

سازمان سما

و اسسهه دانشگاه آزاد اسلامی
دانشگاه سما واحد حاجی آباد



مهندسی نرم افزار

منبع : مهندسی نرم افزار(پرسمن)

دکتر شیر افکن

حمیدرضا رضاپور

مهندسی نرم افزار

فصل ۶: مدل سازی خواسته ها:

سناریوها، اطلاعات و کلاس های تحلیل

نگاهی گذرا

مدل سازی خواسته ها چیست؟

سخنان مکتوب، وسیله ای عالی برای برقراری ارتباط به شمار می روند، ولی ضرورتا بهترین راه برای نمایش خواسته های مربوط به یک نرم افزار کامپیووتری نیستند.

در مدل سازی خواسته ها تلفیقی از شکل های متنی و نموداری برای به تصویر کشیدن خواسته ها استفاده می شود، به شیوه ای که درک آن آسان تر باشد و مهمتر از آن، به سهولت بتوان آن را برای تصحیح ، تکمیل و سازگاری مورد بازبینی قرار داد.

چه کسی آن را انجام می دهد؟

این مدل را یک مهندس نرم افزار (تحلیل گر) با به کار گیری خواسته های استخراج شده از مشتری می سازد.

چرا اهمیت دارد؟

برای اعتبار سنجی خواسته های نرم افزار، باید آنها را از چند دیدگاه متفاوت بررسی کنیم.

- مدل های مبتنی بر سناریو
- مدل های داده ای
- مدل های مبتنی بر کلاس

در هر کدام از این مدل ها خواسته ها از بعدی متفاوت به نمایش درمی آید و از این رو ، احتمال برملا شدن خطاهای روش ناسازگاری ها و کشف جا افتادگی ها افزایش می یابد.

مراحل کار کدام است؟

مدل های مبتنی بر سناریو، سیستم را از دیدگاه کاربر به نمایش می گذارند.

مدل سازی داده ها فضای اطلاعاتی را به نمایش درمی آورد و اشیای داده را که نرم افزار دستکاری می کند و همچنین روابط میان آنها را به تصویر می کشد.

مدل سازی مبتنی بر کلاس ها به تعریف اشیا، صفات و روابط می پردازد.

پس از اینکه مدل های مقدماتی تهیه شدند مورد پالایش و تحلیل قرار می گیرند تا به وضوح، کمال و سازگاری لازم برسند.

محصول کاری چیست؟

آرایه گسترده‌ای از فرم‌های متنی و نموداری را می‌توان برای مدل خواسته‌ها برگزید.
هر کدام از این نمایش‌ها، دیدگاهی از یک یا چند عنصر مدل فراهم می‌سازند.

چطور اطمینان حاصل کنیم که درست از عهده کار برآمده‌ایم؟

محصولات کاری از مدل سازی خواسته‌ها را باید از نظر صحت، کمال و سازگاری بازبینی کرد.
این محصولات باید منعکس کننده نیازهای همه‌ی طرف‌های ذی نفع باشند و بستری برای انجام طراحی
فراهم سازند.

تحلیل خواسته ها

R EQUIREMENTS A NALYSIS

تحلیل خواسته ها:

به تعیین مشخصات خصوصیات عملیاتی نرم افزار منجر می شود،

واسط نرم افزار با سایر عناصر سیستم را مشخص می کند و

قید و بندهایی را که نرم افزار باید رعایت کند، تعیین می نماید.

تحلیل خواسته ها به شما این امکان را می دهد که طی وظایف دریافت، استخراج و چانه زنی جزئیات خواسته های پایه

را تعیین کنید.

کنش مدل سازی خواسته به یک یا چند نوع از مدل های زیر می انجامد:

- مدل های مبتنی بر سناریو از دیدگاه کنش گران گوناگون سیستم.
- مدل های داده ای که دامنه اطلاعاتی مسئله را تصویر می کنند.
- مدل های مبتنی بر کلاس ها که کلاس های شیء گرا و شیوه همکاری این کلاس ها برای دستیابی به خواسته های سیستم را به نمایش می گذارند.
- مدل های جریان گرا که عناصر عملیاتی سیستم و چگونگی تبدیل داده ها توسط این عناصر را به هنگام حرکت در سیستم نمایش می دهند.
- مدل های رفتاری که چگونگی رفتار نرم افزار را به عنوان نتیجه ای از رویدادهای بیرونی به تصویر می کشند.

مدل ها

- | | |
|------------------------------|---------------------------------------|
| <i>Scenario-based models</i> | ۱- مدل سازی مبتنی بر سناریو |
| <i>Data models</i> | ۲- مدل سازی داده ها |
| <i>Class-oriented models</i> | ۳- مدل سازی کلاس ها |
| <i>Flow-oriented models</i> | ۴- مدل های جریان گرا |
| <i>Behavioral models</i> | ۵- مدل های رفتاری |
| | ۶- مدل سازی مبتنی بر الگوها |
| | ۷- مدل های مربوط به برنامه های تحت وب |

فلسفه و اهداف کلی

Overall Objectives and Philosophy

Throughout requirements modeling, your primary focus is on *what*, not *how*.

مدل خواسته ها باید به سه هدف دست پیدا کند :

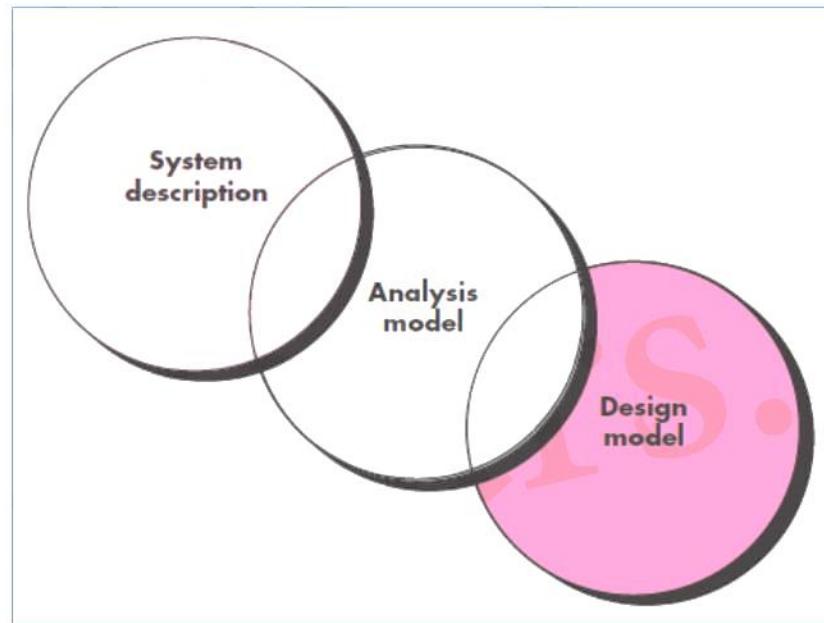
- ۱- توصیف آنچه که مشتری نیاز دارد.
- ۲- ایجاد مبنایی برای تهیه طراحی نرم افزار
- ۳- تعریف مجموعه ای از خواسته ها که پس از ساخته شدن نرم افزار بتوان آنها را اعتبار سنجی کرد.

