

## فصل ۲۱ تحلیل شیءگرا (OOA)

مفاهیم کلیدی (مرتب بر حروف الفبا)

اجزاء مدل OOA ، استفاده مجدد ، بسته ها ، تحلیل دامنه ، زده بندی کلاسها ، زیرسیستم ها ، مدل سازی CRC ، مدل شیء - رابطه ، مدل شیء - رفتار ، مسئولیت ها ، نمودارهای کلاس ، همکاری ، زبان مدل سازی یکپارچه (متحد) ، موارد کاربرد (usecase)

### KEY CONCEPTS

class diagrams , class taxonomy , collaboration , CRC modeling , domain analysis , object-behavior model , object-relationship model , OOA model components , responsibility , package , reuse , subsystems , UML , use-cases

### نگاه اجمالی

تحلیل شیءگرا چیست؟ قبل از این که بتوانید یک سیستم شیءگرا بسازید، باید کلاس‌هایی را تعریف کنید که نمایانگر مسئله مورد بررسی، شیوه‌ای که با آن کلاس‌ها با یکدیگر ارتباط داشته و واکنش متقابل دارند، کارهای داخلی یک شیء و مکلیزم‌های ارتباطی که به آنها اجازه می‌دهد با هم کار کنند. همه این چیزها در طول تحلیل شیءگرا حاصل می‌گردند.

چه کسی این امر را عهده دار است؟ تعریف مدل تحلیل شیءگرا در برگزیده توصیفی از مشخصه‌های ایستا و پویا کلاس‌هایی است که یک سیستم یا محصول را توصیف می‌کنند. این کار توسط مهندس نرم‌افزار صورت می‌گیرد.

چرا تحلیل شیءگرا از اهمیت برخوردار است؟ شما نمی‌توانید نرم‌افزار را بسازید (شیءگرا یا غیره) بدون آنکه درک معقولی از سیستم یا محصول داشته باشید. تحلیل شیءگرا شیوه عینی نمایش درک نیازمندیها را از جانب شما مهیا کرده و سپس مورد آزمون قرار می‌دهد تا معلوم شود که شناخت در برابر "درک مشتری از سیستم" ایجاد شده یا خیر؟

مراحل کار چیست؟ تحلیل شیءگرا با توضیحی از موارد کاربردی آغاز می‌شود، توصیفی مبتنی بر خلاصه برنامه در مورد این که چگونه این بازیگران یعنی افراد، دستگاه و سیستم‌های دیگر با محصولی که قرار است ساخته شود ارتباط متقابل داشته باشند. مدل سازی کلاس - مسئولیت - همکاری (CRC)

اطلاعات انجام گرفته و موارد کاربرد را به نمایشی از کلاس ها و همکاری شان با دیگر کلاس ها انتقال می دهد.

مشخصات ایستا و پویا کلاس ها با استفاده از زبان مدل سازی یکپارچه ای مدل سازی می شود. محصول کار چیست؟ مدل تحلیل شیء گرا ایجاد شده است. مدل تحلیل شیء گرا متشکل از نمایش های گرافیکی و مبتنی بر زبان است که صفات خاصه کلاس، ارتباطات و رفتارها، روابط متقابل و تصویرگری رفتار کلاس را در طول زمان نشان می دهد.

چگونه مطمئن شوم که کارم را درست انجام داده ام؟ در هر مرحله، عناصر مدل تحلیل شیء گرا از نظر وضوح، درستی، کامل بودن و هماهنگی با نیازمندیهای مشتری و دیگر قسمت ها مورد بازبینی قرار می گیرند.

وقتی سیستم یا محصول جدیدی ساخته می شود، چگونه آن را تشریح کنیم که در مهندسی نرم افزار شیء گرا قابل بررسی باشد؟ آیا سوالات خاصی وجود دارند که باید از مشتری پرسیده شوند؟ اشیای مربوطه کدامند؟ چگونه آنها را به یکدیگر ربط می دهیم؟ اشیای چگونه در بستر سیستم رفتار می کنند؟ چگونه مسئله ای را مشخص یا مدل سازی کنیم به طوری که یک طراحی مؤثر خلق کنیم؟

هر یک از این سوالات در بستر تحلیل شیء گرا (OOA) پاسخ داده می شوند، یعنی اولین کار فنی که به عنوان بخشی از مهندسی نرم افزار شیء گرا صورت می گیرد. به جای بررسی مسئله با استفاده از مدل جریان اطلاعاتی کلاسیک، تحلیل شیء گرا چندین مفهوم جدید را معرفی می نماید. کد و یوردون این موضوع را زمانی بررسی می کنند که می نویسند:

تحلیل شیء گرا بر پایه مفاهیمی است که ابتدا ما در مهد کودک آنها را آموخته ایم یعنی: اشیای و صفات خاصه، کلاس بندی ها و اعضا، کل و جزء، چرا این همه مدت زمان بزرگ تا ما این مفاهیم را در تحلیل و مشخص سازی سیستم های اطلاعاتی به کار ببریم، حدس می زنیم ...

OOA در مجموعه ای از مفاهیم و اصول مقدماتی پایه گذاری شده که در فصل ۱۱ معرفی شده بودند. به منظور ایجاد یک مدل تحلیلی باید ۵ اصل مقدماتی به کار گرفته شوند: (۱) میدان اطلاعات مدل سازی می شود. (۲) کارکرد توصیف می شود (۳) رفتار نمایان می گردد. (۴) مدل های داده ای، کارکردی و رفتاری تقسیم بندی می شوند تا در برابر جزئیات بیشتری قرار گیرند و (۵) مدل های اولیه نمایان گر اصل مسئله است در حالی که مدل های بعدی جزئیات اجرایی را آشکار می سازند. این اصول بنیان رهیافت تحلیل

**نقل قول**

شیء گرا در این فصل را، تشکیل می دهند.

یک مسئله "خوب تعریف شده" یک مسئله "نیمه منظور از OOA عبارتست از تعریف تمام کلاس هایی که با مسئله ای که حل می شود در ارتباطند - حل شده" است. چارلز عملیات و صفات خاصه مربوط به آنها و ارتباطات ملین آنها و رفتاری که از خود بروز می دهند. برای کترینگ حصول این امر، چند کار باید صورت گیرد:

- ۱- نیازمندیهای اولیه کاربر باید بین مشتری و مهندس نرم افزار مورد تبادل قرار گیرد.
  - ۲- کلاس ها باید شناسایی شوند (مثل صفات خاصه و روش هایی که تعریف شده اند).
  - ۳- سلسله مراتب کلاس باید مشخص می شوند.
  - ۴- روابط شیء با شیء نمایان گردند. (ارتباطات شیئی)
  - ۵- رفتار شیء مدل سازی می شود.
  - ۶- وظایف ۱ تا ۵ به صورت تکراری به کار گرفته می شوند تا وقتی که مدل تکمیل گردد.
- نکته مهم مورد توجه این است که توافق همگانی در مورد مفاهیمی وجود دارد که به عنوان پایه و اساس OOA عمل می کنند. اما چند ایده اصلی محدود به صورت مرتب تکرار می شوند و این ایده ها همان هایی است که ما در این فصل بررسی می کنیم.

### ۱-۲۱ تحلیل شیء گرا

هدف از تحلیل شیء گرا، ارائه مدلی است که نرم افزاری را توصیف می کند که اجزای آن، نیازمندیهای معین شده از جانب مشتری را مرتفع می سازد. OOA مانند روش های تحلیلی قراردادی که در فصل ۱۲ توصیف شدند، یک مدل تحلیل چند بخشی می سازد که این هدف را تأمین می کنند. مدل تحلیلی، اطلاعات، کارکرد و رفتار موجود در بستر عناصر مدل شیء، توصیف شده در فصل ۲۰ را به تصویر درمی آورد.

#### ۱-۱-۲۱ رهیافت های متعارف در برابر رهیافت های شیء گرا

آیا تحلیل شیء گرا واقعاً از رهیافت تحلیل ساختنیافته که در فصل ۱۲ ارائه شده متفاوت است؟ فیچمن و کمر این سؤال را به این صورت مورد خطاب قرار داده اند: [FIC92]

ما رهیافت تحلیل شیء گرا را به پایان رسانیدیم ... که باز نمایی از یک تغییر بنیادی نسبت به دیگر متدولوژی ها نظیر تحلیل ساختنیافته می باشد، اما در برابر متدولوژی های مبتنی بر داده مانند مهندسی اطلاعات تنها افزودن تغییراتی چند محسوب می گردد. متدولوژی های مبتنی بر فرایند، بر خاصیت های ذاتی و اصلی اشیاء طی فرایند مدل سازی متمرکز نمی باشند و حوزه مسئله را به گونه ای مدل می کنند که بر اصولی اساسی سه گانه شیء گرایی استوار است: بسته بندی، رده بندی اشیاء، و توارث.

به بیان ساده، تحلیل ساختنیافته (SA) نیازمندیها را به صورت ورودی - پردازش - خروجی می بیند. داده ها مجزا از فرآیندهایی که تبدیل آنها را عهده دارند در نظر گرفته می شوند. رفتار سیستم، هر قدر مهم باشد، در تحلیل ساختنیافته از اولویت کمتری برخوردار است. رهیافت تحلیل ساختنیافته در تفکیک و تجزیه کارکردی تواناست (تقسیم بندی نمودار جریان داده ها، فصل ۱۲).



کنام معیارهای اصلی  
برای مقایسه مدل های  
متعارف و OO به کار  
می روند؟

فیشمن و کمرر [FIC92] ۱۱ "ابعاد مدل سازی" را پیشنهاد نموده اند که شیوه های مختلف تحلیل ساخته شده و و تحلیل شیء گرا را مقایسه می نماید:

- ۱- شناسایی / رده بندی موجودیت ها
- ۲- رابطه های عام-به-خاص و کل-به-جزء روابط موجودیت ها
- ۳- دیگر روابط موجودیت ها
- ۴- توصیف صفات خاصه موجودیت ها
- ۵- تفکیک و قسمت بندی در مقیاس بزرگ
- ۶- وضعیت ها و انتقال بین وضعیت ها
- ۷- جزئیات مشخصه های کارکردی
- ۸- تجزیه و شکستن بالا به پایین
- ۹- توالی آخر-به آخر فرایند
- ۱۰- شناسایی خدمات دوجانبه
- ۱۱- ارتباطات موجودیت ها (با پیام ها یا رویدادها)

از آنجا که طی سالها تحلیل ساخته شده از تنوع بسیاری برخوردار شده و شیوه های شیء گرای نیز متعدد گردیده است (بخش ۲۱-۲۰ را ببینید) ارائه یک مقایسه کلی بین دو شیوه امری دشوار می نماید. شاید تنها بتوان اذعان داشت که ابعاد ۸ و ۹ همواره در تحلیل ساخته شده وجود دارند ولی در تحلیل شیء گرا هرگز مورد استفاده ندارند.

## ۲۱-۱- چشم انداز تحلیل شیء گرا

محبوبیت فناوری شیء گرا باعث تولید یک دوجین شیوه تحلیل شیء گرا طی اواخر دهه ۱۹۸۰ و دهه ۱۹۹۰ گردید. هر یک از آنها فرایندی را برای تحلیل یک سیستم یا محصول پیشنهاد کردند، مجموعه نمودارهایی که نه تنها فرایند را شامل می شد بلکه نمادگذاری هایی که مهندس نرم افزار را قادر می ساخت تا مدل تحلیلی را به گونه ای سازگار ایجاد نماید. آنهایی که بیشتر مورد اقبال واقع شدند عبارتند از:

### شیوه بوج

شیوه بوج مشتمل بر یک "ریزفرایند توسعه" [BOO94] و "یک درشت فرایند توسعه" می باشد. سطح میکرو مجموعه ای از وظائف تحلیلی را تعریف می کند که در هر گام درشت فرایند باید تکرار شوند. بنابراین یک رهافت تکاملی وجود خواهد داشت. فرایند توسعه تحلیل شیء گرا بوج، کلاسها و اشیاء و معانی کلاس و شیء را تعریف می کند که به همراه رابطه میان کلاسها و اشیاء ارائه می شود. و یکسری پالایش برای قسمت بندی مدل تحلیل، تعریف می نماید.

## شیوه رامباف

رامباف [RUM91] و کالج او برای تحلیل و طراحی سیستمها و طراحی سطح شیء، یک فن مدل سازی شیء (OMT) را توسعه دادند. فعالیت تحلیل سه مدل را ایجاد می نماید: مدل شیء (یک بازنمایی از اشیاء، کلاسها، سلسله مراتب، و رابطه ها)، مدل پویا (یک بازنمایی از شیء و رفتار سیستم)، و مدل کارکردی (یک بازنمایی از جریان اطلاعات درون سیستم. شبیه به یک DFD سطح بالا).

## شیوه یا کوپسن (جا کوپسن)

که همچنین مهندسی نرم افزار شیء گرا نیز نامیده می شود (OOSE)، شیوه یا کوپسن [JAC92] نسخه پیاده ای از شیوه شیء اختصاصی است که البته توسط یا کوپسن توسعه یافته است. این شیوه با توجه به تاکید قوی که بر Use-Case - یک تعریف یا سناریو که مجاوره کاربر را با محصول یا سیستم مجسم می کند- با دیگر شیوه ها متفاوت است.

## نقل قول

فعالیت مرکزی در کار با اشیاء بر تامله سازی نیست بلکه بازنمایی است. دیوید نیلور

## شیوه کاد و یوردان

این روش اغلب به عنوان ساده ترین روش تحلیل شیء گرای از نظر یادگیری، بررسی می شود. یادداشت برداری از مدل سازی نسبتاً ساده بوده و رهنمودهای تولید مدل تحلیل صریح و ساده می باشند. شرح خلاصه ای از فرایند تحلیل شیء گرای کاد و یوردان در زیر آمده است:

- شناسایی اشیاء با استفاده از معیار «در جستجوی چه چیزی باشیم»
- تعریف ساختار تعمیم و تخصیص
- تعریف ساختار کل و جزء
- شناسایی موضوعات (نمایان گر اجزای سیستم فرعی)
- تعریف صفات خاصه

## روش ورفز - براک

این روش تمایز مشخصی بین کار طراحی و تحلیل ایجاد نمی کند. بلکه فرایند مستحضر است که با ارزیابی مشخصات مشتری آغاز شده و با طراحی خاتمه می یابد. خلاصه ای از کارهای مربوط به طراحی و تحلیل در روش ورفز - براک در ادامه آمده است: [COA91]

- مشخصه های مشتری را ارزیابی کرده.
- کلاس های کاندید شده را با استفاده از تجزیه گرامری از مشخصه ها جدا می سازد.
- در تلاش برای شناسایی کلاس های برتر، کلاس ها را گروه بندی می کند.

