخستین بار کاربرد مطالب نظری آموخته شده مهندسی نرم افزار را در صنعت دیدم. رویکردهایی مثل معماری مدل محور، تست نرم افزار، نسخه بندی نرم افزار، استفاده از ابزارهای تولید خودکار مثل ant و maven، ابزارهای تولید گزارش، ابزارهای مدیریت پروژه و پیگیری مسئله مثل Jira و ابزارهای برنامه سازی گروهی مثل Svn.

وجود انعطاف در تنظیم ساعت کاری در شرکت نیز یکی از مزیت های قابل توجه کار در این شرکت برای من بود. ساعات کاری معمول شرکت بین ۹ تا ۱۷ است ولی بعد از آن نیز در امکان حضور در شرکت فراهم است. هم چنین روزهای پنجشنبه نیز امکان کار وجود دارد.

فضای فیزیکی شرکت نیز در حد مطلوب بود. هر چند با کمی بهبود جزئیات، می توان آن را به حد ایده آل رساند.

اعوان با محیطی آرام و دوستانه و نیز وجود نیروهای با دانش فنی بالا محیط مناسبی برای کارآموزی و کسب تجربه برای من بود. مشاهده صرف صبحانه و ناهار دسته جمعی در شرکت و برنامه فوتبال آخر هفته کارکنان شرکت برایم جالب بود. مدیران و همکاران نیز همواره برخوردی با احترام، دوستانه و صمیمانه با من داشتند. از آن ها ممنونم. آقای مهندس پیشوایی همواره راهنما و پیشنهاد کننده راه حل های خوب در تصمیم گیری ها و برخورد با خطاها بودند. از ایشان سپاسگزارم. از دوست و همکار محترم آقای خلیل قاسم زاده و نیز دوست عزیزم مجتبی ورمزیار بابت راهنمایی هایشان تشکر می کنم.

۶- گزارش فنی

در این بخش، یک گزارش فنی با یک دید جامع نسبت به کل کار آمده است. در این گزارش به موارد

آموخته در جریان کارهای انجام شده در دوره اشاره شده است.

۶-۱- نگاه کلی به پروژه تولید کد

همان طور که در بخش گزارش هفتگی اشاره شد، اعوان در پروژه های خود از یک متامدل میانی استفاده می کند. این متامدل میانی به صورت مفهومی وجود دارد. مدل های آن نیز به چند صورت مختلف از جمله به صورت فایل های xml وجود دارند و برای تولید کد پروژه ها استفاده می شوند. یکی از اهداف پروژه تولید کد این است که ابتدا این متامدل با EMF Ecore توصیف شود. بعد از این توصیف، دیگر می توان مدل های پروژه های جدید را از این متامدل گرفت و به این ترتیب، میتوان از ابزارهای تولید کد برای تولید کد مدل مورد نظر استفاده کرد. همچنین می توانیم با تبدیل مدل های موجود به مدل EMF، تولیدگر کد را برای تولید کد این مدل ها نیز به کار ببریم.

۶-۱-۱- مدلسازی با EMF Ecore

EMF، یک چارچوب مدلسازی و تولید کد بر مبنای یک مدل داده ای ساختار یافته است. Ecore نیز، متامدلی است برای متامدل ها، که در دل چارچوب مدلسازی Eclipse وجود دارد. [۳].

برای توصیف متامدل میانی شرکت در EMF، به یادگیری و تمرین مدلسازی با Ecore پرداختیم. سپس با بررسی مدل های موجود به شکل xml و پرسش از سرپرست کارآموزی، توصیف این متامدل را با Ecore به انجام رساندم. در شکل، متامدل نهایی دیده می شود.

- ▲ Package -> NamedElement
 - ▷ entities : Entity
 - ▷ enums : Enumeration
 - ▷ packages : Package
- ▲ NamedElement
 - ▷ name : EString
- ▷ Entity -> NamedElement
- ▲ Enumeration -> NamedElement
 - ▷ items : Literal
- ▷ Literal -> NamedElement
- ▷ Attribute -> NamedElement
- ▷ Identifier -> Attribute
- ▷ Property -> Attribute
- ▷ PropertyValidator
- ▷ EmailValidator -> PropertyValidator
- ▷ NonnumericValidator -> PropertyValidator
- ▷ LengthValidator -> PropertyValidator
- ▷ LongValidator -> PropertyValidator
- ▷ DoubleValidator -> PropertyValidator
- ▷ PatternValidator -> PropertyValidator
- ▷ RequiredValidator -> PropertyValidator
- ▷ TelephoneValidator -> PropertyValidator
- ▷ Relation -> NamedElement
- ▷ EditorType
- ▷ GeneratorClass
- ▷ SearchMethod
- ▷ RelationType

تصویر ۹ - متامدل میانی پروژه‌های اعوان توصیف شده با emf (نسخه نهایی پایان دوره)

۶-۱-۲- تولید کد از مدل با ابزار Xpand

بعد از بررسی ابزارهای تولید کد موجود در Xpand، EMF را انتخاب کرده و شروع به یادگیری تولید

کد با آن کردم. xpand زبانی است که برای تولید کد از مدل‌های EMF طراحی شده است [۴].