مدل خواسته ها به عنوان پلی میان توصیف سیستم و مدل طراحی

The requirements model as a bridge between the system description and the design model

مدل آنالیز، پلی است میان توصیف در سطح سیستم و یک طراحی نرم افزار.

طراحی نرم افزار ، معماری کاربرد نرم افزار، واسطه های کاربری و ساختار را در سطح مولفه ها توصیف می کند.



همه عناصر در مدل خواسته ها به طور مستقیم تا بخش هایی از مدل طراحی قابل ردیابی خواهند بود.

مقداری از طراحی به عنوان بخشی از تحلیل و مقداری از تحلیل در طول طراحی انجام می شود.

قواعد ساده تحلیل

بهتر است هنگام ایجاد مدل تحلیل ، قواعد زیر رعایت شوند:

- مدل باید خواسته هایی را کانون توجه قرار دهد که در دامنه تجاری یا مسئله، قابل مشاهده باشند. سطح انتزاع باید نسبتا بالا باشد.
- هر عنصر از مدل خواسته ها باید در کل چیزی به درک ما از خواسته های نرم افزار بیفزاید و دیدی از دامنه اطلاعاتی عملکرد و رفتار سیستم فراهم سازد.
- ملاحظات زیر ساختی و سایر مدل های غیر عملیاتی را تا طراحی به تأخیر اندازید.
- ارتباط ها را در سرتاسر سیستم به حداقل برسانید.
- مدل خواسته ها باید حتما رضایت همه طرف های ذی نفع را جلب کند.
- سادگی مدل را تا حد امکان حفظ کنیم.

تحلیل دامنه

Domain Analysis

تحلیل دامنه نرم افزار عبارت است از :

شناسایی، تحلیل و تعیین مشخصات خواسته های رایج از یک دامنه کاربردی خاص، معمولاً برای استفاده مجدد در چندین پروژه که در همان دامنه‌ی کاربردی قرار دارند..

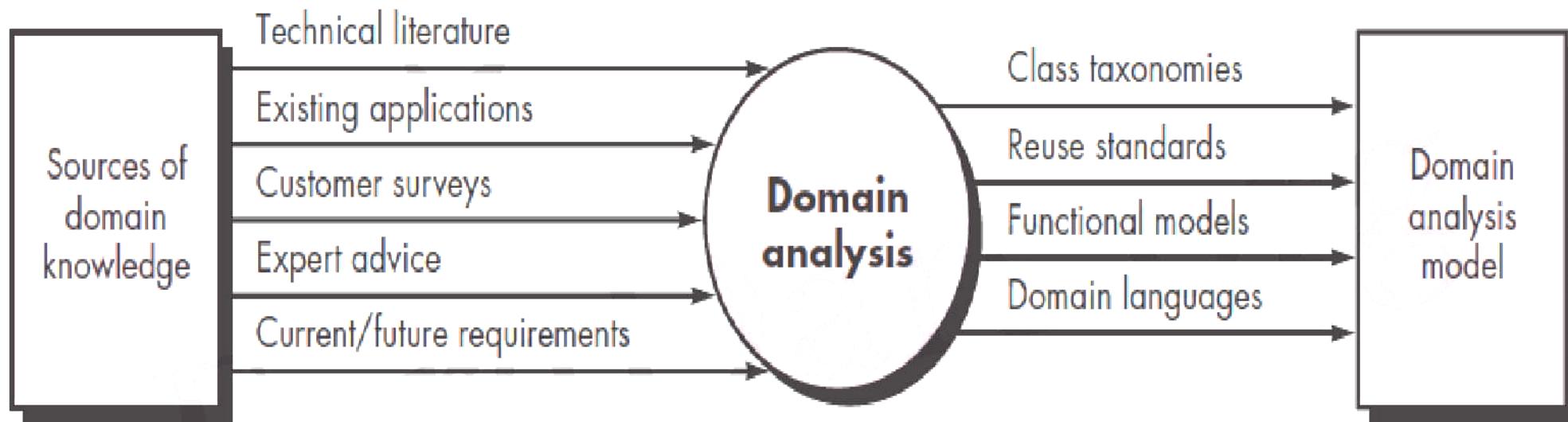
دامنه کاربرد خاص می‌تواند از هوانوردی تا بانکداری، از بازی‌های چند رسانه‌ای تا نرم افزارهای تعبیه شده در دستگاه‌های پزشکی را در برگیرد.

هدف تحلیل دامنه :

یافتن یا ایجاد کلاس‌های تحلیل که دارای کاربردی گسترده‌اند و می‌توان دوباره از آنها استفاده کرد.

ورودی و خروجی برای تحلیل دامنه

منابع اطلاعاتی دامنه، مورد نظر خواهی قرار می گیرد تا اشیای قابل استفاده مجدد در آن دامنه شناسایی گردد.



تحلیل دامنه

SAFEHOME

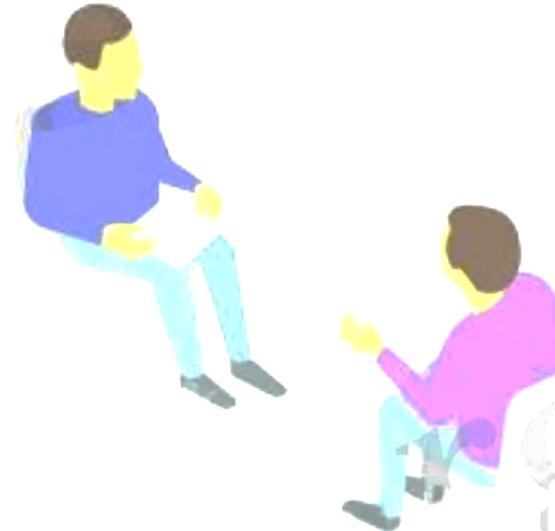


Domain Analysis

The scene: Doug Miller's office, after a meeting with marketing.