- مسئولیت‌های هر کلاس را مشخص می‌سازد.
- مسئولیت‌هایی را به هر کلاس تخصیص می‌دهد.
- ارتباط میان کلاس‌ها را مشخص می‌کند.
- همکاری میان کلاس را بر اساس مسئولیت‌ها تعریف می‌کند.
- نمایشی از سلسله مراتب کلاس را ایجاد می‌سازد.
- نمودار همکاری را برای سیستم ایجاد می‌کند.

گرچه اصطلاحات و مراحل کار در مورد هر یک از روش‌های OOA با هم فرق می‌کنند، اما فرآیندهای کلی آن کاملاً مشابه یکدیگرند. به منظور انجام تحلیل شیء گرا، مهندس نرم افزار مراحل کلی زیر را باید انجام دهد:

- ۱- نیازمندیهای مشتری را از نظر سیستم پیدا کند.
  - ۲- سناریوها یا موارد کاربردی را شناسایی کند.
  - ۳- با استفاده از نیازمندیهای اولیه به عنوان یک راهنما، کلاس‌ها و اشیا را انتخاب کند.
  - ۴- صفات خاصه و عملیات را برای هر یک از اشیای سیستم شناسایی کند.
  - ۵- ساختار و سلسله مراتب نوزمان دهنده کلاس را تعریف کند.
  - ۶- یک مدل ارتباط شیئی را بسازد.
  - ۷- یک مدل حالت و رفتار شیئی را بسازد.
  - ۸- مدل تحلیل شیء گرا را در مقایسه با موارد کاربرد/سناریوها بازبینی کند.
- این مراحل کلی با جزئیات دقیق تری در بخش‌های ۲۱-۳ و ۲۱-۴ بررسی شده‌اند.



یک مجموعه گامهای  
عمومی طی آنالیز  
(تحلیل) شیء گرا  
(OOA) مرفظ از  
شیوه تحلیل انتخاب  
شده، خواهد بود.

### ۲۱-۳-۱ یک رهیافت یکتواخت برای تحلیل شیء گرا

در طول دهه گذشته گزادی بوج، جیمز رامباخ و ایوار جاکوسن با همکاری یکدیگر بهترین مشخصه‌های روش‌های طراحی و تحلیل شیء گرا را در یک روش متحدالشکل، ترکیب کرده‌اند. نتیجه آن که «زبان مدل سازی متحد یا UML» است به طور گسترده در صنعت استفاده شده است.<sup>۱</sup> UML به مهندس نرم افزار امکان می‌دهد تا یک مدل تحلیل را با استفاده از علائم مدل سازی که تحت کنترل یک سری قواعد نحوی، معنایی و عمل گرا هستند، ارائه دهد. این کسوت و بنکر این قواعد را به شکل زیر توصیف می‌کنند: [ERI98]

نحو به ما می‌گوید که علائم چگونه باید باشند و چگونه ترکیب شوند. این نحو با کلمات در زبان طبیعی مقایسه می‌شود. نکته مهم این است که بدقت چگونه آنها را به درستی هجی کرده و کلمات مختلف



UML برخی علائم  
حاضر OO را یکسان  
می‌سازد. بنابراین یک  
نقطه منفرد از ارجاع  
ها برای بسیاری از  
مفاهیم خواهد بود.  
بیشتر هروشکا

۱. بوج، رامباخ و جاکوسن (ژاکوسن) مجموعه سه کتاب توصیفی در خصوص UML تالیف نموده‌اند خوانندگان مشتاق می‌توانند به [BOO99]، [RUM99] و [JAC99] مراجعه نمایند.

را در کنار هم قرار دهیم تا یک جمله تشکیل دهند. قواعد معنایی به ما می‌گویند که معنی هر نماد چیست و چگونه باید آن را از نظر خود و در متن سایر نشانه‌ها تفسیر کرد؟ آنها با معنای لغات در یک زبان طبیعی مقایسه می‌شوند.

قوانین عمل‌گرایانه هدف از نشانه‌ها را تعریف می‌کنند که از طریق آن هدف از مدل حاصل شده و برای دیگران قابل درک می‌شود. این عمل در زبان طبیعی با قواعد ساخت جمله مرتبط است که مشخص و قابل درک هستند.

در UML سیستمی که ارائه می‌شود از پنج دیدگاه مختلف استفاده می‌کند که سیستم را از جهات کاملاً متفاوتی توصیف می‌کنند. هر دیدگاه توسط مجموعه‌ای از نمودارها ارائه و تعریف می‌شود. دیدگاه‌های زیر در UML ارائه شده‌اند:

دیدگاه مدل کاربر. این دیدگاه نمایان‌گر سیستم از نقطه نظر کاربر است (که در UML به آنها هنرپیشه، بازیگر یا اکتور می‌گویند). مورد کاربری روش مدل‌سازی انتخابی است برای دیدگاه مدل کاربر. این نمایش مهم تحلیلی طرح کاربر را از دیدگاه کاربر نهایی توصیف نموده و به صورت جزئیات در فصل ۱۱ مورد بحث قرار گرفته است.<sup>۲</sup>

دیدگاه مدل ساختاری. که در آن اطلاعات و قابلیت عمل از داخل سیستم بررسی می‌شود. یعنی ساختار ایستا (کلاس‌ها، اشیا و ارتباطات) مدل‌سازی می‌شود.

دیدگاه مدل رفتاری. این بخش از مدل تحلیل نمایان‌گر جنبه‌های رفتاری یا پویای سیستم است.

هم‌چنین روابط درونی یا همکاری‌های بین عناصر ساختاری مختلف که در مدل کاربر و دیدگاه‌های مدل ساختاری توصیف شده‌اند را مشخص می‌سازد.

دیدگاه مدل پیاده‌سازی. جنبه‌های رفتاری و ساختاری سیستم همان‌گونه که ساخته می‌شوند به

نمایش درمی‌آیند.

دیدگاه مدل محیطی. جنبه‌های ساختاری و رفتاری محیطی که در آن قرار است سیستم اجرا شود.

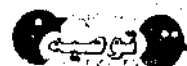
نمایش داده می‌شوند.

به‌طور کلی، مدل‌سازی تحلیل UML بر دیدگاه‌های مدل ساختاری و کاربر در سیستم متمرکز

می‌شود.

مدل‌سازی طراحی UML (که در فصل ۲۲ مورد بررسی قرار گرفته) مدل رفتاری، مدل اجرایی و

دیدگاه‌های مدل محیطی را مورد خطاب قرار می‌دهد. این امر در فصل به‌طور دقیق‌تر توصیف شده است.



مانند تمام ره‌یافت

های تحلیل، تعیین

نیازمندیها (فعالیتی)

کلیدی خواهد بود.

اطمینان حاصل کنید

که مدل کاربر را

درست دیده اید.

آسایش به دنبال آن

1 Eriksson, H.E. and M.

۲. اگر شما هم اینک (بر این امر) مسلط نیستید و آگاهی کامل کسب ننموده‌اید، لطفاً به قسمت ۱۱-۲۴ برای توصیف مفصل از Use-Case ها مراجعه و مطالعه نمایید.



## ۲-۲۱ تحلیل حوزه

تحلیل سیستم‌های شیءگرا می‌تواند در سطوح مختلف انتزاعی رخ دهد. در سطح تجارتي یا شرکتي، می‌توان فنون مربوط به OOA را با روش طراحی فرآیند تجاری در تلاش برای تعریف کلاس‌ها، اشیاء، روابط و رفتارهایی که کل کار را مدل‌سازی می‌کند، مرتبط ساخت. در سطح حوزه تجاری یک مدل شیء، که کارهای به‌خصوص حوزه تجاری را توصیف می‌کند، می‌توان تعریف نمود. در سطح کاربردی، مدل شیء روی نیازهای خاص مشتری متمرکز می‌شود، یعنی نیازمندیهایی که بر کاربرد موردنظر تأثیر می‌گذارند.

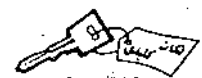
OOA در بالاترین سطح تجریدی (سطح شرکتي) فراتر از دامنه کار این کتاب است. خوانندگان علاقه‌مند برای بحث بیشتر در این مورد باید به قسمت‌های مورد اشاره، مراجعه کنند: [FIN96] و [TAY95]، [EEL98]، [CAR98]، [MAT 94] و [SUL94]. پایین‌ترین سطح این تجرید نیز در بازنگری کلی مهندسی نرم‌افزار قفل گرفته و نقطه تمرکز سایر بخش‌های این فصل است. در این بخش ما این کار را در سطح متوسط انتزاعی و تجریدی انجام می‌دهیم. این کار که تحلیل دامنه نامیده می‌شود وقتی صورت می‌گیرد که سازمانی بخواهد کتابخانه‌ای از کلاس‌های قابل استفاده مجدد درست کند که به‌طور گسترده‌ای در یک کلاس کامل از برنامه‌های کاربردی قابل اجرا باشند.

## ۲-۲۱-۱ استفاده مجدد و تحلیل حوزه

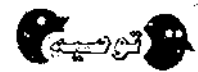
فناوری‌های مربوط به شیء از طریق استفاده مجدد دارای تأثیر و قدرت می‌شوند. مثالی ساده را در نظر بگیرید. تحلیل نیازها در مورد یک برنامه کاربردی جدید نشان‌گر این است که ۱۰۰ کلاس لازمند. دو تیم عهده‌دار ساخت این برنامه هستند. هر کدام یک محصول نهایی را طراحی و تولید می‌کنند. هر تیم دارای افرادی با سطح مهارت یکسان و تجربه یکسان می‌باشد.

تیم A به کتابخانه کلاس دسترسی ندارد و بنابراین باید تمام ۱۰۰ کلاس را از ابتدا بسازد. تیم B از یک کتابخانه توانمند کلاس استفاده کرده و هم‌اکنون ۵۵ کلاس وجود دارند. این احتمال بسیار بالا وجود دارد که:

- ۱- تیم B پروژه را زودتر از تیم A تمام سازد.
- ۲- هزینه محصول تیم B تا حد قابل ملاحظه‌ای کمتر از محصول تیم A باشد.
- ۳- محصول کاری تیم B نقایص کمتری نسبت به محصول تیم A داشته باشد.



هدف اصلی تحلیل  
حوزه ای (حیطه ای)  
تعریف مجموعه ای  
از رده ها (کلاس ها) و  
اشیاء ای است که در  
حوزه کاربردی وجود  
دارند. آنها را از آن پس  
می توان در بسیاری از  
برنامه های کاربردی  
مورد استفاده مجدد  
قرار داد.



دیگر مزایای مرتب بر  
استفاده مجدد،  
سازگاری و هم  
خانوادگی است.

الگوهای نرم افزاری  
سازگارتر شده، قابلیت  
نگهداری را افزایش می  
دهند. اطمینان حاصل  
کنید که مجموعه ای  
از "قوانین طراحی"

استفاده مجدد را برقرار  
نموده اید. آنگاه مزایای  
مذکور را به دست  
خواهید آورد.

- 1.Fingar,P.
- 2.Taylor, D. A.
- 3.Eeles, P.and O.
- 4.Carmichael
- 5.Mattison, R.



گرچه حاشیه‌ای که تیم B در آن کار می‌کند، اغلب از حاشیه تیم A تجاوز می‌کند اما در مورد موفقیت‌ها بحث‌هایی وجود دارد که تنها اندکی از آنها حکایت از این دارند که استفاده، مزایای مهم چندانی متوجه تیم B نمی‌کند.

اما این کتابخانه قدرتمند کلاس از کجا آمده است؟ چگونه معلوم می‌شود که مدخل‌های کتابخانه برای استفاده در برنامه جدید مناسبند؟ برای پاسخ دادن به این سؤالات، سازمانی که کتابخانه را ایجاد و نگهداری می‌کند باید تحلیل دامنه را به کار گیرد.

### ۲۱-۲-۲۱ فرآیند تحلیل حوزه

فایر اسمیت تحلیل دامنه نرم‌افزاری (تحلیل دامنه شیء گرا) را به شکل زیر تعریف می‌کند:

تحلیل دامنه نرم‌افزار عبارتست از شناسایی، تحلیل و تعیین مختصات و نیازهای معمول از دامنه خاص برنامه کاربردی به‌ویژه برای استفاده مجدد در یک دامنه کاربردی خاص، از نظر اشیای معمول، کلاس‌ها، مونتاژهای فرعی و چارچوب‌های کاری ...

دامنه کاربرد خاص می‌تواند از اوبوینک یا الکترونیک هواپیما تا بانکداری و از بازی‌های چند رسانه‌ای ویدیویی تا برنامه‌های کاربردی داخل وسیله MRI را دربرگیرد. هدف از تحلیل دامنه بسیار ساده و واضح است: یافتن یا ایجاد کلاس‌هایی که به‌طور گسترده قابل استفاده‌اند به‌طوری‌که بتوانند مورد استفاده مجدد قرار گیرند.

با استفاده از فرهنگ داده‌ها و اصطلاحاتی که پیش‌تر در این کتاب معرفی شد، ممکن است تحلیل دامنه را به‌عنوان یک فعالیت پوششی برای فرآیند پردازش نرم‌افزار تصور نمود. با این تعریف منظورمان این است که تحلیل دامنه یک کار مهندسی نرم‌افزاری مستمر است که به هیچ پروژه نرم‌افزاری متصل و مرتبط نیست. از یک نظر، نقش تحلیلگر دامنه مشابه نقش تولیدکننده ابزار است که در یک محیط تولیدی سنگین کار می‌کند. کار این شخص تولید و طراحی ابزار است که ممکن است توسط بسیاری از مردم به‌صورت مشابه مورد استفاده قرار گیرند، اما الزاماً کارها یکسان نیست. نقش تحلیلگر دامنه عبارتست از طراحی و ساخت اجزای قابل استفاده مجدد که ممکن است توسط خیلی از افرادی که روی موضوعات مشابه اما نه الزاماً یکسان کار می‌کنند، به کار گرفته شود.



