

آشنایی با مفاهیم اولیه مدلسازی شیء گرا و UML

ساعدی، مهدی؛ "آشنایی با مفاهیم اولیه مدل سازی شیء گرا و UML؛ مجله رایانه؛ سال دهم؛ شماره ۹۲؛ فروردین ماه ۱۳۷۹؛ صفحه ۱۱۵-۱۱۸.

خلاصه

در این مقاله ابتدا با تاریخچه پیدایش مدلسازی شیء گرا و UML آشنا می شوید. سپس به این پرسش پاسخ داده می شود که "چرا مدلسازی می کنیم؟". در ادامه با کلیات دیدگاه شیء گرا و عناصر تشکیل دهنده آن نظیر: شیء، کلاس، صفت، عمل، ارتباط، تعمیم، تناظر، ترکیب، تجمع، وابستگی و ... آشنا می شویم. در انتها نیز آینده UML و ادعاهای آن و لزوم حرکت به سمت آن را خواهیم دید.

تاریخچه

دیدگاه شیء گرا از اواسط دهه ۷۰ میلادی در مباحث برنامه نویسی کامپیوتر متولد شد. پس از گذشت چند سال و در اوایل دهه ۹۰ به جهت ناکارآمدی روش های سنتی در مباحث تحلیل و طراحی سیستم های اطلاعاتی و کامپیوتری و نیز ظهور سیستم هایی که مدل کردن آنها به روش های سنتی بسیار ناقص بود، تحلیلگران و طراحان سیستم به این فکر افتادند تا از دیدگاه شیء گرا علاوه بر برنامه نویسی، در زمینه تحلیل و طراحی سیستم نیز استفاده کنند.

مباحث مطرح شده در این زمینه تا اواخر دهه ۹۰ در حال تغییرات جدی و سیر تکاملی خود قرار داشت و در سال های ۹۲ به بعد بود که دیدگاه های مختلف این مبحث به نقاط مشترک خوبی نزدیک شد. در طی سال های قبل از این، روش شناسی های مختلفی توسط دانشمندان شیء گرا به دنیا عرضه شده بود و هر کس به فراخور علاقه خود یکی از آنها را انتخاب می نمود و طبق آن عمل می کرد. از جمله این روش شناسی های شیء گرا می توان مواردی نظیر: روش شناسی BON، روش شناسی Booch، روش شناسی Coad/Yourdon، روش شناسی Fusion، روش شناسی Hodge/Mock، روش شناسی Jacobson، روش شناسی OMT، روش شناسی OSA، روش شناسی Rumbaugh و روش شناسی Yourdon را برشمرد. شایان ذکر است که پایه و اساس همه این روش ها یکسان است و

۱) Object-Oriented که در برخی متون به موجودیت گرا و موضوع گرا نیز ترجمه شده است.

۲) نظیر سیستم های ماهواره ای و سیستم های بلادرنگ (Real-time).

۳) Methodology. تعریف: روش شناسی، فرایندی ساختارمند برای حل یک مساله می باشد که بوسیله مجموعه ای از ابزارها (Tools) و فنون (Techniques) حمایت می گردد [Norman-96]. تعریف دیگر روش شناسی عبارت است از نمادگذاری باضافه یک روش مدون و مشخص برای ساخت مدل یا طرح سیستم [Ramsin].

تفاوت عمده در تعییرات و علائم به کار گرفته شده توسط هر یک از این روش ها می باشد. علاوه بر آن هر یک از این روشها فنون خاصی را برای مدلسازی ممکن است بکار گیرند که در روشهای دیگر موجود نباشد. اما در سال ۹۶، آقایان Booch, Rumbaugh و Jacobson در کنار هم گرد آمدند و پایه های "زبان مدلسازی یکنواخت" (Unified Modeling Language) معروف به UML را ایجاد کردند. این زبان در سال ۹۷ توسط گروه مدیریت شیء (Object Management Group) معروف به OMG در آمریکا به عنوان یک استاندارد پذیرفته شد و شرکت های مشهوری نظیر: Oracle, Microsoft و Hewlett-Packard آنرا پشتیبانی نمودند. [Alhir-98]