The players: Doug Miller, software engineering manager, and Vinod Raman, a member of the software engineering team.

مدیر مهندسی نرم افزار



عضو تیم نرم افزار

روش های مدل سازی خواسته ها

Requirements Modeling Approaches

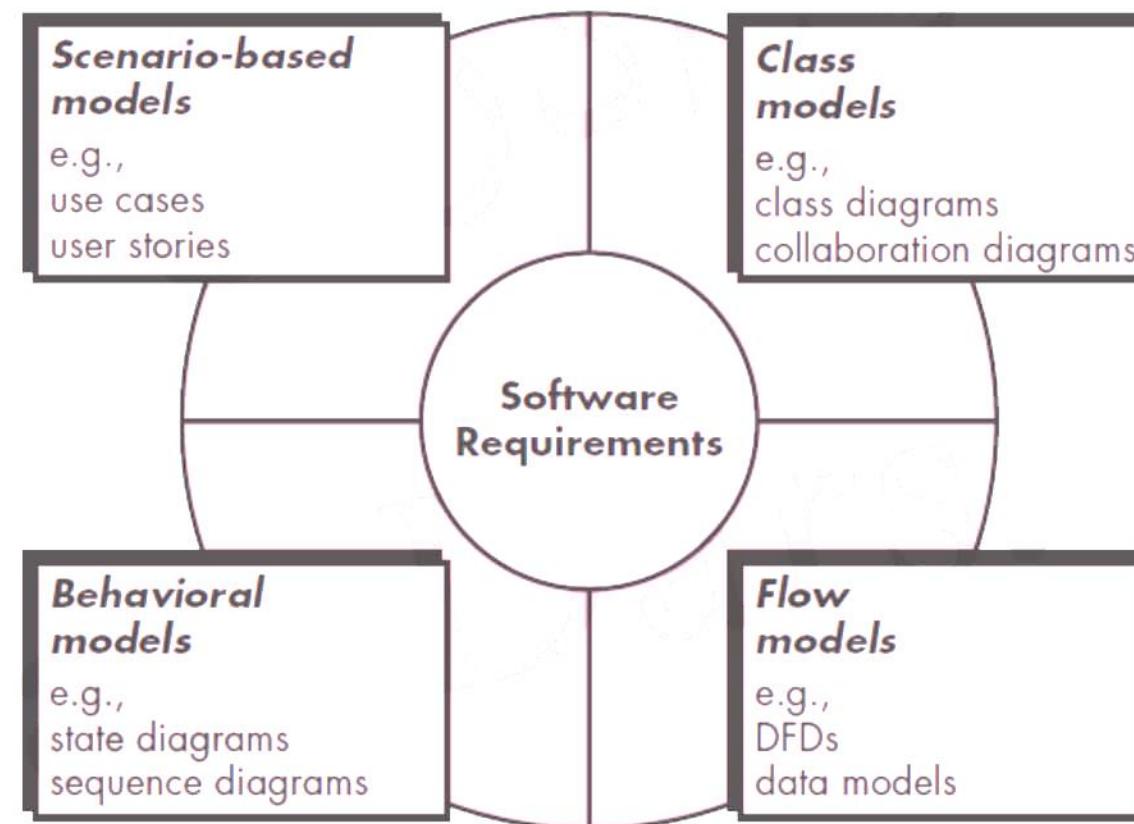
در یک نما از مدل سازی خواسته ها، که به تحلیل ساخت یافته مرسوم است، داده ها و فرایندهایی که این داده ها را تبدیل می کنند، به عنوان موجودیت هایی مجزا در نظر گرفته می شوند.

رویکرد دوم برای مدل سازی تحلیل، که تحلیل شیء گرا نامیده می شود، بر تعریف کلاس ها و شیوه همکاری آنها با یکدیگر برای برآورده ساختن خواسته های مشتری تاکید دارد.

و فرایند یکپارچه عمدتاً شیء گرا هستند. **UML**

عناصر مدل تحلیل

هر عنصر از مدل خواسته ها از دیدگاهی متفاوت نشان می دهد.



مدل سازی مبتنی بر سناریو

SCENARIO-BASED MODELING

ایجاد یک use case مقدماتی

- (1) what to write about,
- (2) how much to write about it,
- (3) how detailed to make your description,
- (4) how to organize the description?

توسعه یک سناریوی کاربری مقدماتی

برای قابلیت پایشی SafeHome surveillance function

SAFEHOME



The scene: A meeting room, during the second requirements gathering meeting.

The players: Jamie Lazar, software team member; Ed Robbins, software team member; Doug Miller, software engineering manager; three members of marketing; a product engineering representative; and a facilitator.

SafeHome surveillance function

Facilitator



نمای شستی: thumbnail views

قابلیت پایش در محصول **safe home** آنها را **homeowner** قابلیت های زیر را مشخص می کند که کنش گر آنها را انجام می دهد:

- انتخاب دوربین برای مشاهده
- درخواست تصاویر کوچکی از همه دوربین ها
- به نمایش درآوردن نمای دوربین ها در یک پنجره pc
- کنترل زاویه و زوم یک دوربین مشخص
- ضبط انتخابی خروجی دوربین ها
- پخش خروجی دوربین ها
- دستیابی به پایش دوربین ها از طریق اینترنت

ACS-DCV مربوط به Use case

دستیابی به پایش دوربینی از طریق اینترنت - نمایش خروجی دوربین ها

کنش گر: homeowner

- ۱- صاحبخانه وارد وب سایت **SafeHome Products** می شود.
- ۲- صاحبخانه نام کاربری (**user ID**) خودش را وارد می کند .
- ۳- صاحبخانه دو کلمه عبور وارد می کند.
- ۴- سیستم همه های عملیاتی اصلی را به نمایش در می اورد.
- ۵- صاحبخانه پایش (**surveillance**) را از دکمه های اصلی انتخاب می کند.
- ۶- صاحبخانه ”**pick a camera**“ را برمی گزیند. (انتخاب دوربین)
- ۷- سیستم، نقشه ساختمان را نمایش می دهد.
- ۸- صاحبخانه ایکون یکی از دوربین ها را از روی نقشه انتخاب می کند.
- ۹- صاحبخانه دکمه **view** را انتخاب می کند.
- ۱۰- سیستم ، یک پنجره نمایش ظاهر می کند که با شماره شناسایی دوربین مشخص می شود.
- ۱۱- سیستم، خروجی دوربین را در پنجره نمایش با سرعت یک فریم در ثانیه نشان می دهد.