#### ارجاع به وب

یک آموزش جامع به همراه لیستی از منابع UML، مشتمل بر ابزار، مقالات و مثالها در آدرس زیر وجود دارد:

[www.mini.net/ce-tus/oo-uml.html](http://www.mini.net/ce-tus/oo-uml.html)



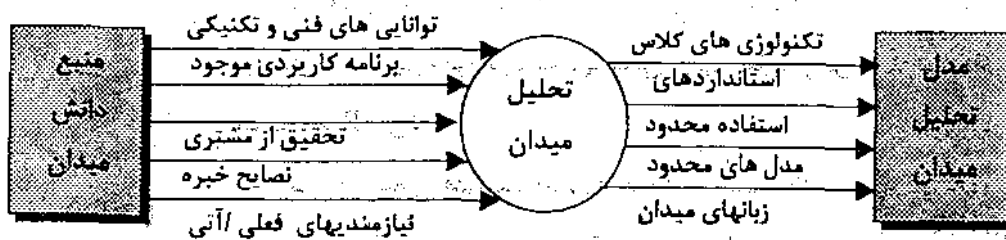
استفاده مجدد، بنیاد

مهندسی نرم‌افزار

مبتنی بر اجزاء است.

در فصل ۲۷ موضوع

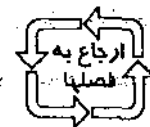
توضیح داده شده است.



شکل ۱-۲۱ ورودی و خروجی برای تحلیل میدان

شکل ۱-۲۱ [ARA89]<sup>۱</sup> ورودی ها و خروجی های اصلی و کلیدی را در مورد تحلیل دامنه تشریح می کند. منابع شناخت دامنه به منظور تلاش برای شناسایی که در طول دامنه مجدداً استفاده می شوند، مورد بررسی قرار می گیرند. در اصل تحلیل دامنه کاملاً مشابه مهندسی اطلاعات است. در مهندسی و طراحی اطلاعات و دانش، حوزه مشخصی از علایق به منظور بدست آوردن حقایق اصلی که ممکن است در ایجاد یک سیستم تخصصی یا شبکه عصبی مصنوعی به کار روند، مورد تحقیق قرار می گیرد. در طول تحلیل دامنه، استخراج شیء (و کلاس) رخ می دهد. برنارد فعالیتهای زیر را [BER93]<sup>۱</sup> پیشنهاد می کند:

دامنه مورد تحقیق را تعریف کنید. برای حصول این امر، تحلیلگر ابتدا باید حوزه کاری، نوع سیستم یا طبقه بندی محصول مورد علاقه را مجزا سازد. سپس قلم های شی گرا و غیر شی گرا استخراج می گردند. قلم های شی گرا شامل: مشخصات، طراحی ها و برنامه برای کلاس های کاربردی شی گرا موجود، کلاس های پشتیبان (مثل کلاس های رابط گرافیکی کاربر (GUI)) یا کلاس های دسترسی به بانک اطلاعاتی)، کتابخانه های اجزای غیر سفارشی تجاری که مربوط به دامنه بوده و موارد آزمونی هستند. قلم های غیر شی گرا در برگزیده سیاست کاری، رویه ها، طرح ها، استانداردها و رهنمودها، بخش هایی از برنامه های غیر شی گرا موجود (از جمله اطلاعات آزمون و طراحی و مشخصه ها، متریک ها و نرم افزارهای غیر شی گرای غیر سفارشی تجاری می باشد).



یک راهبرد کامل

تحلیل حوزه ای، باید

مماری را نیز به خوبی

اجزاء در نظر گیرد.

توضیحی مفصل از

مماری نرم افزار در

فصل ۱۴ ارائه گردیده

است.

اقدام بدست آمده از دامنه را طبقه بندی کنید. قلم های طبقه بندی شده در طبقه ها و مشخصه های تعریف کلی طبقه تعریف شده اند. طرح ردیابی این طبقات پیشنهاد شده و موارد قراردادی نام گذاری هر یک از قلم ها تعریف گردیده اند. در موقع مناسب، سلسله مراتب کلاس بندی و ردیابی ایجاد می گردند.

یک نمونه از برنامه های کاربردی در دامنه را بدست آورید. برای انجام این کار، تحلیل گر باید مطمئن شود که برنامه مورد بحث دارای قلم هایی است که متناسب کلاس هایی هستند که هم اکنون تعریف شده اند.

<sup>۱</sup> Arango

برارد بیان می‌دارد که در مراحل اولیه استفاده از فناوری‌های شیء گرا، سازمان نرم‌افزاری دارای برنامه‌های کاربردی شیء گرای بسیار محدودی است. بنابراین، تحلیل‌گر دامنه باید اشیای مفهومی را در هر برنامه (غیر شیء گرا) شناسایی کند.

هر برنامه موجود در نمونه را تحلیل کنید. مراحل زیر توسط تحلیل‌گر پی‌گیری می‌شوند:

- شناسایی اشیای کاندید قابل استفاده مجدد
- اشاره به دلایلی که اشیاء به‌خاطر آن برای استفاده مجدد شناسایی شده‌اند.
- تعریف تطبیقاتی با شیء که ممکن است قابل استفاده مجدد نیز باشد.
- تخمین درصد برنامه‌های دامنه که ممکن است از شیء استفاده مجدد کنند.
- شناسایی اشیاء با نام و استفاده از فنون مدیریت پیکربندی برای کنترل آنها علاوه بر این، وقتی اشیاء تعریف شدند، تحلیل‌گر باید درصد احتمال ساخت برنامه خاصی را با استفاده از اشیای قابل استفاده، تخمین بزند.

یک مدل تحلیل برای اشیاء توسعه دهید. مدل تحلیل به‌عنوان پایه و اساس طراحی و ساخت اشیای دامنه عمل می‌نماید. علاوه بر این مراحل، تحلیل‌گر دامنه باید مجموعه‌ای از رهنمودهای استفاده مجدد را ایجاد نموده و نمونه‌ای ارائه دهد که چگونگی استفاده از اشیای دامنه را برای تولید برنامه کاربردی جدید تشریح کند.

تحلیل دامنه اولین فعالیت فنی به مفهوم گسترده‌تر است که بعضی آن را مهندسی دامنه<sup>۱</sup> می‌نامند. وقتی دامنه محصول، سیستم یا کاری طوری تعریف شد که راهبرد آن کار در دراز مدت باشد، باید از تلاش مستمری برای خلق یک کتابخانه قدرتمند قابل استفاده مجدد، بهره گرفت. هدف، عبارتست از برخورداری از توانایی خلق نرم‌افزاری در دامنه با درصد بالایی از اجزای قابل استفاده مجدد. هزینه کمتر، کیفیت بهتر و بهبود زمان ارائه در بازار، از مباحث مورد علاقه در حوزه کار مهندسان هستند.

### ۲۱-۳ اجزاء عمومی مدل تحلیل شیء گرا

فرآیند تحلیل شیء گرا یا OOA با مفاهیم اولیه تحلیل و اصول مورد بحث در فصل ۱۱ مطابقت دارد. با این‌که فرهنگ اصطلاحات، عبارات و فعالیت‌های مورد استفاده در OOA از روش‌های موسوم متفاوت است. اما همان اهداف نهفته را مدنظر دارد. رامیاف و همکاران این مسئله را مورد بحث قرار داد. و این‌طور بیان می‌دارند که: [RUM91]<sup>۲</sup>



ارجاع به وب

یک آموزش ارزنده در

خصوص تحلیل حوزه ای،

در آدرس زیر قرار دارد:

[www.sei.cmn.edu/st/r/descriptions/deda.html](http://www.sei.cmn.edu/st/r/descriptions/deda.html)

1. Berard, E. V.

2. domain engineering

3. Jakobson, I.

تحلیل مربوط به طراحی یک مدل دقیق، مختصر، قابل درک و درست از جهان واقعی است ... هدف از تحلیل شی گرا عبارتست از مدل سازی از جهان واقعی به طوری که قابل درک باشد. برای انجام این کار باید نیازمندیها را بررسی نمود. اشاراتشان را تحلیل کرده و آنها را به صورت جدی بازگویی نمایید. باید ابتدا مشخصه های اجزای جهان واقعی را تمکیک نموده و جزئیات کوچک را تا بعد، مختصراً تمایز دهید.

برای ارائه چنین مدلی مهندس نرم افزار باید روشی را انتخاب کند که مجموعه ای از اجزای کلی یک مدل تحلیل شی گرا را پیاده سازی می کند. مونارکی و پور مجموعه ای از جزءهای عمومی نمایشی را تعریف می کنند که در همه مدل های تحلیل شی گرا ظاهر می شوند. اجزای ایستا به طور طبیعی ساخت یافته هستند و نشان گر مشخصه هایی هستند که در سراسر طول عمر برنامه حفظ می شوند. این مشخصه ها باعث تفاوت یک شیء از شیء دیگر می شود. اجزای پویا روی کنترل متمرکز شده و نسبت به زمان بندی و پردازش رویدادها حساسند. آنها تعریف می کنند که چگونه یک شیء با شیء دیگر

تداخل عمل در طول زمان دارد. اجزای زیر شناسایی شده اند: [MON92]

دیدگاه ایستا در مورد کلاس های معنایی طبق بندی کلاس های عادی در فصل ۲۰ شناخته شد. نیازمندیها برآورد شده و کلاس ها به عنوان بخشی از مدل تحلیل استخراج می گردند. این کلاس ها در تمام طول عمر برنامه پایدار مانده و براساس اصول معنایی نیازمندیهای مشتری بدست می آیند. دیدگاه ایستا در مورد صفت خاصه. دو کلاس باید منحصرأ توصیف شود. صفات خاصه مربوط به این کلاس، توصیفی از کلاس به علاوه توضیح اولیه ای از عملیاتی را مهیا می سازد که مربوط به این کلاس است.

دیدگاه ایستا در مورد روابط. اشیا به طرق مختلف با یکدیگر مرتبطند. مدل تحلیل باید نمایان گر این ارتباطات باشد. به طوری که می توان عملیات (انهایی که بر این ارتباطات اثر می گذارند) را شناسایی نموده و طراحی رهیافت پیام رسانی را بدست آورد.

دیدگاه ایستا در مورد رفتارها. ارتباطات مورد اشاره فوق مجموعه ای از رفتارها را تعریف می کند که سناریو کاربرد سیستم را تطابق می دهد. این رفتارها با تعریف یک سری عملیاتی که بدست می آوریم، پیاده سازی می شوند.

دیدگاه پویا در مورد ارتباطات. اشیا باید با یکدیگر ارتباط برقرار کرده و این کار را بر اساس یک سری حوادثی انجام می گیرند که باعث انتقال از یک حالت به حالت دیگر می شود.

دیدگاه پویا در مورد کنترل و زمان. ماهیت و زمان بندی رویدادها که باعث فعل و انفعالات میان حالات مختلف می شوند، باید توصیف گردند.



اجزاء ایستا در اجرای برنامه، تغییری نمی یابند. اجزاء پویا تحت تاثیر زمان و رویدادها می باشند.



جزء کلیدی یک مدل OOA کدام است؟

۱. نویسندگان [MON92] تحلیلی نیز از ۲۳ مدل اولیه شی گرا ارائه داده اند و تعیین کرده اند که چگونه آنان این اجزاء و مولفه ها را مشخص نموده اند.

2. Monarchi, D.E.

دوشامپ [CHA93]<sup>۱</sup> و همکارانش یک دیدگاه کاملاً متفاوت از نمایش تحلیل شیء گرا تعریف می کنند. اجزای پویا و وابسته برای روابط درونی اشیا و بازنمایی های درون خود شیء شناسایی می شوند. دیدگاه پویا در مورد روابط بین اشیا را می توان به عنوان تاریخچه طول عمر شیء توصیف کرد، یعنی حالات تغییر شیء به مرور زمان همان گونه که عملیاتی روی این صفات خاص صورت می گیرد.

## ۲۱-۴ فرآیند تحلیل شیء گرا (OOA)

فرآیند OOA با توجه به اشیا شروع نمی شود. بلکه با شناخت حالتی آغاز می گردد که در آن سیستم مورد استفاده قرار می گیرد، مثلاً توسط مردم اگر سیستم دارای ارتباط دو سویه با انسان است، با ماشین ها، اگر سیستم درگیر فرآیند کنترل است یا به وسیله دیگر برنامه ها، اگر که سیستم برنامه های کاربردی را کنترل و منظم کند. وقتی سناریو کاربرد برنامه مشخص شد، مدل سازی نرم افزار آغاز می شود. بخش هایی که در زیر آمدم یک سری فنی را تعریف می کنند که ممکن است برای جمع آوری نیازمندی های اولیه مشتری استفاده شده و سپس مدل تحلیلی را برای سیستم شیء گرا تعریف می کند.

### ۲۱-۴-۱ موارد کاربرد (Use-Case ها)

همان گونه که در فصل ۱۱ توجه کردیم، موارد استفاده از نقطه نظر کاربر نهایی، مدل سازی می شود. موارد استفاده ای که در طول کشف و استخراج نیازمندی ها ایجاد شده اند، باید به اهداف زیر دست یابند:

- نیازمندی های عملیاتی و کارکردی سیستم (محصول) را با تعریف سناریو کاربرد که توسط کاربر نهایی و تیم مهندسی نرم افزار مورد موافقت قرار گرفته، تعریف می کند.
- توصیف مشخص و غیر مبهمی از چگونگی ارتباط متقابل سیستم و کاربر نهایی را با یکدیگر فراهم سازد.

- پایه و اساسی برای اعتبار آزمون اعتبار سنجی و تصدیق فراهم کند.

در طول تحلیل شیء گرا، موارد استفاده به عنوان پایه و اساس اولین عنصر مدل تحلیل باشد. با استفاده از روش UML، نمایش نموداری از یک مورد استفاده به نام دیاکرام یا نمودار مورد کاربرد<sup>۲</sup>، ایجاد می شود. مانند بسیاری از عناصر مدل تحلیلی، می توان نمودار مورد استفاده را در بسیاری از سطوح تجزید نمایش داد. بازیگران<sup>۳</sup> موجودی هایی هستند که با سیستم ارتباط متقابل دارند. آنها می توانند کاربران انسانی یا سیستم ها و ماشین های دیگری باشند که رابط با سیستم را تعریف کرده اند.



Use-Case ها ابزار

متمم برای تعیین

نیازمندی ها می باشند.

که مستقل از شیوه

تحلیل به کار می روند.