چرا مدلسازی می کنیم؟

مدلسازی یک سیستم نرم افزاری با قدرت صنعتی قبل از ساخت یا نوسازی بسیار ضروری است. همانطور که برای یک ساختمان بزرگ، نقشه اولیه معروف به Blue print می سازند. مدل های خوب، برای ارتباط افراد در گروه های پروژه یا یکدیگر و نیز اطمینان از قوت معماری بسیار ضروری است. ما مدل هایی از سیستم های پیچیده را بدلیل آنکه نمی توانیم کل سیستم را یکجا درک کنیم ایجاد می کنیم. همانگونه که پیچیدگی سیستم ها افزایش می یابد، بنابراین اهمیت تکنیک های مدلسازی خوب نیز بیشتر می شود. عوامل متعدد دیگری برای موفقیت پروژه وجود دارد که داشتن یک زبان استاندارد مدلسازی یکی از عوامل ضروری است و این همان چیزی است که UML فراهم می کند.

کلیات

دیدگاه شیء گرا در اصل نگرشی جدید به دنیا و سیستمهاست. این دیدگاه سعی دارد تا با نگرش خود به عناصر یک سیستم، کل آن سیستم را مدلسازی کند. دیدگاه شیء گرا بر مباحثی نظیر: شیء (Object)، کلاس (Class)، مسئولیت (Responsibility)، عمل (Operation) و سناریو (Scenario) استوار است و جهان را به صورت مجموعه ای از اشیاء به هم مرتبط می بیند که هر شیء، رفتار خاص خود و مسئولیت خاص خود را دارد. تعریف مختصر این عناصر در زیر آورده شده است:

۱) شیء (Object): انسان، مکان یا هر چیزی نظیر: دانشجو، استاد، وسیله نقلیه، خانه، اتاق، میز، خودکار و ... یک شیء است.

۲) مسئولیت (Responsibility): چیزی است که به شیء اختصاص داده می شود و سه جنبه دارد:

۱-۲) آنچه که شیء راجع به خودش می داند، که به آنها صفت

(Attribute) گفته می شود. به عنوان مثال شیء "دانشجو"، نام، نام

خانوادگی، نام پدر، شماره شناسنامه، شماره دانشجویی و ... را راجع به خودش می داند. به هر یک از این موارد یک صفت گفته می شود.

۱) در روش شناسی های مختلف، از کلمات مختلفی برای "عمل" استفاده شده است، نظیر: روش (Method)، خدمت (Operation). هم اکنون در UML از کلمه "عمل"، معادل Operation برای این تعبیر استفاده می گردد.

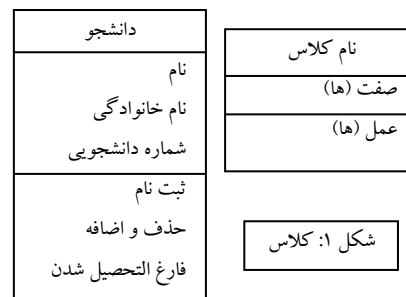
۲-۲) **کسانی را که شیء می شناسد.** در حوزه مساله اشیاء زیادی وجود دارد. تمام کسانی (اشیائی) را که شیء مورد نظر، آنها را می شناسد، با آنها **ارتباط (Relationship)** برقرار می کند. به عنوان مثال شیء "دانشجو" بر اساس رابطه "دانشجو در درس ثبت نام می کند"، شیء "درس" را می شناسد و با آن ارتباط دارد. در ادامه انواع ارتباطات درون یک مدل شیء گرا تعریف می گردد.

۳-۲) **کارهایی را که شیء انجام می دهد.** به عنوان مثال شیء "دانشجو" کارهایی نظیر: "ثبت نام، حذف و اضافه، فارغ التحصیل شدن و ..." را انجام می دهد. به هر یک از این موارد یک **عمل (Operation)** گفته می شود.