پالیش یک use case مقدماتی

Refining a Preliminary Use Case

- *Can the actor take some other action at this point?*
- *Is it possible that the actor will encounter some error condition at this point?*
- *Is it possible that the actor will encounter some other behavior at this point ?*

- ۱- صاحبخانه وارد وب سایت SafeHome Products می شود.
- ۲- صاحبخانه نام کاربری (user ID) خودش را وارد می کند.
- ۳- صاحبخانه دو کلمه عبور وارد می کند.
- ۴- سیستم همه دکمه های عملیاتی اصلی را به نمایش در می اورد.
- ۵- صاحبخانه پایش (surveillance) را از دکمه های اصلی انتخاب می کند.

- ۶- صاحبخانه ”pick a camera“ را برمی گزیند.
- ۷- سیستم، نقشه ساختمان را نمایش می دهد.

- ۸- صاحبخانه ایکون یکی از دوربین ها را از روی نقشه انتخاب می کند.
- ۹- صاحبخانه دکمه view را انتخاب می کند.
- ۱۰- سیستم ، یک پنجره نمایش ظاهر می کند که با شماره شناسایی دوربین مشخص می شود.
- ۱۱- سیستم، خروجی دوربین را در پنجره نمایش با سرعت یک فریم در ثانیه نشان می دهد.

نوشتن یک use case رسمی

Use case های غیر رسمی گاهی برای مدل سازی خواسته ها کفایت می کنند.

ولی هنگامی که یک use case شامل فعالیتی مهم می شود یا مجموعه پیچیده ای از مراحل را با تعداد چشمگیری از استثناهای توصیف می کند، روش رسمی تر ممکن است مطلوب باشد.



Use Case Template for Surveillance

Use case: Access camera surveillance via the Internet—display camera views (ACS-DCV)

Iterations: 2, last modification: January 14 by V. Raman.

Primary actor: Homeowner.

Goal in context: To view output of camera placed throughout the house from any remote location via the Internet.

Preconditions: System must be fully configured; appropriate user ID and passwords must be obtained.

Trigger: The homeowner decides to take a look inside the house while away.

Scenario:

1. The homeowner logs onto the SafeHome Products website.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects the "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Exceptions:

1. ID or passwords are incorrect or not recognized—see use case **Validate ID and passwords**.
2. Surveillance function not configured for this system—system displays appropriate error message; see use case **Configure surveillance function**.
3. Homeowner selects "View thumbnail snapshots for all cameras"—see use case **View thumbnail snapshots for all cameras**.
4. A floor plan is not available or has not been configured—display appropriate error message and see use case **Configure floor plan**.
5. An alarm condition is encountered—see use case **Alarm condition encountered**.

Priority:

Moderate priority, to be implemented after basic functions.

When available: Third increment.

Frequency of use: Moderate frequency.

Channel to actor: Via PC-based browser and Internet connection.

Secondary actors: System administrator, cameras.

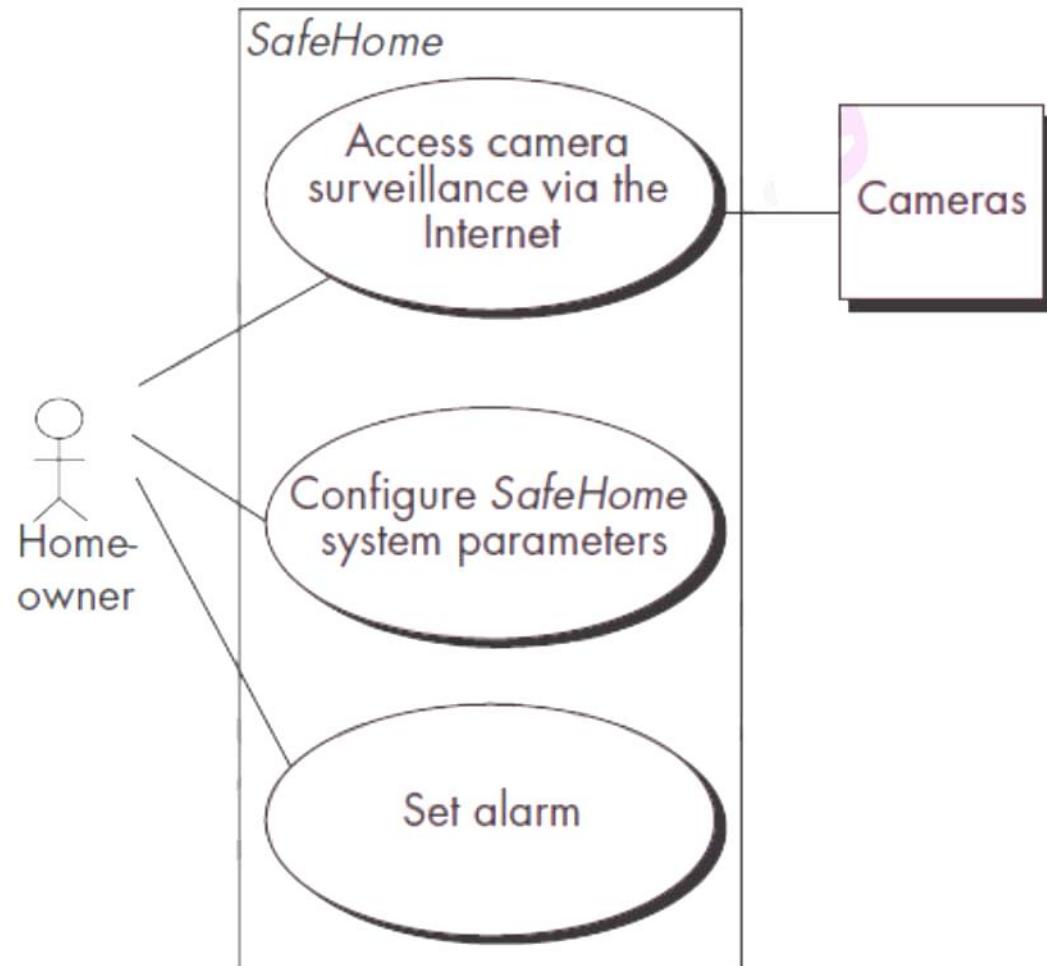
Channels to secondary actors:

1. System administrator: PC-based system.
2. Cameras: wireless connectivity.

Open issues:

1. What mechanisms protect unauthorized use of this capability by employees of SafeHome Products?
2. Is security sufficient? Hacking into this feature would represent a major invasion of privacy.
3. Will system response via the Internet be acceptable given the bandwidth required for camera views?
4. Will we develop a capability to provide video at a higher frames-per-second rate when high-bandwidth connections are available?