فصل ۱۱ را برای

اطلاعات بیشتر مرور

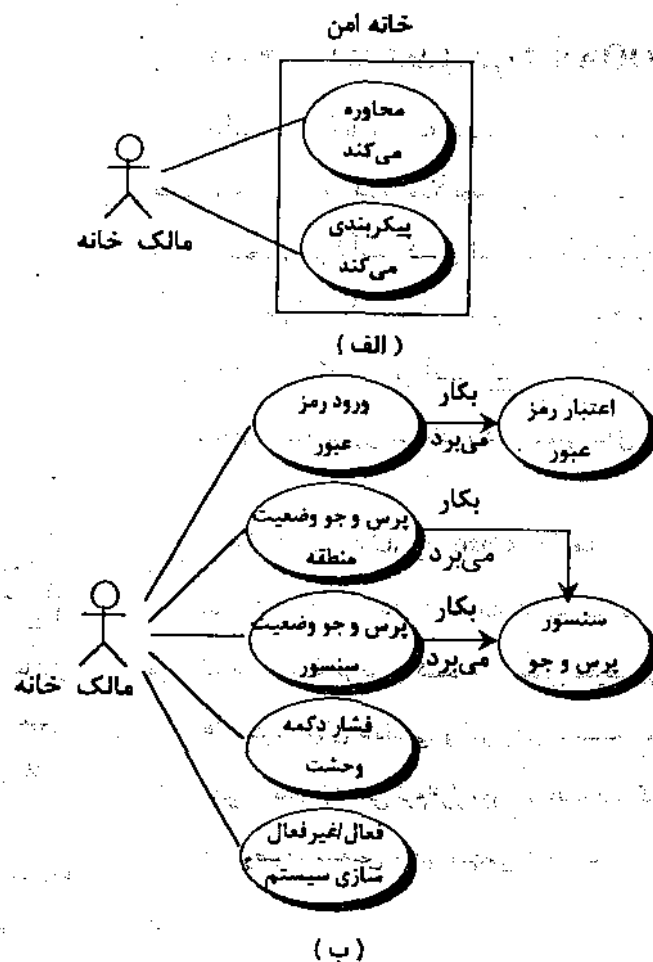
کنید.

1. De Champeaux, D

2. use-case diagram

3. actores

به منظور تشریح توسعه دیاگرام مورد استفاده، موارد کاربرد را برای سیستم ایمنی خانه امن که در بخش ۲-۱۱ آمده، در نظر می‌گیریم. سه بازیگر وجود دارند: مالک خانه، سنسورها و سیستم فرعی مشاهده و عکس‌العمل، به منظور هدفی که از این مثال دنبال می‌کنیم، تنها مالک خانه در نظر گرفته می‌شود.



شکل ۲-۲۱ (الف) نمودار Use Case سطح بالا

(ب) نمودار Use Case تفصیلی

شکل ۲-۲۱ الف نمودار سطح بالایی از مورد کاربرد را در مورد مالک خانه نشان می‌دهد. با رجوع به شکل ۲-۲۱ الف، مورد استفاده شناسایی شد. هر یک از این موارد سطح بالا ممکن است با نمودارهای سطح پایین تشریح شوند. به طور مثال، شکل ۲-۲۱ ب نمایانگر نموداری است که کارکرد (تلیع) رابطه متقابل یا محاوره را تشریح می‌کند.

مجموعه کاملی از نمودارهای مورد استفاده برای تمام بازیگران ایجاد می‌شود. بحث مفصلی از مدل‌سازی مدل کاربرد با استفاده از UML در کتابهای تخصصی یافته به این روش وجود دارد.



۲۱-۴-۲ مدل سازی کلاس - مسئولیت - مشارکت (CRC)<sup>۱</sup>

وقتی سناریوهای اولیه کاربرد برای سیستم ارائه شدند، زمان آن فرا می‌رسد که کلاس‌های کاندید شده شناسایی شده و مسئولیت‌ها و همکاری‌هایشان مشخص شود. مدل سازی CRC وسیله ساده‌ای برای شناسایی و سازمان‌دهی کلاس‌هایی است که با نیازهای سیستم یا محصول مرتبطند. امیلر مدل سازی CRC را به شکل زیر توصیف می‌کنند: [WIR90]<sup>۲</sup>

مدل CRC واقعاً مجموعه‌ای از کارت‌های شاخص استاندارد است که نمایانگر کلاس‌هاست. کارت‌ها به سه بخش تقسیم می‌شوند. در بالای کارت نام کلاس نوشته می‌شود. در متن کارت مسئولیت‌های کلاس را در سمت چپ و همکاری‌ها را در سمت راست می‌نویسند.

در واقع، مدل CRC ممکن است از کارت‌های ایندکس<sup>۳</sup> مجازی یا حقیقی استفاده کند. هدف توسعه یک نمایش سازمان‌دهی شده از کلاس‌هاست. مسئولیت‌ها، صفات خاص و عملیاتی هستند که مربوط به کلاس هستند. اگر بخواهیم به‌طور ساده‌تر بیان کنیم، مسئولیت، هر چیزی است که کلاس می‌داند یا انجام می‌دهد. همکاری‌ها<sup>۴</sup> کلاس‌هایی هستند که برای مهیا کردن اطلاعات لازم برای کلاسی که جهت تکمیل مسئولیت آن ضروری هستند. به‌طور کلی، همکاری نشانگر تقاضای اطلاعات یا یک عمل است.

## کلاس‌ها

رهنمودهای اولیه در مورد شناخت کلاس‌ها و اشیا در فصل ۲۰ نشان داده شدند. به‌طور خلاصه، اشیا خود را به شکل‌های مختلف جلوه‌گر می‌سازند (بخش ۲۰-۳-۱): موجودیت‌های خارجی، اجسام، وقایع یا حوادث، نقش‌ها، واحدهای سازمانی، مکان‌ها یا ساختارها. یک تکنیک شناسایی این موارد در بستر یک مسئله نرم‌افزاری عبارتست از انجام تجزیه دستوری (گرامری) روی روایت پردازشی سیستم. تمام اسم‌ها اشیا بالقوه هستند. هر شیء بالقوه باعث توقف نمی‌گردد. شش مشخصه انتخاب، تعریف شده‌اند:

۱- اطلاعات بازیابی شده. شیء بالقوه در طول تحلیل تنها در صورتی مفید خواهد بود که اطلاعات در مورد آن بازیابی می‌شوند به‌طوری‌که آن سیستم بتواند عمل کند.

۲- خدمات مورد نیاز. شیء بالقوه باید دارای مجموعه‌ای از عملیات قابل شناسایی باشد که می‌توانند ارزش صفات خاصه را به گونه‌ای تغییر دهند.



چگونه باید تحقیق

کنیم که یک شیء

بالقوه، شایستگی وارد

شدن در یک کارت

شاخص CRC را

دارد یا خیر؟

1. Class Responsibility - Collaborator (CRC)

2. Wirfs-Brock, R.B.

3. index cards

4. collaborators



۳- صفات خاصه چندگانه. در طول تحلیل نیازمندیها، نقطه توجه روی اطلاعات عمده است. در واقع، یک شیء با یک صفت خاصه ممکن است در طول طراحی مفید باشد، اما احتمالاً بهتر است که در طول کار تحلیل، به عنوان صفت خاصه شیء دیگری ارائه شود.

۴- مشخصه های معمول و مشترک. مجموعه ای از صفات خاصه را می توان برای شیء بالقوه ای تعریف کرد و این صفات خاصه در مورد تمام وقایع مربوط به شیء و به کار می روند.

۵- عملیات مشترک. مجموعه عملیاتی که می توان برای شیء بالقوه تعریف کرد و این عملیات در تمام وقایع مربوط به شیء به کار می روند.

۶- نیازمندی های ضروری. موجودیت های خارجی که در فضای مسئله ظاهر شده و اطلاعاتی را تولید یا مصرف می کنند که در مورد عملیات هر سیستمی لازمند زیرا سیستم تقریباً همیشه به عنوان اشیایی در مدل نیازمندیها، تعریف می شود.

یک شیء بالقوه باید تمامی شش مشخصه را دارا باشد اگر که قرار است برای گنجاندن در مدل CRC در نظر گرفته شود.

فایر اسمیت طبقه بندی انواع کلاس های فوق را با پیشنهاد ضمیمه زیر، بسط می دهد: [FIR93]<sup>۱</sup> کلاس بندی وسیله ها. موجودیت های برون مدل سنسورها، موتورها، صفحه کلیدها را مدل سازی کنید.

کلاس بندی خصوصیات. نمایان گر بعضی از ویژگیها و خصوصیات مهم محیط مسئله است (مثل نرخ اعتباری در بستر برنامه و ام رهنی).

کلاس بندی روابط متقابل. روابطی را که میان سایر اجسام رخ می دهد مدل سازی می کند. علاوه بر این، اشیاء و کلاس را می توان توسط مجموعه ای از مشخصه ها کلاس بندی کرد.

قابلیت لمس کردن. آیا این کلاس نمایان گر چیزهای ملموس است (مثل صفحه کلید یا سنسور) یا نمایان گر اطلاعات انتزاعی است (مثل نتیجه پیش بینی شده)؟

شمولیت. آیا این کلاس اتمیک<sup>۲</sup> است یعنی شامل کلاس های دیگری نمی شود یا جمعی است و شامل حداقل یک شیء می شود؟

تسلسل. آیا کلاس همروند است یعنی دارای مسیر کنترلی خود است یا تسلسلی است یعنی توسط منابع خارجی کنترل می شود؟



آیا راهی برای طبقه بندی کلاسها وجود دارد؟ چه ویژگیهایی ما را در این راه یاریگر است؟

1. Firesmith, D.G.

2. atomic

پایداری. آیا کلاس گذرا<sup>۱</sup> است؟ (یعنی ایجاد شده و در طول برنامه حذف می‌شود؟)، موقت<sup>۲</sup> (در طول عملیات برنامه ایجاد شده و وقتی برنامه خاتمه یافت از بین می‌رود؟) یا دائمی است؟ (در بانک اطلاعاتی ذخیره شده است).

جامعیت و یکپارچگی. آیا این کلاس قابل خراب شدن<sup>۳</sup> است؟ (یعنی در مقابل نفوذ خارجی، از منابع حفاظت نمی‌کند؟) آیا این که حفاظت شده<sup>۴</sup> است یعنی در مورد دسترسی به منابعش کنترل‌ها را به اجرا درمی‌آورد؟

با استفاده از کلاس‌بندی‌های فوق کارت ایندکسی که به‌عنوان بخشی از مدل CRC ایجاد شده ممکن است بسط یابد تا در برگزیده نوع کلاس و مشخصه‌های آن شود (شکل ۲۱-۳).

|  |            |
|--|------------|
| نام کلاس:  |            |
| نوع کلاس: (برای مثال دستگاه، خاصیت، نقش، رویداد) |            |
| ویژگیهای کلاس: (برای مثال ملموس، مجرد، همروند)   |            |
| مسئولیت‌ها:                                      | همکاری‌ها: |
|  |            |
|  |            |
|  |            |
|  |            |
|  |            |
|  |            |
|  |            |
|  |            |
|  |            |

شکل ۲۱-۳ یک کارت شاخص مدل CRC (کلاس - مسئولیت - همکاری)

- 1.transient
- 2.temporary
- 3.corruptible
- 4.guarded

## مسئولیت‌ها

راهنمایی‌های اولیه در مورد شناسایی مسئولیت‌ها (صفات خاصه و عملیات) در فصل ۲۰ نیز مورد بررسی قرار گرفتند. اگر بخواهیم به‌طور خلاصه بگوییم، مشخصات یا صفات نمایانگر خصوصیات پایدار یک کلاسند، یعنی اطلاعات در مورد کلاسی که باید حفظ شود تا به اهداف مشخص شده به نرم افزار توسط مشتری دست یابیم. گاهی می‌توان این صفات خاصه را از دامنه استخراج نمود یا با شناخت موجودیت کلاس به آنها رسید. عملیات را می‌توان با اجرای تجزیه دستوری روی پردازش روایت سیستم بدست آورد. تمام فعل‌ها به عملیات کاندید شده تبدیل می‌شوند. هر عملیاتی که برای یک کلاس انتخاب می‌شود رفتار آن کلاس را نشان می‌دهد.

ورفرز - براک و همکارانش پنج راهنمایی را برای تخصیص مسئولیت به کلاس ارائه می‌دهند:

[WIR90]

## ۱- هوش سیستم باید به‌طور یکنواخت توزیع شود.

هر برنامه کاربردی در برگیرنده درجه معینی از هوش است یعنی آنچه که می‌داند و آنچه که می‌تواند انجام دهد. می‌توان این هوش را در طول کلاس‌هایی به طرق مختلف توزیع کرد. کلاس‌های گنگ یا Dumb یعنی آنهایی که مسئولیت خیلی کمی دارند، را می‌توان مدل‌سازی کرد تا به عنوان خدمتگزاری برای کلاس‌های هوشمندتر عمل کنند (یعنی کلاس‌های دارای مسئولیت زیاد). با این که این روش جریان کنترلی را در سیستم ساده می‌کند اما دارای چند عیب است: (۱) این کار هوش را در چند کلاس خاص متمرکز می‌کند و تغییرات را مشکل‌تر می‌سازد. (۲) مایل است کلاس‌های بیشتری بدست آورد بنابراین کار توسعه بیشتر است.

بنابراین، هوش سیستم باید به‌طور یکنواخت در کلاس‌ها در یک برنامه، توزیع شود. از آن جا هر شیء در مورد کارهای محدودی اطلاعات داشته و کارهای محدودی را نیز انجام می‌دهد (که معمولاً به خوبی روی آنها متمرکز می‌شوند)، انجام سیستم بهبود می‌یابد. علاوه بر این، اثرات جانبی مربوط به تغییر کاهش می‌یابد زیرا هوش سیستم در بسیاری از اشیا از طریق آنها مرتبط شده است.

برای تعیین این که آیا هوش سیستم به‌طور یکنواخت توزیع شده یا نه، مسئولیت‌های مورد اشاره در هر کارت ایندکس مدل CRC باید ارزیابی شود تا معلوم شود آیا هر کلاس دارای فهرست اضافی از مسئولیت‌هاست یا خیر؟ این کار تمرکز هوش را نشان می‌دهد. علاوه بر آن، مسئولیت‌های هر کلاس باید سطح انتزاعی یکسانی را نشان دهند. مثلاً در میان عملیات فهرست‌بندی شده برای کلاس تجمیعی به نام کنترل حساب جاری، یک بازبینی‌کننده دو مسئولیت دارد: توازن حساب و کنارگذاری چک‌های تسویه شده.



مسئولیت‌های یک

کلاس مشتمل بر

صفات خاصه و عملیات

می‌باشد.



چگونه می‌توانیم

مسئولیت را به یک

کلاس تخصیص دهیم؟



اگر یک کلاس دارای

لیست فوق‌العاده بلند

بالایی از مسئولیت‌ها

بود، تعریف آن را به

گونه ای بشکنید که در

بیش از یک کلاس قرار

گیرد.