۳) **سناریو (Scenario):** عبارت از یک مجموعه عمل پشت سر هم یا متوالی می باشد که منجر به انجام یک کار خاص می گردد. به عنوان مثال شیء "دانشجو" دارای سناریو "ثبت نام در نیم سال تحصیلی" می باشد که توالی اعمال زیر می باشد: "دریافت برگ ثبت نام از آموزش، سپس انتخاب دروس نیم سال جدید و درج در برگ ثبت نام، و پس از آن دریافت تایید و امضای استاد راهنما و نیز تایید و مهر دانشکده و در نهایت تایید اداره آموزش و دریافت برگ ثبت نام توسط اداره آموزش".

۴) **کلاس (Class):** مجموعه اشیائی که دارای صفات، اعمال و ارتباطات یکسان هستند، در یک کلاس قرار می گیرند و هر یک از این اشیاء به عنوان نمونه ای (instance) از آن کلاس به حساب می آیند. به عنوان مثال کلاس "دانشجو" دارای صفات "نام، نام خانوادگی، شماره دانشجویی، ..." و اعمال "ثبت نام، حذف و اضافه، فارغ التحصیل شدن، ..." می باشد. اشیاء یا نمونه هایی از این کلاس، دانشجویان زیر می باشند:

"مهدی، ساعدی، ۷۸۱۷۵۱۰۰۵، ..."، "حمید رضا، صحراگرد، ۷۵۳۲۶۰۰۱، ..."، "عباس، عطاری، ۷۷۳۲۰۰۶۲، ..."؛ که هر یک از این دانشجویان نیز به صورت مستقل دارای اعمال فوق می باشند. نحوه نمایش یک کلاس به این صورت است که از یک مستطیل که به سه قسمت تقسیم شده است به "شکل ۱" استفاده می گردد. در قسمت بالای مستطیل، نام کلاس، در قسمت میانی، صفات و در قسمت پایینی، اعمال نوشته می شود. کلاس "دانشجو" مثالی از این دست می باشد.



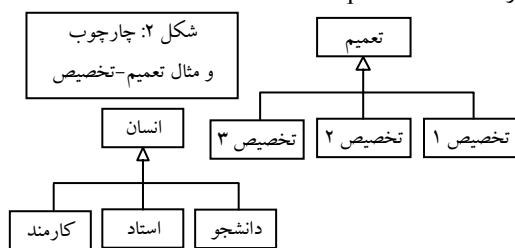
انواع ارتباطات درون یک مدل شیء گرا

ارتباطات در دیدگاه شیء گرا را به طور کلی می توان به سه دسته تقسیم کرد، اما علاوه بر این، ارتباطات دیگری نیز در UML بیان شده است که در ادامه توضیح داده می شوند. این سه دسته ارتباط عبارتند از: تعمیم

(Genralization)، وابستگی (Dependency) و تناظر (Association). این ارتباطات به خوبی برای پیاده سازی شدن توسط زبانهای برنامه نویسی شیء گرا تطابق دارند. UML برای هر یک از انواع ارتباطات، نمایش گرافیکی مناسبی را ارائه می کند.

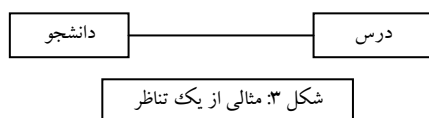
۱) تعمیم (Generalization)