در این ابزار از قالب‌های تولید کد استفاده می‌شود. نمونه ای از این قالب‌ها در شکل دیده می‌شود. در

این شکل امکان استفاده از محدوده‌های حفاظت شده^{۲۵} برای جلوگیری از دوباره‌نویسی^{۲۶} را هم می‌بینید.

^{۲۵} protected region

^{۲۶} overwrite

```

«DEFINE Entity FOR data::Entity»
  «FILE classFileName()»
«PROTECT CSTART "/*" CEND "*/" ID this.name»
  // generated at «timestamp()»
«ENDPROTECT»
  public class «className()» {
    «FOREACH attribute AS a»
      private «a.type» «a.name»;
      public void «a.setterName()»( «a.type» value ) {
        this.«a.name» = value;
      }
      public «a.type» «a.getterName()»() {
        return this.«a.name»;
      }
    «ENDFOREACH»
  }
«ENDFILE»
«ENDDFINE»

```

تصویر ۱۰- نمونه قالب های ابزار xpanse برای تولید کد

۶-۲- استفاده از کتابخانه ی dom4j برای پردازش فایل های xml

برای تولید مدل EMF از روی مدل های توصیف شده با فایل های xml از کتابخانه dom4j استفاده

کردم و مدل EMF معادل خروجی ساخته شد.

۷- نتیجه گیری

کارآموزی در شرکت اعوان تجربه موفقی برای ورود به صنعت نرم افزار برای من بود. من با کارآموزی در اعوان، با رویکرد مهندسی در تولید نرم افزار، آن هم نرم افزارهای سازمانی بزرگ آشنا شدم و برای نخستین بار کاربرد مطالب نظری آموخته شده مهندسی نرم افزار را در صنعت دیدم. رویکردهایی مثل معماری مدل محور، تست نرم افزار، نسخه بندی نرم افزار، استفاده از ابزارهای برنامه نویسی گروهی، ساخت خودکار، تولید گزارش، مدیریت پروژه و پیگیری مسئله.

وظیفه دوره ی کارآموزی من بهبود بخش تولید کد در نرم افزار wise شرکت اعوان بود. تولید کد در راستای معماری مدل محور و از رویکردهای رایج مهندسی نرم افزار است. در این رابطه، مدلسازی با EMF Ecore را یاد گرفتم و مدل میانی پروژه های شرکت اعوان را با آن توصیف کردم. هم چنین برنامه ای نوشتم که مدل توصیف شده با فایل های موسوم به gen.xml را به مدل EMF تبدیل کند. فرایند تولید کد با Xpand و چگونگی اجرای آن با maven را نیز فرا گرفتم. نتایج همه ی این کارها در پروژه ی ir.asta.wise.generator.plugin وارد شد.

در پایان از آقای مهندس پیشوایی، سرپرست کارآموزی به خاطر راهنمایی ها و راهکارهایشان تشکر می کنم. از دکتر حیدرنوری، استاد کارآموزی هم به خاطر تلاش ها، پیگیری ها و یادآوری هایشان از تعریف تا پایان کارآموزی سپاسگزارم.

“If you can dream it, you can do it.”

Walt Disney

مراجع

- [۱] جلال، علی، «مفاهیم و ابزارها در معماری مدل محور»، همایش رقابت و فراگیری جاوا (JCAL)، دانشگاه علوم و فنون مازندران، آبان ۸۸.
- [۲] محصولات شرکت اعوان [آنلاین]، در دسترس:
http://www.asta.ir/avan/index.php?option=com_content&view=article&id=66&Itemid=78، شرکت مشاوران نرم‌افزاری اعوان، [تاریخ دسترسی: ۱۱ مرداد ۹۳]
- [3] The Eclipse Foundation, “Eclipse Modeling – EMF - Home.” [Online]. Available:
<https://www.eclipse.org/modeling/emf/>, Eclipse, [Accessed: 30-Aug-2014].
- [4] The Eclipse Foundation, “Xpand - Eclipsepedia.” [Online]. Available:
<https://wiki.eclipse.org/Xpand>, Eclipse, [Accessed: 30-Aug-2014].