اولین عملیات (مسئولیت) نشان گر رویه منطقی و ریاضی پیچیده‌ای است. دومین عملیات یک کار دفتری ساده است. از آنجا که این عملیات در یک سطح انتزاعی نیستند، کنارگذاری چک‌های تشویه حساب شده باید جزو مسئولیت‌های مدخل چک قرار گیرد که یک کلاسی است که جزو کلاس گروهی کنترل حساب جاری است.

۲- هر مسئولیتی باید تا جایی که ممکن است به‌طور کلی بیان شود.

این رهنمود نشان گر این است که مسئولیت‌های کلی (صفات خاصه و عملیات) باید در سلسله مراتب بالا قرار گیرند (از آنجا که عمومی هستند در مورد کلاس‌های فرعی به کار می‌روند). علاوه بر این چند ریختی باید به منظور تعریف عملیاتی به کار رود که به‌طور کلی در مورد کلاس فرعی به کار رفته اما به‌طور مختلف در هر کلاس به اجزا درمی‌آیند.

۳- اطلاعات و رفتار مربوطه باید در همان کلاس قرار گیرند.

این کار به همان اصل شیء گرایی می‌رسد که آن را بسته بندی می‌نامیم (فصل ۲۰). داده‌ها و پردازش‌های مربوط به تغییر این داده‌ها باید به عنوان یک واحد یکپارچه در یک بسته قرار گیرند.

۴- اطلاعات مربوط به هر چیز باید در یک کلاس قرار گیرد. آنکه در کلاسهای چندگانه توزیع شود. هر کلاس مسئولیت ذخیره و تغییر نوع خاصی از اطلاعات را به عهده دارد. این مسئولیت به‌طور کلی بین چند کلاس تقسیم نمی‌شود. اگر اطلاعاتی توزیع شد، نگهداری نرم افزار سخت تر شده و آزمون آن تلاش بیشتری را می‌طلبد.

۵- مسئولیت‌ها باید به صورت مناسب میان کلاس‌های مربوطه تقسیم شوند. موارد متعددی وجود دارد که در آن یک سری اشیای مربوطه باید همه در یک زمان رفتار یکسانی را نشان دهند. به عنوان مثال بازی ویدیویی را در نظر بگیرید که باید اشیای زیر را نشان دهد: بازیکن - بدن بازیکن - دستهای بازیکن - پاهای بازیکن - سر بازیکن، هر یک از این اشیای دارای مشخصه‌های خود است (مثل: موقعیت، جهت گیری، رنگ، سرعت) و همه باید به روز شده و با استفاده کاربر از دسته بازی به نمایش<sup>۱</sup> درآیند. مسئولیت به روزسازی و نمایش باید توسط هر یک از اشیای ذکر شده، تقسیم شوند. بازیکن می‌داند که چیزی تغییر کرده و به روزسازی لازم است. این قسمت با سایر اشیای همکاری می‌کند تا موقعیت یا جهت گیری جدیدی را انجام دهد، اما هر شیء نمایش خود را کنترل می‌کند.

1. encapsulation

2. update

3. display

## مشارکت (همکاری)



کلاس‌ها به یکی از دو روش زیر مسئولیتشان را به انجام می‌رسانند: (۱) کلاس می‌تواند عملیات خود برای تغییر مشخصه‌هایش استفاده کند، بدین‌وسيله مسئولیت خاصی را به انجام می‌رساند یا (۲) کلاس می‌تواند با سایر کلاس‌ها همکاری کند.

یک شیء خادم

(سرویس دهنده) با

یک شیء مخدوم

(سرویس گیرنده) در

انجام برخی مسئولیتها

همکاری دارند

همکاری آنها مشتمل

بر رد و بدل کردن

پیامهاست.

ورفر - براک و همکارانش مشارکت‌ها را به‌طریق زیر تعریف کرده‌اند:

همکاری‌ها نمایان‌گر تقاضا از جانب مشتری به یک سرویس‌دهنده برای تکمیل مسئولیت مشتری است. همکاری عبارتست از ایجاد قرارداد بین مخدوم و خادم ... ما می‌گوییم در صورتی یک شیء با شیء دیگر همکاری می‌کند تا مسئولیتی را به انجام برساند باید پیام‌هایی را به شیء دیگر برساند. یک همکاری در یک جهت جریان دارد که نمایان‌گر تقاضا از جانب مخدوم به خادم است. از دیدگاه مشتری، هر یک از این همکاری‌ها با مسئولیت به‌خصوص به اجرا درآمده توسط خادم، مرتبط هستند.

این همکاری‌ها روابط بین کلاس‌ها را شناسایی می‌کنند. وقتی مجموعه‌ای از کلاس‌ها همگی با هم کار می‌کنند تا برخی نیازمندیها را به انجام برسانند، می‌توان آنها را به‌صورت یک سیستم فرعی سازمان‌دهی کرد (یک موضوع طراحی).

همکاری‌ها با تعیین این مطلب شناسایی می‌شوند که آیا یک کلاس می‌تواند مسئولیتی را به‌خودی‌خود انجام دهد یا نه؟ اگر نمی‌تواند، پس لازم است با کلاس دیگر ارتباط داشته باشد و این یک همکاری است.

به‌طور مثال، برنامه کاربردی خانه امن را در نظر بگیرید.<sup>۱</sup> به‌عنوان بخشی از شیوه فعال‌سازی<sup>۲</sup>، شیء صفحه‌کنترل، باید تعیین کند که آیا سنسورها باز هستند یا خیر؟ یک مسئولیت به نام تعیین وضعیت سنسور تعریف شده است. اگر سنسورها باز باشند، صفحه کنترل باید وضعیتی را تنظیم کند که حالت «غیر آماده یا Not Ready» را نشان می‌دهد. اطلاعات سنسور را می‌توان از شیء سنسور بدست آورد. بنابراین، مسئولیت تعیین وضعیت سنسور تنها در صورتی انجام می‌گیرد که صفحه‌کنترل با سنسور همکاری کند.

به‌منظور کمک به شناسایی همکارها، تحلیل‌گر می‌تواند سه رابطه کلی مختلف را بین کلاس‌ها، بررسی کند:

بخشی یا قسمتی از ...

(۱) ارتباط is-part-of

اطلاعی یا دانشی از ...

(۲) رابطه has-knowledge-of

مبتنی یا وابسته به ...

(۳) رابطه depends-upon

۱. قسمت ۲۰-۲ را برای توصیف کلاسهای خانه امن، مرور کنید.

با کشیدن نمودار رابطه کلاس (بخش ۲۱-۴-۴)، تحلیل گر روابط لازم برای شناسایی این ارتباطات را به وجود می آورد. هر یک از سه رابطه کلی به طور خلاصه در پاراگرافهای زیر مورد بررسی قرار گرفته اند. تمام کلاس هایی که بخشی از یک کلاس جمعی را تشکیل می دهند از طریق رابطه Is-Part-Of با کلاس کل مرتبطند. کلاس های تعریف شده برای بازی ویدیویی را که پیش تر بیان شده بود، در نظر بگیرید، کلاس بدن بازیکن بخشی از بازیکن است همچون، سِر، پاها و دستهای بازیکن.

وقتی یک کلاس باید اطلاعاتی را از کلاس دیگر کسب کند، رابطه has-knowledge-of به وجود می آید. مسئولیت «تعیین وضعیت سنسور» که پیش تر مورد اشاره قرار گرفت، نمونه ای از این رابطه است.

رابطه depends-upon نشان دهنده این است که دو کلاس دارای وابستگی هستند که به وسیله دو رابطه دیگر یعنی is-part-of و has-knowledge-of به وجود نیامده است. مثلاً سِر بازیگر باید همیشه به بدن متصل باشد، با این حال هر شیء می تواند بدون شناخت مستقیم دیگری وجود داشته باشد. یکی از مشخصه های شیء «سِر بازیگر» به نام موقعیت مرکزی از روی موقعیت مرکزی بدن بازیکن، بستگی دارد.

در تمام موارد، نام کلاس همکاری روی کارت ایندکس مدل CRC در کنار مسئولیتی که باعث ایجاد همکاری شده، نوشته و ثبت می شود. بنابراین، کارت ایندکس حاوی فهرستی از مسئولیت ها و همکاری های



مربوطه بوده که انجام آنها را ممکن و مقدور می سازد. (شکل ۲۱-۲)

موقتی مدل کامل CRC ارائه شده، نمایندگان از سازمان های مهندسی نرم افزاری و مشتری می توانند با استفاده از روش زیر از مدل بازیابی کنند:

یک رهیافت موثر برای بازیابی یک مدل است CRC کدام است؟

۱- به تمام شرکت کنندگان در این بازیابی زیر مجموعه ای از کارت های ایندکس مدل CRC

معمولاً به صورت یک بسته داده می شود، کارت هایی که باعث ایجاد همکاری می شوند باید جدا گردند.

۲- تمام سناریوهای مورد کاربرد (و نمودارهای مربوطه) باید در طبقه بندی های سازمان دهی

شوند.

۳- مسئول بازیابی آگاهانه و تعیناً مورد کاربرد را می خواند. وقتی به شیء دارای نامی رسید،

آن را به شخصی که کارت ایندکس کلاس مربوطه را دارد می سپرد. مثلاً، یک مورد استفاده در مورد

خانه امن دارای شکل روایی زیر است:

مالک خانه صفحه کلید کنترلی خانه امن را مشاهده می کند تا بفهمد آیا سیستم برای ورودی آماده

است یا نه؟ اگر سیستم آماده نیست، مالک باید به طور فیزیکی پنجره درها را ببندد به طوری که نشانگر

آماده، حاضر شود. [نشانگر غیر آماده نشانگر این است که سنسوری باز است یعنی در یا پنجره ای باز است]

وقتی مسئول بازیابی به کنترل پائل می رسد نشانه به شخصی می رسد که کارت ایندکس کنترل پائل

را دارا می باشد. عبارت «نشان می کند که سنسور باز است» نیازمند این است که کارت ایندکس دارای

مسئولیتی باشد که این نشانه را ارزیابی می کند. (مسئولیت «تعیین وضعیت سنسور» به این امر دست

می یابد. در کنار قسمت مسئولیت‌ها روی کارت ایندکس، سنسور همکار است. این نشانه به شیء سنسور می‌رسد.

۴- وقتی نشانه رد می‌گردد، از دارنده کارت کلاس یا کلاس خواسته می‌شود مسئولیت‌های مورد اشاره بر روی کارت را تعریف کند. گروه تصمیم‌گیری می‌کند که آیا یک (یا چند) مسئولیت نیازمندی مورد کاربرد را رفع می‌کند یا خیر؟

۵- اگر مسئولیت‌ها و همکاری‌های مورد اشاره فوق بر روی کارتهای ایندکس با مورد کاربرد سازگاری نداشته، تغییراتی روی کارتهای صورت می‌گیرد. این کار ممکن است شامل تعریف کلاس‌های جدید (و کارتهای ایندکس ERC مربوطه) با مشخص کردن مسئولیت‌ها یا همکاری‌های جدید یا اصلاح شده روی کارتهای موجود است.

این روش کاری ادامه می‌یابد تا وقتی که مورد استفاده تمام می‌شود. وقتی همه موارد کاربرد (یا نمودارهای آن) مورد بازبینی قرار گرفت، OOA ادامه می‌یابد.

### ۲۱-۴-۳ تعریف ساختارها و سلسله مراتب

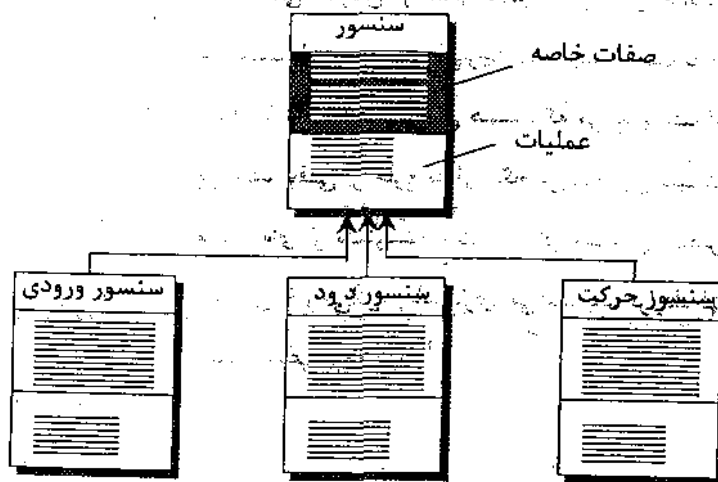
وقتی کلاس‌ها و اشیا با استفاده از مدل CRC شناسایی شدند، تحلیل‌گر شروع به تمرکز روی ساختار مدل کلاس و سلسله مراتب منتج از آن می‌کند که با ظهور کلاس‌ها و زیر کلاس‌ها به وجود می‌آیند. با استفاده از روش UML، یک سری نمودارهای کلاس را می‌توان ایجاد کرد. ساختار کلاس‌های تعمیم یافته<sup>۱</sup> را می‌توان برای شناسایی کلاس‌ها، به وجود آورد.

برای تشریح این موضوع، شیء سنسور را که برای خنک‌کنش تعریف و در شکل ۲۱-۴ آمده، در نظر بگیرید. در این جا، کلاس تعمیم یافته یعنی سنسور به صورت مجموعه‌ای از موارد تخصیصی در می‌آید. یعنی سنسور ورودی، سنسور دود و سنسور حرکتی. صفات خاصه و عملیات اشاره شده برای کلاس سنسور از طریق تخصیص‌سازی کلاس، به ارث می‌رسند. ما یک سلسله مراتب ساده در کلاس ایجاد کرده‌ایم.

در سایر موارد، یک شیء که در مدل اولیه نشان داده شده، ممکن است به واقع متشکل از چند جزء باشد که خودشان می‌توانند به نوبه خود به عنوان اشیا تعریف شوند. این اشیا گروهی را می‌توان به عنوان گروه یا کلاس مرکب<sup>۱</sup> نشان داد و با استفاده از علامت نشان داده در شکل ۲۱-۵، تعریف می‌شوند. شکل ۲۱-۵ لوزی نشانگر رابطه ساخت و مونتاژ است. باید توجه داشت که اتصال خطوط ممکن است با علائم اضافی (که نشان داده نشده‌اند) افزایش یابد تا نمایانگر اهمیت باشد. این موارد از روی علائم مدل‌سازی رابطه موجودیت (Entity-Relationship) که در فصل ۱۲ مورد بحث قرار گرفته، برگرفته شده‌اند.