تعمیم ارتباطی است که بین یک چیز عمومی با چند نوع خاصتر از آن چیز عمومی برقرار می گردد. این نوع ارتباط یکی از پایه ای ترین ارتباطات شیء گراست که در اکثر نمودارهای UML نیز کاربرد دارد. این ارتباط برای پرهیز از تکرار صفات، اعمال و ارتباطات بین کلاس ها در یک نمودار بکار می رود. برای این منظور می توان عناصر تکراری فوق را در کلاسی مجزا به عنوان کلاس **تعمیم** و عناصر غیر مشترک را در خود کلاسها قرار داد و با برقراری ارتباط تعمیم بین هر یک از کلاسها و کلاس تعمیم، کلیه عناصر فوق را از کلاس تعمیم به ارث گرفت. به عنوان مثال در سیستم یک دانشگاه، کلاس های "دانشجو"، "استاد" و "کارمند" وجود دارد که صفات "نام، نام خانوادگی، ..." بین آنها مشترک است. ما می توانیم بجای تکرار این صفات در هر یک از کلاس های فوق، کلاسی جدید با نام "انسان" ایجاد کنیم که شامل کلیه صفات، اعمال و ارتباطات مشترک کلاس های فوق است. در این حالت به کلاس "انسان"، **تعمیم** و به کلاس های دیگر، **تخصیص** گفته می شود. به "شکل ۲" و علامت تعمیم توجه کنید. **کلاس های تخصیص، کلیه صفات، اعمال و ارتباطات کلاس(های) تعمیم را به ارث (inherit)** می برند. این خاصیت معروف به خاصیت **ارث بری یا وراثت (inheritance)** می باشد. نامهای دیگری نیز برای کلاسهای تعمیم-تخصیص کاربرد دارد، از جمله: فوق کلاس-زیر کلاس (superclass-subclass) و والد-فرزند (parent-child).



۲) تناظر (Association)

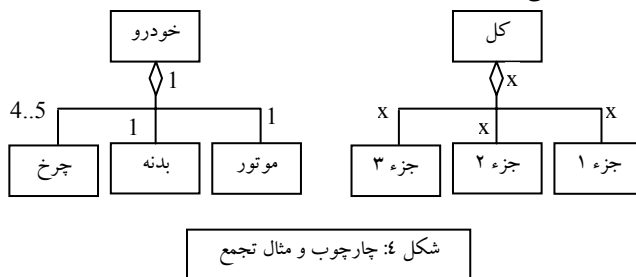
وقتی که دو شیء ارتباط ساختاری با یکدیگر دارند این نوع ارتباط کاربرد دارد. به عنوان مثال در سیستم دانشگاه، اشیاء یا کلاسهای "دانشجو" و "درس" با یکدیگر ارتباط ساختاری دارند. به این معنا که این دو کلاس از داده ها (صفات) یا اعمال یکدیگر استفاده می کنند. به عنوان مثال دانشجو از عمل "ثبت نام در درس" از کلاس "درس" استفاده می کند. برای نمایش این نوع ارتباط در UML از یک خط بین دو کلاس استفاده می شود. به "شکل ۳" توجه کنید.



در حالت خاص در یک ارتباط تناظر می تواند بروز کند که به خاطر متداول بودن، به صورت مجزا و با علامت گذاری خاص بیان شده اند. این دو حالت خاص از تناظرها عبارتند از: تجمع و ترکیب، که در زیر توضیح داده می شود.

۱-۲) تجمع (Aggregation)

بسیاری از اشیاء از به هم پیوستن چند قسمت (part) به وجود آمده اند. در این حالت برای آنکه نشان دهیم مجموعه چند قسمت تشکیل یک کل می دهند، از ارتباط تجمع استفاده می کنیم. به عنوان مثال، یک "خودرو" از "چهار چرخ"، "یک موتور" و "یک بدنه" تشکیل شده است. به "شکل ۴" و علامت تجمع (لوزی توخالی) و نیز اعداد توجه کنید.

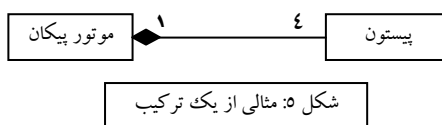


شکل ۴: چارچوب و مثال تجمع

۲-۲) ترکیب (Composition)

ترکیب نوع خاصی تجمع است که وقتی اشیاء دیگر سیستم بخواهند با اجزاء یا قسمتهای تشکیل دهنده ترکیب ارتباط برقرار کنند، مستقیماً قادر به برقراری این نوع ارتباط نیستند و فقط می توانند با ترکیب مرتبط شوند و این کلاس ترکیب است که اطلاعات اجزایش را در صورت صلاحدید در اختیار دیگران می گذارد. مسئولیت ایجاد یا از بین بردن اجزاء نیز به عهده ترکیب است. در UML برای نمایش چنین حالتی از لوزی توپر استفاده می شود.