نمودار یک مقدماتی برای سیستم use case



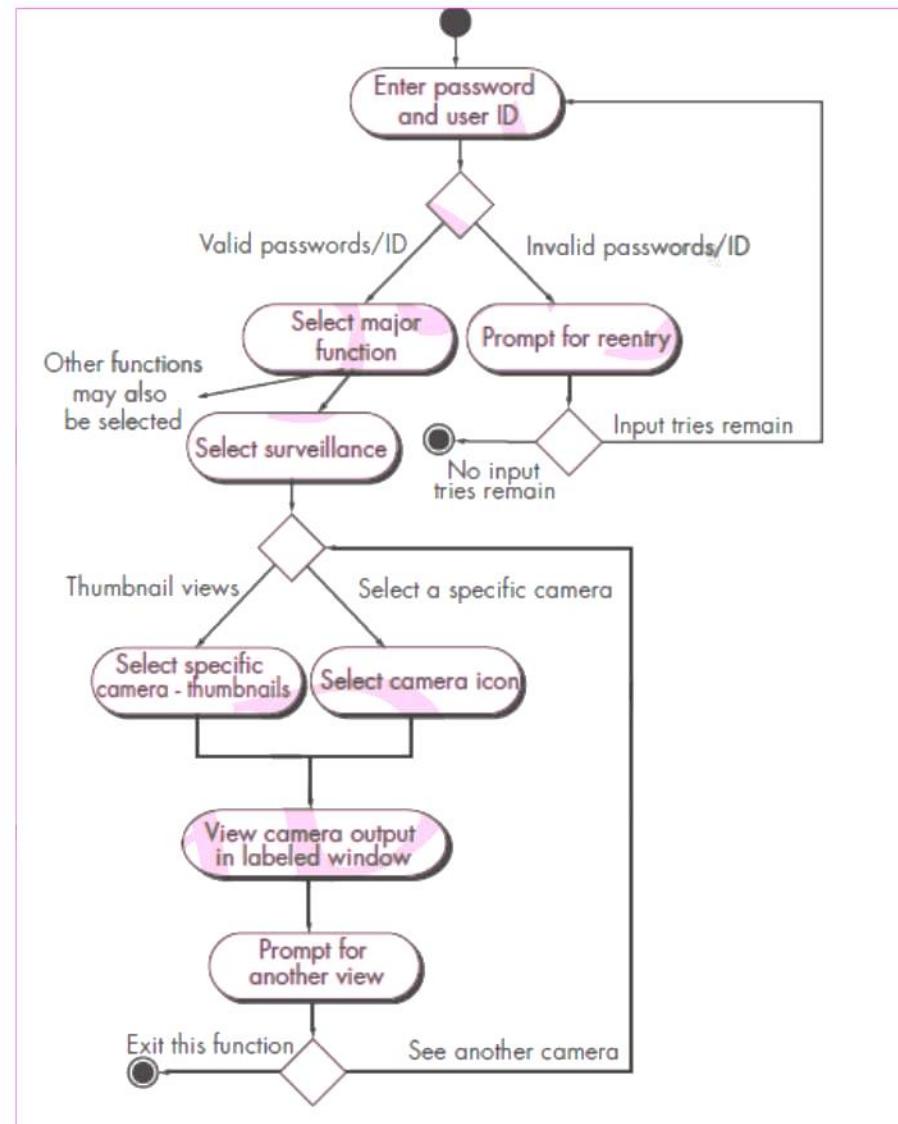
مدل های UML که use case را تکمیل می کنند

وضعیت های زیادی وجود دارد که مدل های مبتنی بر متن ممکن است اطلاعات را واضح ارائه ندهد.