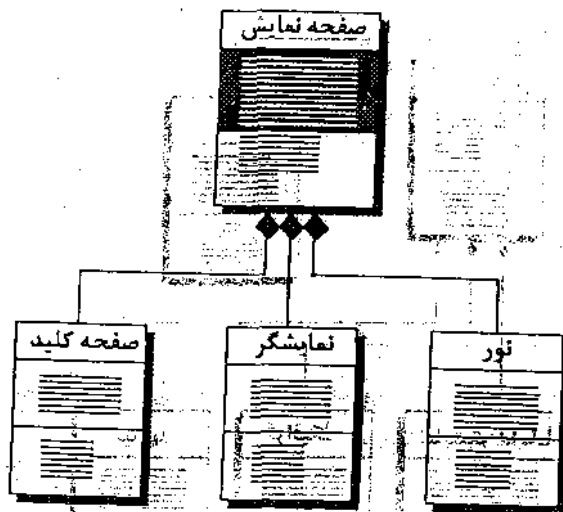
۱. generalization/specialization





شکل ۴-۲۱ نمودار کلاس برای تعمیمی / تخصصی

بازنمایی ساختاری وسیله‌ای برای تعمیم‌دهی مدل CRC برای تحلیل‌گر فراهم آورده و این تقسیم‌بندی را به صورت گرافیکی نشان می‌دهد. به هر کلاس جزئیات لازم را برای بازیابی و طراحی مربوطه فراهم می‌کند.



شکل ۵-۲۱ نمودار کلاس برای اجماع ترکیبی

#### ۴-۴-۲۱ تعریف موضوعات و زیر سیستمها

ممکن است مدل تحلیلی در مورد یک برنامه پیچیده دارای صدها کلاس و دهها ساختار باشد. به همین دلیل، لازم است شکل دقیقی را مشخص کرد که چگونه و عناصر مدل‌های ساختاری و CRC توصیف شده فوق باشند.



یک سیستم (بسته)

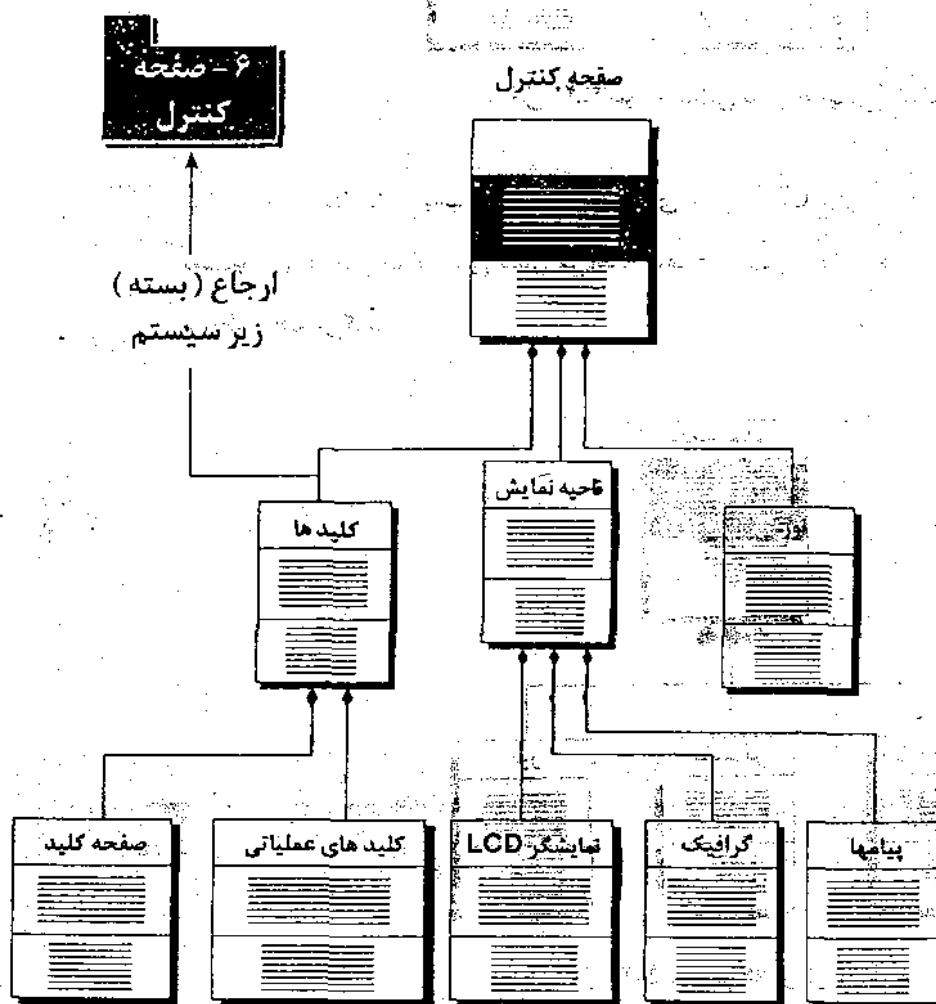
UML به یک

سلسله مراتب کلاس

با جزئیات بیشتر، اشاره

دارد

وقتی گروهی از تمام کلاس‌ها در میان خودشان به همکاری می‌پردازند تا به مجموعه‌ای از مسئولیت‌های منسجم و یکپارچه دست یابند، اغلب به آنها سیستم فرعی یا (بسته) پکیج<sup>۱</sup> می‌گویند. این بسته‌ها، خلاصه‌بندی‌هایی هستند که مرجع یا نشانگری به منظور جزئیات بیشتر در مدل تحلیل مهیا می‌کنند. وقتی از خارج به آن نگاه می‌کنید این سیستم فرعی می‌تواند یک جعبه سیاه باشد که دارای مجموعه‌ای از مسئولیت‌هاست و دارای همکاران خودش است. سیستم فرعی یک یا چند قرارداد را با همکاران خارجی‌اش، پیاده‌سازی می‌کند. یک قرارداد فهرست ویژه‌ای از تقاضاهایی است که همکاران از سیستم فرعی توقع دارند.<sup>۲</sup>



شکل ۶-۲۱ ارجاع به بسته (زیر سیستم)

۱. subsystem or package

۲. به خاطر داشته باشید که کلاسها با یک فلسفه خادم / مخدوم مجاوره می نمایند. در این مورد سیستم فرعی خادم و همکاران خارجی، مخدوم‌ها می باشند.

سیستم‌های فرعی را می‌توان با بافت مدل‌سازی CRC به وسیله ایجاد کارت ایندکس سیستم فرعی نشان داد. این کارت نشانگر نام سیستم‌های فرعی، قراردادی که سیستم منعقد می‌کنند و کلاس‌ها یا سایر سیستم‌های فرعی است که این قرارداد را پشتیبانی می‌کنند.

بسته‌ها از نظر هدف و محتوا مشابه سیستم‌های فرعی هستند اما به صورت گرافیکی در UML نمایش داده می‌شوند. مثلاً، فرض کنید که صفحه کنترل خانه امن تا حد قابل توجهی پیچیده‌تر از آن چیزی است که در شکل ۲۱-۵ آمده و دارای نقاط نمایش چندگانه، آرایش پیچیده کلیدها و دیگر مشخصه‌هاست. ممکن است آن را به عنوان یک ساختار جمعی مرکب همان‌طور که در شکل ۲۱-۶ آمده، نشان داد. اگر مدل کلی نیازمند به حاوی ده‌ها ساختار از این قبیل باشد، بدست آوردن کل این شکل به یک باره سخت است. با تعریف یک مرجع بسته همان‌طور که در شکل نشان داده شده، ساختار کلی را می‌توان به وسیله یک پیکان (شعاع) مورد ارجاع قرار داد. (فولدر یا پوشه فایل). مرجع‌های بسته (پکیج) را می‌توان برای هر ساختاری که دارای اشای چندگانه است استفاده نمود.

در امتزاعی ترین و تجزیه‌ی ترین سطح، مدل OOA تنها حاوی مراجع بسته است مثل همان‌هایی که در شکل ۲۱-۷ در بالا آمده است. هر یک از مراجع‌ها به صورت ساختارهایی توسعه می‌یابند. ساختارهای انبساطی صفحه کنترل و سنسور در شکل زیر نشان داده شده‌اند. ساختارهای سیستم، رویداد سنسور و زنگ خطر صوتی نیز ایجاد می‌گردند.

پیکان خط چین نشان داده شده در قسمت بالای شکل ۲۱-۷ نمایانگر رولپ و بستگی میان بسته‌ها نشان داده شده‌اند. برای مثال سنسور به بسته یا پکیج Sensor Event (وضعیت سنسور) بستگی دارد. پیکان‌های سیاه رنگ نمایانگر ترکیب هستند. در مثال نشان داده شده، بسته سیستم از بسته‌های صفحه کنترل، سنسور و زنگ خطر صوتی تشکیل شده است.

### ۲۱-۵ مدل رابطه شیء

بافت مدل‌سازی CRC اولین عناصر کلاس و رولپ شیء را بنیان می‌نهد. اولین مرحله در ایجاد روابط عبارتست از درک مسئولیت‌های هر کلاس، کارت مدل CRC حاوی فهرستی از مسئولیت‌هاست. مرحله بعد تعریف کلاس‌های همکاری است که در دست‌یابی به هر مسئولیت به ما کمک می‌کنند. این کار «ارتباط» بین کلاس‌ها را به وجود می‌آورد.

بین دو کلاس که به هم مرتبط هستند، یک رابطه وجود دارد.<sup>۱</sup> بنابراین، همکاران همیشه به شکلی با هم مرتبطند. رایج‌ترین نوع ارتباط دو دویی است یعنی رابط بین دو کلاس وجود دارد. وقتی رابطه دو دویی

۱. اصطلاحات دیگری که برای رابطه استفاده می‌شود، شرکت پذیری [RUM91] و اتصال [COA91] می‌باشد.

در بافت سیستم شیء گرا در نظر گرفته می شود. دارای جهت بخصوصی است که بر اساس آن تعریف شده

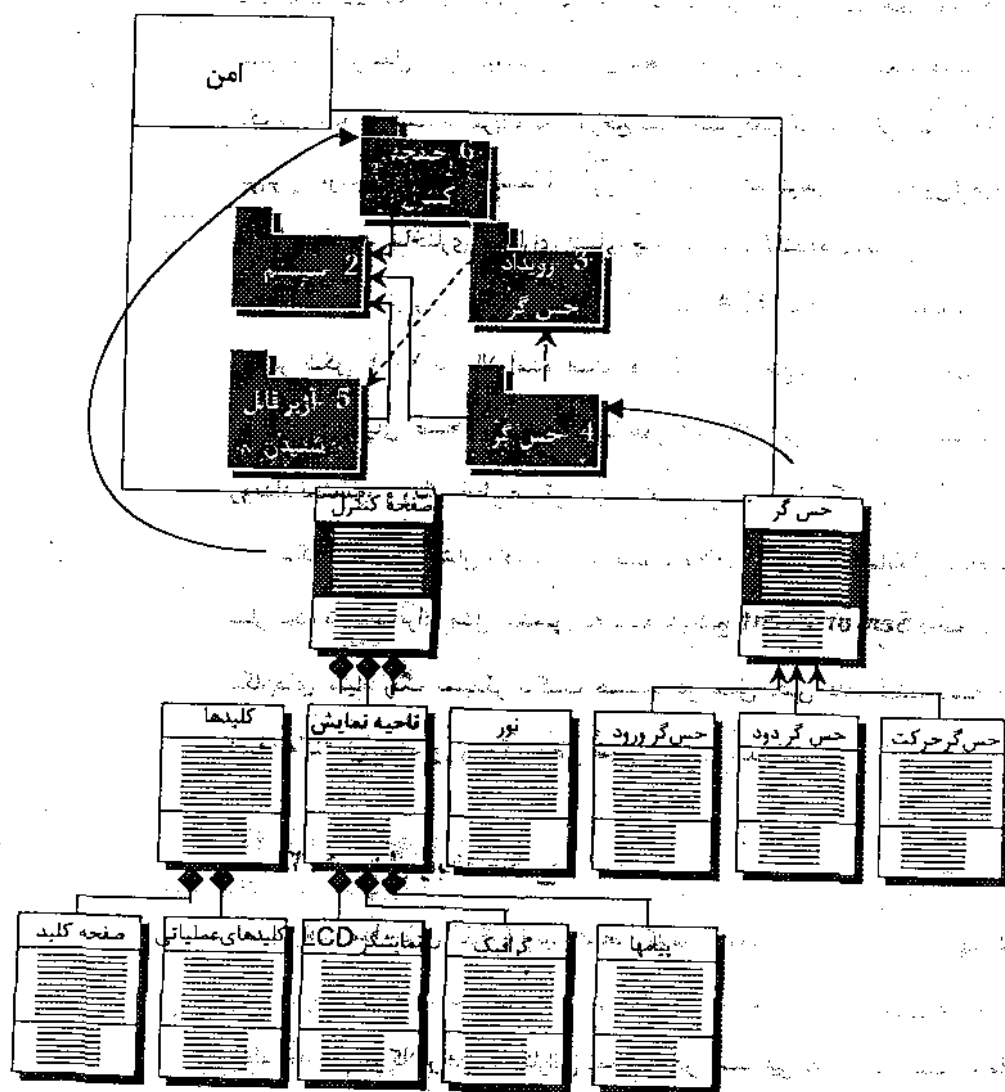
و در آن نقش کلاس به عنوان مختم یا خادم تعیین می گردد.

رامباخ و همکارانش [RUM91]<sup>۲</sup> روابطی را بیان می دارند که می توانند با بررسی افعال حالت یا

عبارات فعلی (منسوب به فعل، کنونی منظور نمی باشد) جمله دلفه یا موارد کاربرد سیستم، بدست

آیند. با استفاده از تجزیه گرافری، تحلیل گز افعالی را مجزا می سازد که نشان گر محل یا جای گیری فیزیکی،

نسب ارتباطات، مالکیت و برقراری شرطی هستند. این موارد نشانه رابطه را مهیا می کنند.



شکل ۲۱-۷ یک مدل تحلیلی با ارجاعات به بسته ها

۱. توجه به این نکته مهم است که اینیک حرکت از طبیعت دو سویه (دو جهته) رابطه استفاده شده در مدل سازی داده می

باشد. (فصل ۱۲)

2. Rumbaugh, J.

روش یکپارچه زبان مدل سازی در مورد مدل رابطه-شیء از نشانه گذاری استفاده می کند که از فنون مدل سازی رابطه- موجودیت که در فصل ۱۲ مورد بحث قرار گرفت، برگرفته شده اند. در اصل، اشیاء با استفاده از روابط نام گذاری شده به اشیاء دیگر مرتبط می شوند. اهمیت این ارتباط معین شده و شبکه کلی ارتباطات به وجود آمده است.