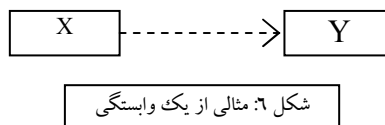
به "شکل ۵" توجه کنید:



شکل ۵: مثالی از یک ترکیب

۳) وابستگی (Dependency)

از دیگر ارتباطات درون یک مدل شیء گرا "وابستگی" است. وقتی که می گوئیم کلاس X به کلاس Y وابسته است بدین معناست که اگر در هر یک از اعمال Y تغییری صورت پذیرد، می تواند باعث ایجاد تغییری در X گردد. این ارتباط وقتی برقرار است که اساساً نتوان بین دو کلاس X و Y ارتباطی از نوع انجمن یا تناظر برقرار کرد. مشخصاً زمانی که Y به عنوان پارامتر یکی از عملهای X به این کلاس ارسال می گردد، حتماً X به Y وابسته است. به "شکل ۶" توجه کنید.



شکل ۶: مثالی از یک وابستگی

آینده مدلسازی در دستان UML

تعریفی که UML از خود بیان می دارد عبارت است از: "زبانی برای مشخص سازی (Specifying)، مجسم سازی (Visualizing)، ساخت

(Construction) و مستند سازی (Documenting) دست آوردهای سیستم های نرم افزاری، و مدلسازی کسب و کار (Business) و دیگر سیستم های غیر نرم افزاری". UML بر این نکته تاکید دارد که می تواند برای مدلسازی سیستم های غیر نرم افزاری نیز کارآمد باشد. UML مجموعه ای از بهترین تجربیات مهندسی که موفقیتشان در مدلسازی سیستم های بزرگ و پیچیده به اثبات رسیده است را عرضه میدارد. UML، یک زبان مدلسازی را مشخص می کند که اتفاق نظر جماعت شیء گرا (OO) بر مفاهیم اساسی مدلسازی است.

در UML برای ایجاد مدلها و نمودارهای حوزه مسأله هیچ توصیه ای نمی شود بلکه تنها تجربیات و یادگیری افراد است که تشخیص استفاده از کدام نمودارها و مدلها را به ایشان می دهد. در پایان صرفاً به نام نمودارهای گرافیکی که در UML ارائه شده است اکتفا می کنیم و علاقمندان را به مطالعه مراجع UML دعوت می نمایم:

- نمودار مورد کاربرد (Use Case Dia.)
- نمودار کلاس (Class Dia.)
- نمودارهای رفتار: (Behavior Dia.)
- نمودار حالت (State chart Dia.)
- نمودار فعالیت (Activity Dia.)
- نمودارهای تعاملی: (Interaction Dia.)
- نمودار توالی (Sequence Dia.)
- نمودار همکاری (Collaboration Dia.)
- نمودارهای پیاده سازی (Implementation Dia.)
- نمودار اجزاء (Component Dia.)
- نمودار استقرار (Deployment Dia.)

مراجع

1. [Norman-96]
Ronald J. Norman, *Object-Oriented Systems Analysis and Design*, Prentice Hall, 1996.
2. [Ramsin]
رامان رامسین، جزوه درس "طراحی شیء گرای سیستمها"، دانشگاه صنعتی شریف، دانشکده مهندسی کامپیوتر
3. [Booch-98]
Grady Booch, James Rumbaugh, Ivar Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, 1998.
4. [Jacobson-99]
Ivar Jacobson, Grady Booch, James Rumbaugh, *The Unified Software Development Process*, Addison-Wesley, 1999.
5. [Alhir-98]
Sinan Si Alhir, *UML in a Nutshell*, O'Reilly, 1998.
6. [Fowler-97]
Martin Fowler, Kendall Scott, *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley, 1997.
7. [OMG]
OMG Unified Modeling Language Specification, Object Management Group, Internet: www.omg.org