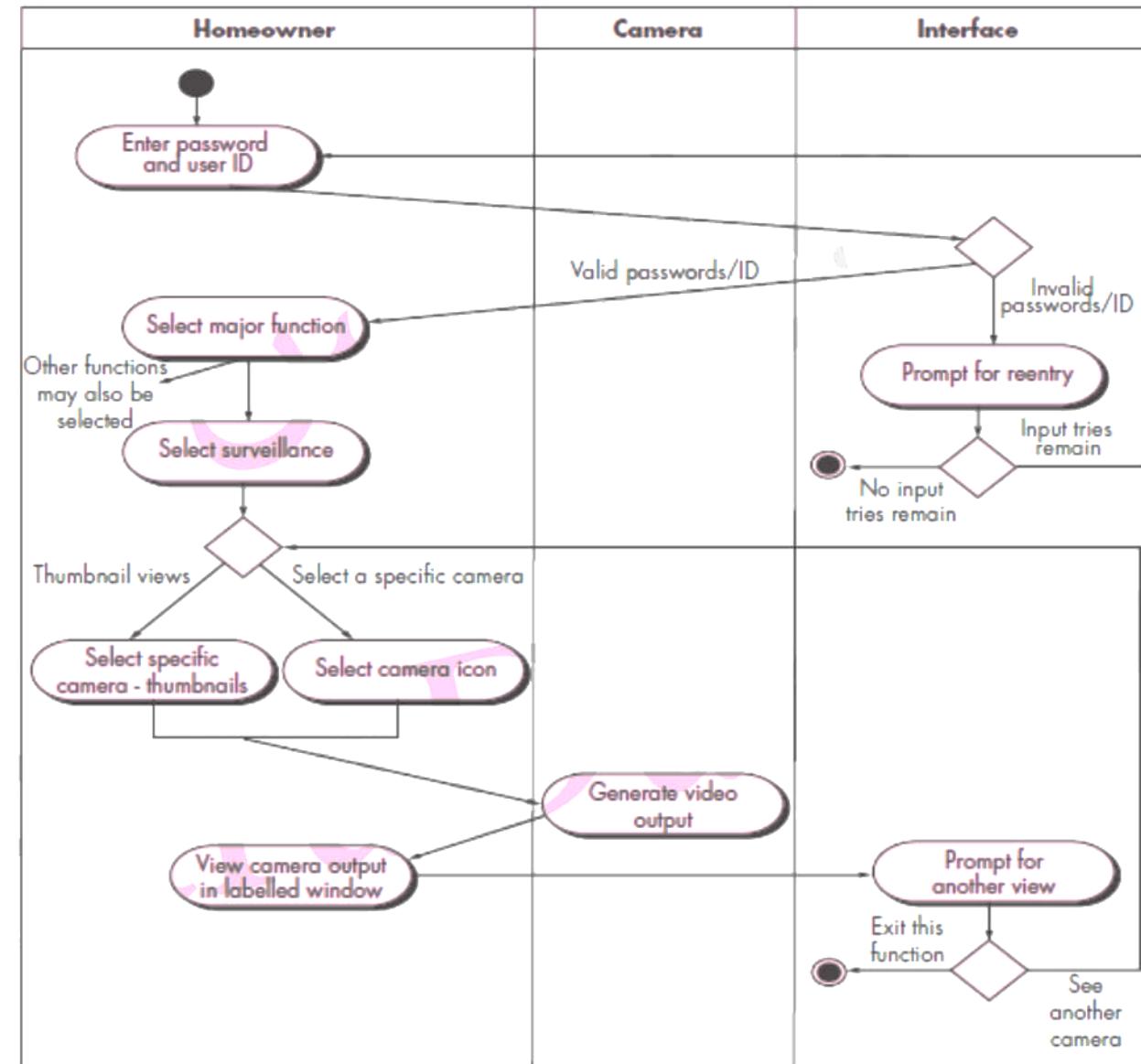
توسعه نمودار فعالیت ها

نمودارهای بخش بندی

نحوه فعالیت ها برای دستیابی به پایش دوربین ها از طریق اینترنت - عملکرد مشاهده دوربین



نمودار بخش بندی برای دستیابی به پایش دوربین ها از طریق اینترنت - عملکرد مشاهده دوربین



مفاهیم مدل سازی داده ها

DATA MODELING CONCEPTS

مهندس نرم افزار یا تحلیل گر، همه اشیای داده را که در سیستم پردازش می شوند، روابط میان اشیای داده و سایر اطلاعات مرتبط با این روابط را تعیین می کند.

نمودار موجودیت-ارتباط (ERD) به این مسائل می پردازد و همه اشیای داده را که در یک برنامه کاربردی، وارد، ذخیره، تبدیل و تولید می شوند، به نمایش می گذارند.

entity-relationship diagram (ERD)

اشیای داده

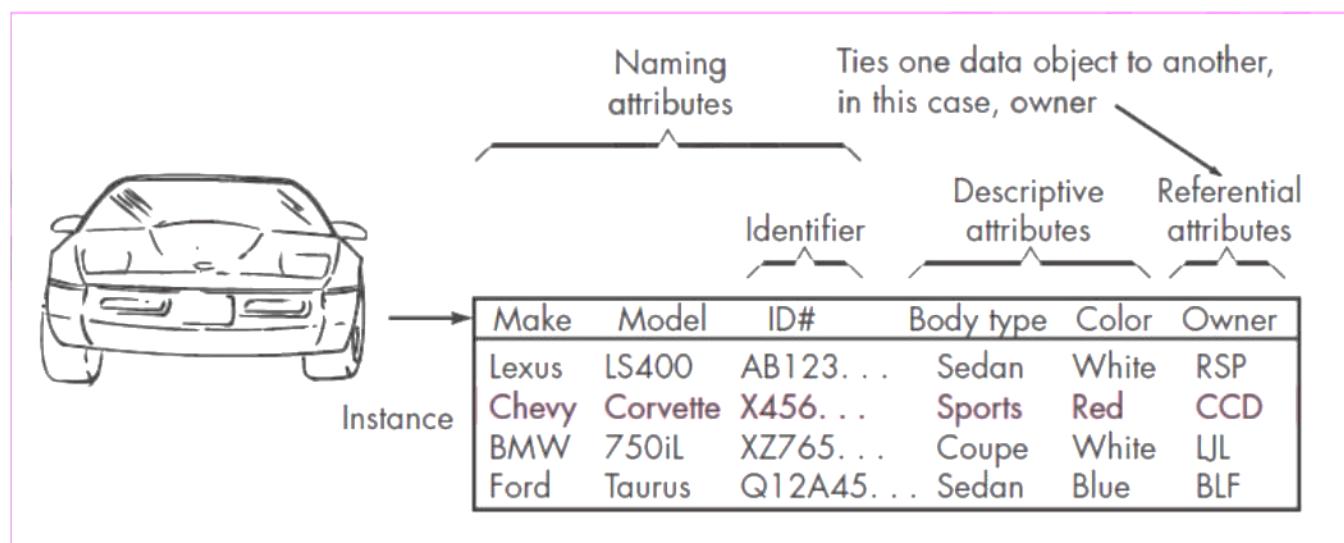
Data Objects

شیء داده نمایشی از اطلاعات مرکب است که باید نرم افزار آنها را بفهمد.

منظور از اطلاعات مرکب چیزی است که دارای چند صفت یا خاصیت متفاوت باشد.

شیء داده تنها داده ها را پنهان سازی می کند.

در داخل یک شیء داده هیچ آدرسی برای عملیات قابل انجام روی داده ها وجود ندارد.
بنابراین، شیء داده را می توان به صورت جدول شکل زیر به نمایش گذاشت.



صفات داده ها

Data Attributes

صفات داده ها، خواص یک شیء داده را تعریف می کنند.

از آنها می توان برای موارد زیر استفاده کرد :

۱- نامگذاری نمونه ای از شیء داده ای

۲- توصیف نمونه

۳- ارجاع به نمونه ای دیگر در جدولی دیگر

ارتباطات

Relationships



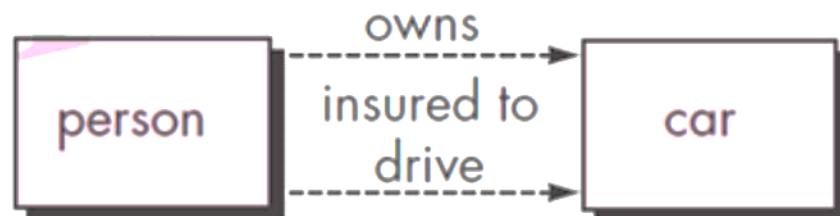
اشیای داده به طرق گوناگون بهم متصل می شوند.

دو شیء داده person و car را در نظر بگیرید.

می توانید مجموعه ای از روابط جفتی میان اشیا تعیین کنید که این ارتباط ها را مشخص کند.

برای مثال

- شخصی صاحب یک خودرو است.
- شخصی مجوز رانندگی خودرو را دارد.



مدل سازی مبتنی بر کلاس ها

CLASS-BASED MODELING

- classes and objects, عناصر مدل :
- attributes,
- operations,
- class-responsibility-collaborator (CRC) models,
- collaboration diagrams,
- packages.