مدل رابطه-شیء (مثل مدل رابطه- موجودیت) می تواند دارای سه مرحله باشد:

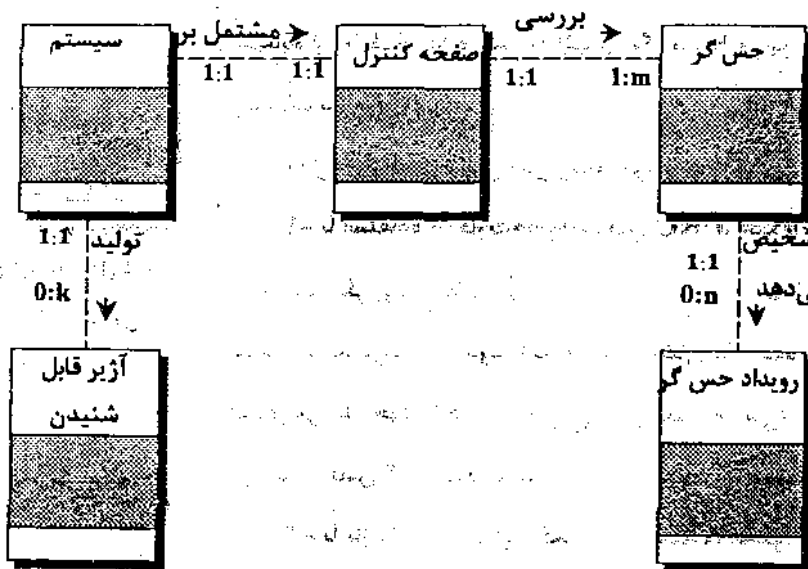
۱- با استفاده از کارت های ایندکس CRC، شبکه ای از اشیاء همکار ترسیم می شود. شکل ۸-۲۱ نمایانگر روابط کلاس در مورد اشیاء خانه امن است. ابتدا اشیاء ترسیم شده توسط خطوط بدون عنوانی بهم مرتبط می شوند (که در شکل نشان داده نشده) که نشانگر این هستند که رابطه ای میان اشیاء مرتبط وجود دارد. روش مشابهی برای نقش هر شیء که باید به جای رابطه تعریف شود وجود دارد. این امر در فصل ۱۲ توصیف شده است.

۲- با بازبینی کارت ایندکس مدل CRC، مسئولیت ها و همکاران ارزیابی شده و هر خط ارتباطی بدون عنوانی نام گذاری می شود. به منظور پیشگیری از ابهام، سر پیکان نمایانگر جهت رابطه است (شکل ۸-۲۱).

۳- وقتی روابط نام گذاری شده ایجاد گردیدند هر پایان مورد ارزیابی قرار می گیرد تا کاردینالیت آن تعیین شود. چهار گزینه وجود دارد: صفر به یک، یک به صفر، صفر به بسیار، یک به بسیار. مثلاً، سیستم خانه امن حاوی یک کنترل پائل است. (کاردینالیت ۱:۱ نشانگر این است)، حداقل یک سنسور باید برای انتخاب پوشش صفحه کنترل وجود داشته باشد. ممکن است سنسورهای متعددی موجود باشند (عبارت ۱: m نشانگر این امر است). یک سنسور می تواند میان ۱ رویداد سنسور یا بیشتر تمایز قائل شود. (مثلاً دود شناخته شده یا خرابی رخ داده).



چگونه یک مدل شیء  
رابطه بدست می  
آید؟



شکل ۲۱-۸ رابطه های بین اشیاء

مراحل مورد اشاره فوق تا وقتی ادامه می یابند که مدل کامل رابطه شیء تولید شده باشد. با گسترش

مدل رابطه- شیء تحلیل گر، بعد دیگری به مدل کلی تحلیل اضافه می کند. نه تنها روابط میان اشیاء

شناسایی شده اند، بلکه مسیرهای مهم پیام نیز تعریف گردیده اند (فصل ۲۰). در بحث ما از شکل ۲۱-۷ هر

پیکان نشانگر تبادل میان پیامها در سیستمهای فرعی مدل است.

## ۲۱-۶ مدل رفتار- شیء

مدل CRC و مدل رابطه- شیء، نمایانگر عناصر ایستای مدل تحلیل شیء گرا هستند. اکنون وقت

آن رسیده که به سوی رفتار پویا در سیستم شیء گرا یا محصول برویم. به این منظور باید رفتار سیستم را

به عنوان کارکرد رویدادهای خاص و زمان خاص به نمایش درآوریم.

مدل رفتار شیء نشانگر این است که چگونه یک سیستم شیء گرا به حوادث یا محرکهای بیرونی

واکنش می دهد. برای ایجاد مدل تحلیل گر باید مراحل زیر را انجام دهد:

۱- تمام مولد کاربرد را ارزیابی کند تا به طور کامل توالی روابط میان آنها را در سیستم

بشناسد.

۲- حوادثی را شناسایی کند که باعث توالی عمل شده و چگونگی ارتباط آنها را با اشیای

خاص درک کند.

۳- مسیر حادثه های را برای هر مورد کاربرد ایجاد کند.

۴- یک نمودار تغییر وضعیت برای سیستم ترسیم کند.



کدام گامها برای یک

مدل شیء - رفتار مورد

نیاز است؟



۵- مدل رفتار شیء را به منظور تعیین دقت و انسجام بازیابی کند.

هر یک از مراحل فوق در بخش‌های بعدی مورد بحث قرار می‌گیرند.

#### ۲۱-۶-۱ شناسایی رویداد با موارد کاربرد (Use-Case ها)

همان‌گونه که در بخش ۲۱-۴-۱ توجه کردیم، مورد کاربرد نمایانگر یک سری فعالیت‌هایی است که بازیگران سیستم و خود سیستم را در برمی‌گیرد. به‌طور کلی، رویداد وقتی رخ می‌دهد که سیستم و بازیگر نقش در آن (که این بازیگر می‌تواند یک انسان، یک وسیله یا حتی یک سیستم خارجی باشد) اطلاعاتی را رد و بدل می‌کنند. با یادآوری بحث فصل ۱۲ نکته مهم قابل توجه این است که یک رویداد یا حادثه دو ارزشی یا بولین است، یعنی حادثه اطلاعاتی نیست که تبادل شده‌اند بلکه در واقع، خود حقیقت و واقعیت امر، تبادل این اطلاعات است.

یک مورد کاربرد از نظر نکات اطلاعاتی تبادل شده بررسی گردید. برای تشریح این مسئله، مورد کاربرد مربوط به خانه امن را که در فصل ۱۱-۲-۴ توصیف شده بود، یکبار دیگر در نظر بگیرید:

۱- مالک خانه صفحه کلید کنترل خانه امن را مشاهده می‌کند (شکل ۱۱-۲) تا ببیند آیا سیستم برای ورود آماده است ای خیر؟ اگر سیستم آماده نبود، مالک باید به‌طور فیزیکی پنجره‌ها و درها را ببندد به‌طوری‌که علامت آماده بودن روشن شود.

۲- مالک خانه از صفحه کلید برای وارد کردن رمز عبور ۴ رقمی استفاده می‌کند. کلمه رمز با کلمه رمز معتبر ذخیره شده در سیستم مقایسه می‌شود. اگر درست نبود، صفحه کنترل صدای بوق را پخش کرده و خود را برای ورود مجدد کلمه رمز آماده می‌کند. اگر کلمه رمز صحیح بود، صفحه کنترل منتظر اقدامات بعدی می‌شود.

۳- مالک خانه کلید های stay یا away (بودن در خانه یا رفتن از خانه) را انتخاب و وارد می‌کند تا سیستم فعال گردد. کلید Stay تنها سنسورهای محیطی را فعال می‌سازد و کلید Away همه سنسورها را فعال می‌کند.

۴- وقتی عمل فعال‌سازی رخ داد، یک چراغ هشداردهنده قرمز رویت می‌شود.

قسمت‌هایی که زیرشان خط کشیده شده نشان‌دهنده رویدادها هستند. بازیگر یا عمل‌کننده در سیستم برای هر حادثه شناسایی می‌شود. اطلاعاتی که تبادل یافته مورد توجه قرار می‌گیرند و شرایط و محدودیت‌هایی در فهرست قرار می‌گیرند.

به‌عنوان مثالی از یک واقعه خاص، عبارت خط کشیده اول را در جمله ۱ مشاهده کنید. در متن تحلیل شیء گرا، شیء Home Owner یا صاحبخانه یک واقعه را به شیء صفحه کنترل انتقال می‌دهد. این واقعه ممکن است ورود کلمه رمز باشد. اطلاعات انتقال یافته یک کلمه رمز چهار رقمی است که کلمه عبور را می‌سازند اما این بخش ضروری مدل رفتاری نیست. نکته مهم مورد توجه این است که



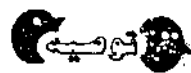
بعضی از این وقایع دارای یک تأثیر مشخص روی جریان کنترل مورد کاربردی هستند در حالی که بقیه تأثیر مستقیمی روی آن ندارند.

مثلاً کلمه رمز وارد می شود اما این واقعه به طور مشخص جریان کنترل را تحت تأثیر قرار نمی دهد اما نتایج حاصل از کار مقایسه کلمه رمز تأثیر مشخصی روی اطلاعات و جریان کنترلی نرم افزار خانه امن وقتی همه وقایع شناسایی شدند، آنها را به اشیای مربوطه اختصاص می دهند. اشیای می توانند مسئول تولید حوادث بوده (مثلاً صاحبخانه عمل ورود کلمه رمز را انجام می دهد) یا حوادثی را شناسایی کنند که در جای دیگری رخ داده (مثلاً صفحه کنترل نتیجه دوگانه عمل مقایسه اسم رمز است).

### ۲۱-۶-۲ بازنمایی وضعیت

در بافت و متن سیستم شیء گرا دو مشخصه متفاوت از حالات و وضعیت ها وجود دارد که باید بررسی شوند: (۱) وضعیت هر شیء که سیستم عمل آن را انجام می دهد و (۲) وضعیت سیستم از بیرون، هنگامی که سیستم عملکرد خود را انجام می دهد.

وضعیت شیء دارای مشخصه های فعال و غیرفعال است. وضعیت غیرفعال صرفاً وضعیت جاری همه صفات خاصه اشیاء می باشد. مثلاً وضعیت غیرفعال شیء ترکیبی **Player** یا بازیکن (در برنامه بازی ویدیویی که پیش تر گفته شد) شامل موقعیت مکانی و جهات مختلف بازیکن همراه با سایر صفات خاصه ای است که به بازی مربوطند: [CHA93]. وضعیت فعال نشان دهنده وضعیت جاری شیء است وقتی که شیء تحت تغییر یا پردازش مستمر قرار می گیرد، شیء بازیکن ممکن است دارای حالات فعال زیر باشد: حرکت، استراحت، مجروح شدن، مداوا کردن، در تله افتاده، گم شده و غیره. یک واقعه باید رخ دهد تا شیء وادار به انتقال از یک حالت فعال<sup>۲</sup> به حالتی دیگر شود. یک جزء از رفتار شیء تنها نمایش حالات فعال شیء برای هر شیء و هر واقعه ای است که باعث تغییراتی میان این حالات فعال می شود. شکل ۲۱-۹ نمایش ساده ای از حالت فعال برای شیء صفحه کنترل در سیستم خانه امن است.



هنگامی که تعریف

وضعیت ها را آغاز می

کنید، بر وضعیت های

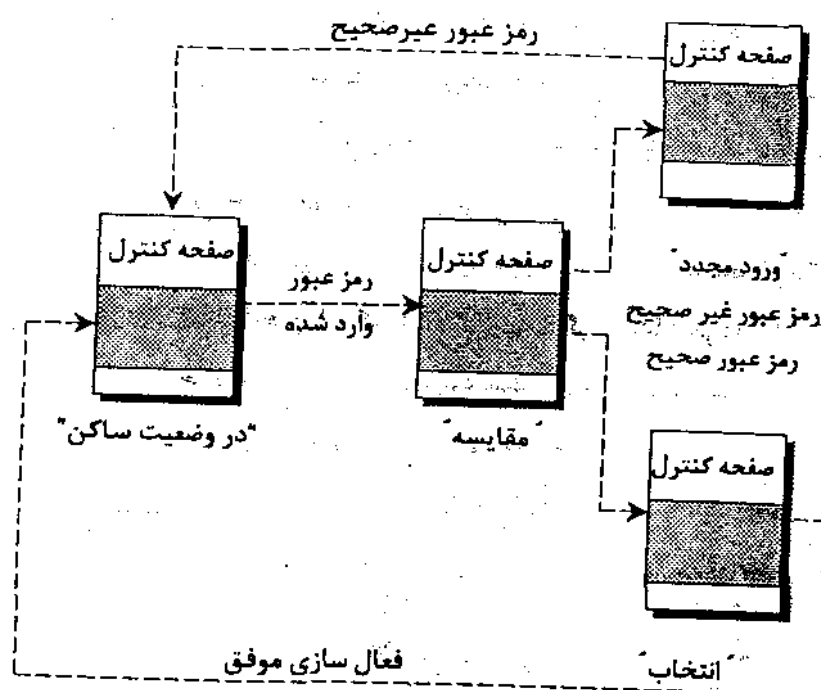
مختلف رفتارهای

خارجی قابل مشاهده

نیز متمرکز شوید.

1. De Champeaux, D.

2. active state



شکل ۹-۲۱ یک باز نمایی از انتقال های وضعیت فعال

هر پیکان که در شکل ۹-۲۱ آمده نشانگر انتقالی از حالت فعال یک شیء به دیگری است. عناوین نشان داده شده برای هر پیکان نمایان دهنده حادثه‌ای هستند که باعث انتقال شده است. گر چه مدل وضعیت فعال دیدگاه مهمی در مورد تاریخچه زندگی شیء فراهم می‌کند اما ممکن است اطلاعات اضافی را برای رسیدن به شناخت عمیق‌تری از رفتار شیء لازم باشد. علاوه بر مشخص کردن حادثه‌ای که باعث ایجاد تغییر شده، تحلیل‌گر یک گارد یا محافظ و یک اقدام را نیز مشخص می‌سازد. گارد یک شرط بولین است که باید به منظور ایجاد انتقال برقرار شود. مثلاً، گارد برای انتقال از حالت «استراحت» به حالت «مقایسه» که در شکل ۹-۲۱ آمده توسط بررسی مورد استفاده، معین می‌گردد:

**If (password input = 4 digits) then make transition to comparing state;**

اگر ورودی رمز برابر چهار رقم بود پس انتقال به حالت مقایسه صورت گیرد.