شناسایی کلاس های تحلیل

Identifying Analysis Classe

می توانیم شناسایی کلاس ها را با بررسی سناریوهای کاربری آغاز کنیم که به عنوان بخشی از مدل خواسته ها توسعه می یابند و use case های تهیه شده برای سیستم را تجزیه گرامی کنیم.

کلاس ها با خط کشیدن زیر اسم ها یا عبارت های اسمی و وارد کردن آنها در یک جدول ساده تعیین می شوند.

کلاس های تحلیل، خود را به یکی از طرق زیر نشان می دهند:

- موجودیت های خارجی که اطلاعات مورد استفاده یک سیستم کامپیوتری را تولید یا مصرف می کنند.
- چیزهایی که بخشیء از دامنه اطلاعاتی مسئله اند.
- رخدادها یا رویدادهایی که در حیطه ای عملیاتی سیستم به وقوع می پیوندند.
- نقش هایی که توسط افراد درحال تعامل با سیستم ایفا می شوند.
- واحدهای سازمانی که به کاربردی خاص مربوط می شوند.
- مکان هایی که حیطه مسئله و عملکرد کلی سیستم را تعیین می کند.
- ساختارهایی که کلاسی از اشیا یا کلاس های مرتبطی از اشیا را تعریف می کند.

روايت پردازش

processing narrative

برای اينکه نشان دهيم کلاس های تحليل را چگونه می توان طی مراحل اوليه مدل سازی تعریف کرد،
تجزیه گرامری روایت پردازش قابلیت امنیتی در محصول **SafeHome** را در نظر بگیرید.
زیر اسم ها خط کشیده شده است.
 فعل ها به صورت ایتالیک نشان داده می شوند.

The SafeHome security function enables the homeowner to *configure* the security system when it is *installed*, *monitors* all sensors connected to the security system, and *interacts* with the homeowner through the Internet, a PC, or a control panel.

During installation, the SafeHome PC is used to *program* and *configure* the system. Each sensor is assigned a number and type, a master password is programmed for *arming* and *disarming* the system, and telephone number(s) are *input* for *dialing* when a sensor event occurs.

When a sensor event is *recognized*, the software *invokes* an audible alarm attached to the system. After a delay time that is *specified* by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, *provides information* about the location, *reporting* the nature of the event that has been detected. The telephone number will be *redialed* every 20 seconds until telephone connection is obtained.

The homeowner *receives* security information via a control panel, the PC, or a browser, collectively called an interface. The interface *displays* prompting messages and system status information on the control panel, the PC ,or the browser window. Homeowner interaction takes the following form . . .

با استخراج اسم ها، چند کلاس بالقوه می توان پیشنهاد داد:

Potential Class	General Classification
homeowner	role or external entity
sensor	external entity
control panel	external entity
installation	occurrence
system (alias security system)	thing
number, type	not objects, attributes of sensor
master password	thing
telephone number	thing
sensor event	occurrence
audible alarm	external entity
monitoring service	organizational unit or external entity

انتخاب کلاس های قانونی

Potential Class

homeowner

sensor

control panel

installation

system (alias security function)

number, type

master password

telephone number

sensor event

audible alarm

monitoring service

Characteristic Number That Applies

rejected: 1, 2 fail even though 6 applies

accepted: all apply

accepted: all apply

rejected

accepted: all apply

rejected: 3 fails, attributes of sensor

rejected: 3 fails

rejected: 3 fails

accepted: all apply

accepted: 2, 3, 4, 5, 6 apply

rejected: 1, 2 fail even though 6 applies

۱- اطلاعات نگه داری شده

۲- سرویس های لازم

۳- صفات چندگانه

۴- صفات مشترک

۵- عملیات مشترک

۶- خواسته های اساسی

نحوه تعیین اینکه یک کلاس بالقوه یک کلاس تحلیل است؟

۱- اطلاعات نگه داری شده

کلاس بالقوه، تنها در صورتی در تحلیل مفید خواهد شد که به خاطر سپردن اطلاعات مربوط به آن، برای عملکرد سیستم ضروری باشد.

۲- سرویس های لازم

کلاس بالقوه باید دارای مجموعه ای از عملیات قابل شناسایی باشد که بتوانند مقدار صفات آن را به طریقی تغییر دهنند.

۳- صفات چندگانه

۴- صفات مشترک

مجموعه ای از صفات که برای کلاس های بالقوه، قابل تعریف است و این عملیات ها در همه نمونه های کلاس مصدق دارند.

۵- عملیات مشترک

مجموعه ای از عملیات ها که برای کلاس های بالقوه، قابل تعریف است و این صفات در همه نمونه های کلاس تعریف می شوند.

۶- خواسته های اساسی

موجودیت های خارجی که در فضای مسئله ظاهر می شوند و اطلاعات ضروری جهت عملکرد هر راهکار برای سیستم را تولید یا مصرف می کنند،

همچنین در مدل خواسته ها همواره به عنوان کلاس تعریف می شوند.

مشخص کردن صفات

Specifying Attributes

صفات، مجموعه ای از اشیای داده هستند که یک کلاس را به طور کامل در حیطه‌ی مساله تعریف می‌کنند.

name,
position,
batting average,
fielding percentage,
years played,
games played



average salary
mailing address
credit toward full vesting
pension plan options chosen

برای توسعه مجموعه ای با معنی از صفات برای یک کلاس تحلیل باید هر کدام از use case ها را مطالعه کنید و آن چیزهایی را انتخاب کنید که به طور منطقی به کلاس تعلق دارند.

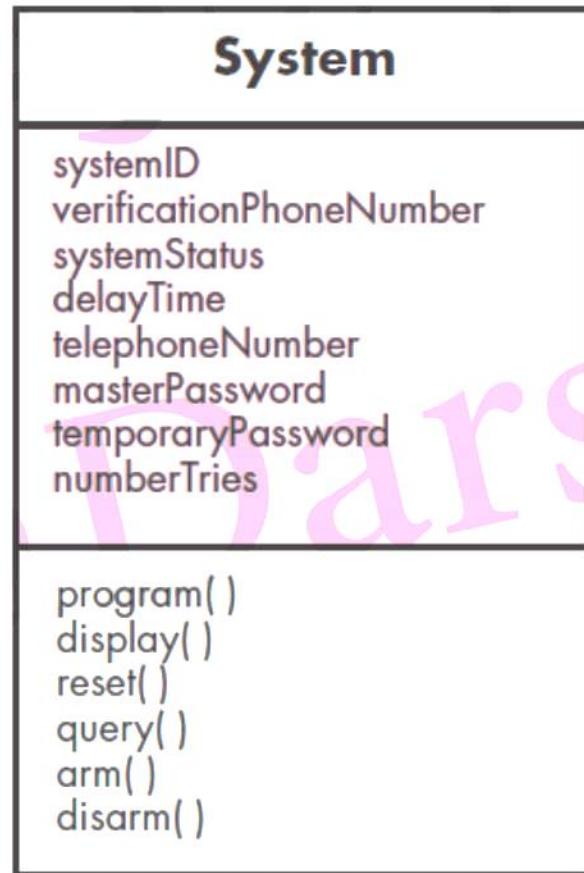
نمایش اقلام
داده ای مرکب:

کلاس SafeHome برای system

identification information = system ID + verification phone number + system status

alarm response information = delay time + telephone number

activation/deactivation information = master password + temporary password + number tries



تعريف عمليات ها

Defining Operations

عمليات ها، رفتار شيء را تعريف می کنند.