به‌طور کلی، گارد یک انتقال عمل، معمولاً به مقدار یک یا چند صفت خاصه شیء بستگی دارد.

به عبارت دیگر، گارد به وضعیت منفعل و غیرفعال شیء بستگی دارد.

یک اقدام همزمان با انتقال وضعیت و به عنوان نتیجه آن رخ می‌دهد و معمولاً مستلزم یک یا چند

عملیات در شیء است. مثلاً، اقدام مربوط به واقعه «وارد کردن اسم رمز» عملی است که به شیء کلمه رمز

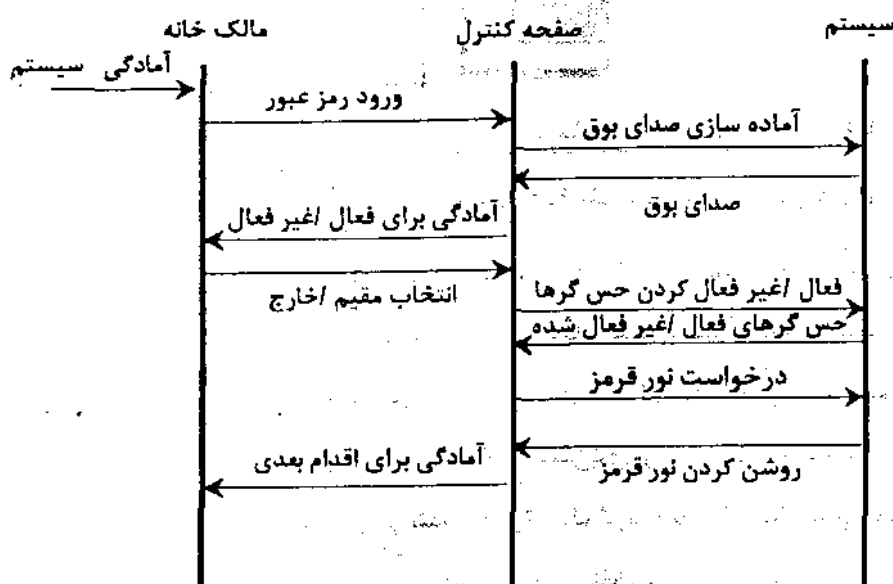
دسترسی یافته و مقایسه عدد به عدد برای ارزیابی کلمه رمز وارد شده، انجام می‌دهد.

1. at rest

2. action

دومین نوع نمایش رفتار در تحلیل شیء گرا نمایش خالتی را برای محصول یا سیستم در کل در نظر می گیرد. این نمایش در برگرفته یک مدل ساده پی گیری واقعه است که نشان می دهد چگونه حوادث باعث انتقال از شیء به شیء دیگر شده و یک نمودار تغییر حالت را نشان می دهد که رفتار پردازش هر شیء را مشخص می سازد.

وقتی رویدادها برای موارد کاربرد شناسایی شدند، تحلیل گر شکلی از چگونگی اثر این وقایع به منظور جریان یافتن از یک شیء به شیء دیگر را نشان می دهد. این کار که پی گیری واقعه است نسخه خلاصه نویسی شده مورد کاربرد است. این کار نمایانگر اشیای اصلی و رویدادهای عمده ای است که باعث جریان یافتن رفتار از شیء به شیء دیگر می شود.



شکل ۱۰-۲۱ یک نمودار ردگیری رویداد جزئی برای



یک انتقال از وضعیتی

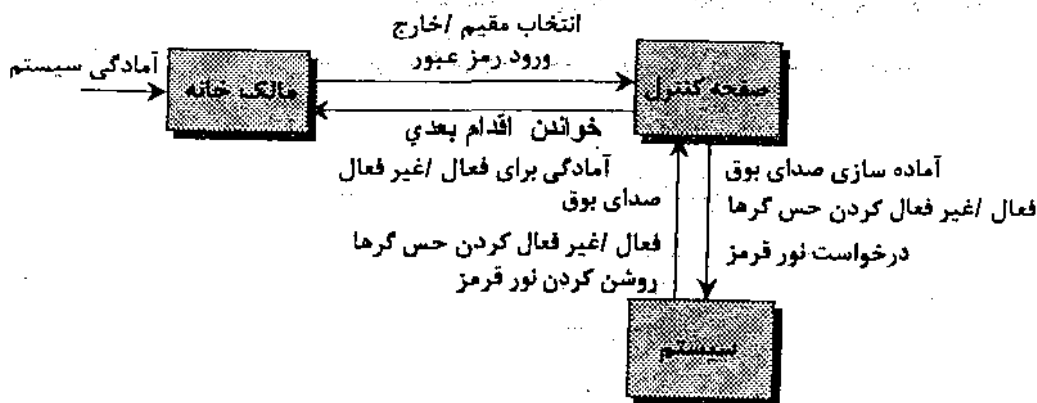
به وضعیت دیگر.

شکل ۱۰-۲۱ پی گیری نسبی یک واقعه را در مورد سیستم خانه امن نشان می دهد. هر یک از پیکانها نمایانگر یک رویداد بوده و چگونگی انتقال رفتار میان اشیای خانه امن را نشان می دهد. اولین واقعه یعنی آمادگی سیستم از محیط خارجی نشأت گرفته و رفتار بین اشیای خانه امن انتقال می یابد؛ صاحبخانه کلمه رمز را وارد می کند. حادثه با صدای بوق شروع می شود و این صدا نشانگر این است که رفتار در صورتی که کلمه رمز بی اعتبار باشد، انتقال می یابد. کلمه رمز درست منجر به جریان برگشتی به خود مالک خانه می شود. حوادث باقی مانده و مسیرشان با فعال شدن یا غیرفعال شدن سیستم، رفتار را ادامه می دهند.

نیازمند رویداد یک واقعه است. رویدادها در طبیعت دو ارزشی هستند (روی می دهند یا نمی دهند) و هنگامی روی می دهند که اشیاء با یکدیگر ارتباط برقرار می سازند.

UML از ترکیبی از نمودارهای حالت، نمودار توالی، نمودار همکاری و فعالیت برای نشان دادن رفتار دینامیک اشیاء و کلاس‌هایی که شناسایی شده و بخشی از مدل تحلیل هستند، استفاده می‌کند.

[BOO99]، [BEN99]، [ALH98]، [ERI98]



شکل ۲۱-۱۱ یک نمودار جریان رویداد جزئی برای خانه امن

## ۲۱-۷ خلاصه

روش‌های تحلیل شیءگرا مهندس نرم‌افزار را قادر می‌سازد مسئله‌ای را از لحاظ مشخصه‌های ایستا و پویای کلاس‌ها و ارتباطاتشان به‌عنوان جزءهای اولیه مدل‌سازی، بازنمایی کند. مثل روش‌های قدیمی‌تر تحلیل شیءگرای، UML یک مدل تحلیلی می‌سازد که دارای مشخصه‌های زیر است: (۱) نمایانگر کلاس‌ها و سلسله مراتب آنهاست. (۲) مدل‌های رابطه شیء را ایجاد می‌کند و (۳) مدل‌های رفتار شیء را نیز خلق می‌نماید.

تحلیل سیستم‌های شیءگرا در سطوح مختلف انتزاعی رخ می‌دهد. در سطح تجاری یا شرکتی، فنون مربوط به تحلیل شیءگرا می‌توانند با هم ادغام شده و همراه با رهیافت مهندسی فرآیند تجاری باشند. این تکنیک اغلب تحلیل دامنه نامیده می‌شود. در سطح برنامه کاربردی، مدل شیء روی نیازمندیهای به‌خصوص مشتری که بر ساخت برنامه تأثیر دارند، متمرکز می‌شود.

1. Booch, G.
2. Bennett, S.
3. Alhir, S.S.
4. Eriksson, H.E. and M.

فرآیند تحلیل شیءگرا با تعریف موارد کاربرد یعنی طرح‌هایی که چگونگی استفاده از سیستم شیءگرا را توصیف می‌کنند، شروع می‌شود. سپس از تکنیک مدل‌سازی CRC (کلاس - مسئولیت - همکاری) برای مستندسازی کلاس‌ها و صفات خاصه و عملیات آنها استفاده می‌شود. این کار هم‌چنین نظر اولیه را در مورد همکاری‌هایی که میان اشیاء رخ می‌دهد، ارائه می‌کند. مرحله بعدی در فرآیند تحلیل شیءگرا کلاس‌بندی اشیاء و ایجاد سلسله مراتب هر کلاس است. سیستم‌های فرعی (پکیج‌ها) را می‌توان برای در برگرفتن اشیای مربوطه استفاده نمود. مدل رابطه شیء، علامت چگونگی اتصال کلاس‌ها به یکدیگر را مهیا نموده و مدل رفتار شیء، نشانگر رفتار تک‌تک اشیاء و رفتار کلی سیستم شیءگراست.

## مسایل و نکاتی برای تفکر و تعمق بیشتر

۲۱-۱ یک یا چند کتاب مختص زبان مدل سازی متحد (UML) تهیه کنید و با استفاده از ابعاد مدل سازی پیشنهاد شده توسط فیشرمن و کمرر [FIC92] در بخش ۲۱-۱۵ آن را با تحلیل ساخت یافته (فصل ۱۲) مقایسه کنید.

۲۱-۲ از نمودار مدل سازی ایستا یا پویای UML، یک ارائه کلاسی داشته باشید. نمودار را در قالب یک مثال ساده ارائه دهید، ولی جزئیات کافی در خصوص بیشتر ویژگی‌های مهم شکل نموداری را تهیه نمایید.

۲۱-۳ برای هر یک از حیطه های زیر زیر یک تحلیل دامنه خلاصه شده تهیه کنید:

الف. سیستم ثبت اطلاعات دانشجویان دانشگاه

ب. یک برنامه کاربردی تجارت الکترونیک (برای مثال لباس، کتاب و ادوات الکترونیکی)

پ. خدمات مشتریان بانک

ت. سرویس مشتریان بانک

ث. بازی ویدئویی

ج. یک برنامه کاربردی پیشنهادی از طرف استادان

اطمینان حاصل کنید که برخی کلاسهای مربوط به حیطه مشترک در برنامه های کاربردی استفاده

شده اند

۲۱-۴ به زبان خودتان، اختلاف میان دیدگاههای پویا و ایستا را در یک سیستم شیء گرا شرح دهید.

۲۱-۵ یک Use-Case برای سیستم خانه امن که در این کتاب بحث شده است، بنویسید. Use-

cae باید سناریوی مورد نیاز برای تعریف یک منطقه امنیتی را پوشش دهد. منطقه امنیتی مشتمل بر مجموعه ای از حسگرهاست که می توان آنها را به صورت مجموعه ای فعال یا غیرفعال کرد (ترجیحا نه به صورت انفرادی). حداکثر ۱۰ منطقه امنیتی قابل تعریف خواهند بود. در اینجا خلاق و مبتکر باشید اما از حدود نابلوی کنترل خانه امن که در ابتدای این کتاب تعیین شده، فراتر نروید.

۲۱-۶ مجموعه ای از Use-Case ها را برای سیستم PHTRS که در مسئله ۱۲-۱۳ معرفی شده،

تهیه کنید. مفروضاتی را در خصوص نحوه محاوره کاربر با این سیستم برقرار سازید.

۲۱-۷ برای هر یک از برنامه های کاربردی زیر مجموعه ای از Use-Case ها را تهیه کنید:

الف. نرم افزاری برای دستیار رقمی شخصی همه منظوره (سازماندهنده کاری-زمانی-م)

ب. نرم افزاری به دلخواه خودتان جهت برای بازی ویدئویی

پ. نرم افزاری برای یک سیستم داخلی کنترل آب و هوای خودرو

ت. نرم افزاری برای یک سیستم راهبری خودرو

ث. سیستم (محصولی) به پیشنهاد استادان

[RUM99] Rumbaugh, J., I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1999.

[SUL94] Sullo, G.C., *Object Engineering*, Wiley, 1994.

[TAY95] Taylor, D.A., *Business Engineering with Object Technology*, Wiley, 1995.

[WIR90] Wirfs-Brock, R., B. Wilkerson, and L. Weiner, *Designing Object-Oriented Software*, Prentice-Hall, 1990.

چند ساعته بر زمینه‌های مختلف کاربردی تحقیق کنید و با تکی چند از هم‌کلاسی‌های خود یک نشست FAST (فصل ۱۱) داشته باشید و نیازمندیهای اساسی را تعیین کنید (استادتان، باریگر شما در این هماهنگی خواهد بود).

۸-۲۱ مجموعه کلمه‌ای از کارت‌های شاخص مدل CRC را برای محصول یا سیستمی که در مسئله

### خواندنیهای دیگر و منابع اطلاعاتی

Use-cases form the foundation of object-oriented analysis, regardless of the OOA method that is chosen. Books by Rosenberg and Scott (*Use Case Driven Object Modeling with UML: A Practical Approach*, Addison-Wesley, 1999); Schneider, Winters, and Jacobson (*Applying Use Cases: A Practical Guide*, Addison-Wesley, 1998); and Texel and Williams (*Use Cases Combined With Booch/OMT/UML: process and Products*, Prentice-Hall, 1997) provide worthwhile guidance in the creation and use of this important requirements elicitation and representation mechanism.

Virtually every recent book published on object-oriented analysis and design emphasizes UML. Those serious about applying UML in their work should acquire [BOO99], [RUM99], and [JAC99]. In addition, the following books are representative of dozens written on UML technology:

Douglass, B., *Real-Time UML: Developing Efficient Objects for Embedded Systems*, Addison-Wesley, 1999.

Fowler, M. and K. Scott, *UML Distilled*, 2nd ed., Addison-Wesley, 2000.

Odell, J.J. and M. Fowler, *Advanced Object-Oriented Analysis and Design Using UML*, SIGS Books, 1998.

Oestereich, B., *Developing Software with UML: Object-Oriented Analysis and Design in Practice*, Addison-Wesley, 1999.

A wide variety of information sources on object-oriented analysis and related subjects is available on the Internet. An up-to-date list of World Wide Web references that are relevant to OOA can be found at the SEPA Web site:

<http://www.mhhe.com/engcs/compsci/pressman/resources/OOA.mhtml>