عمليات ها را در چهار گروه دسته بندی می کنند:

۱- عملياتی که داده ها را به طريقي دستکاري می کنند.

۲- عمليات هايي که محاسبه انجام می دهند.

۳- عمليات هايي که درباره حالت يك شيء تحقيق می کنند.

۴- عمليات هايي که يك شيء را برای وقوع يك رویداد کنترل کننده پايش می کنند.

عمليات باید از ماهیت همبستگی ها و صفات کلاس اگاهی داشته باشد

جدا سازی افعال

- That an *assign()* operation is relevant for the **Sensor** class.
- That a *program()* operation will be applied to the **System** class.
- That *arm()* and *disarm()* are operations that apply to **System** class.

The SafeHome security function enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system, and interacts with the homeowner through the Internet, a PC, or a control panel.

During installation, the SafeHome PC is used to program and configure the system. Each sensor is assigned a number and type, a master password is programmed for arming and disarming the system, and telephone number(s) are input for dialing when a sensor event occurs.

When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location, reporting the nature of the event that has been detected. The telephone number will be redialed every 20 seconds until telephone connection is obtained.

The homeowner receives security information via a control panel, the PC, or a browser, collectively called an interface. The interface displays prompting messages and system status information on the control panel, the PC ,or the browser window. Homeowner interaction takes the following form . . .



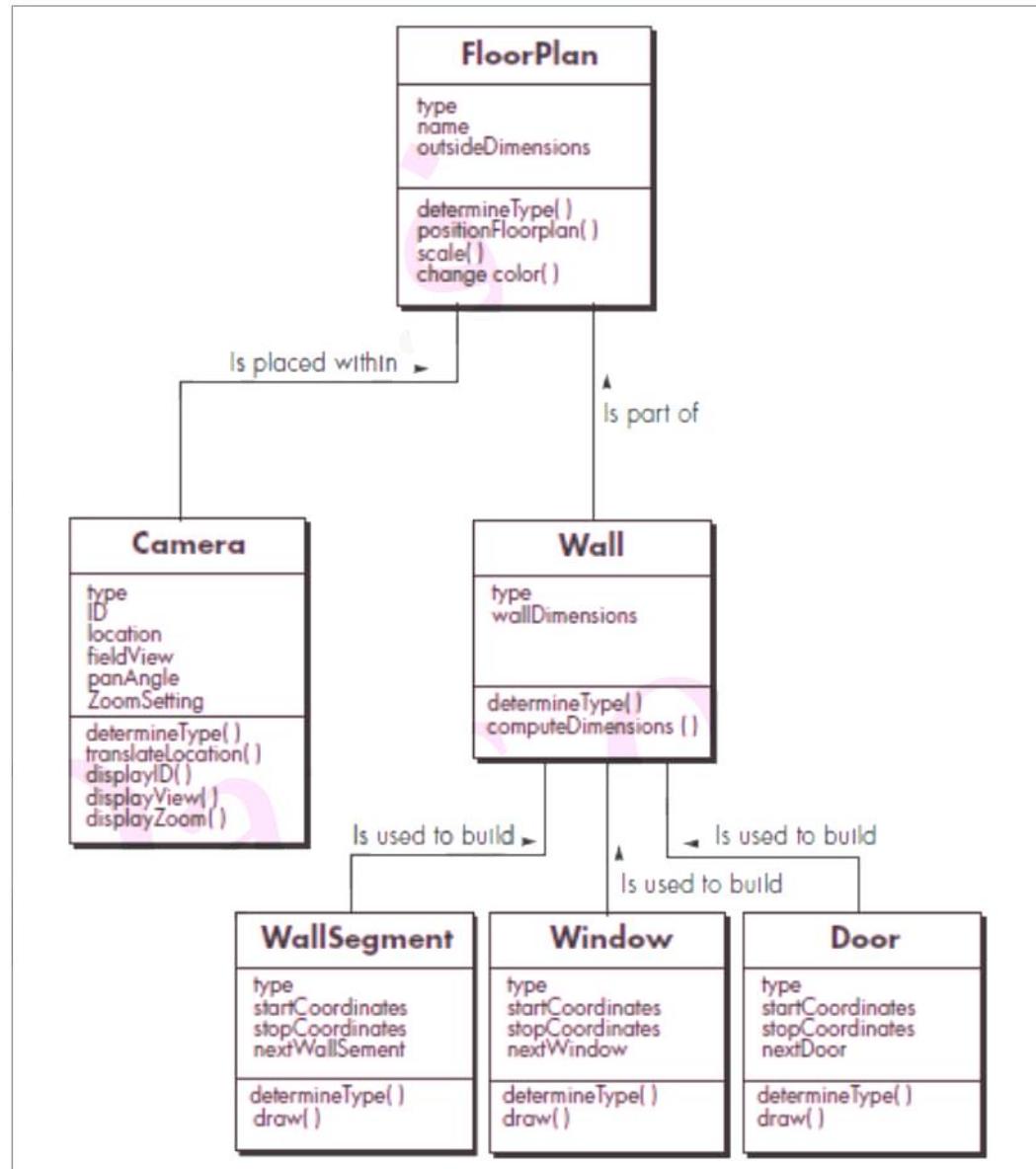
Class Models

The scene: Ed's cubicle, as requirements modeling begins.

The players: Jamie, Vinod, and Ed—all members of the *SafeHome* software engineering team.



موديل كلاس ها



مدل سازی همکار مسئولیت کلاس ها

Class-Responsibility-Collaborator (CRC) Modeling

مدل CRC، مجموعه ای از کارت های شاخص استاندارد است که کلاس ها را به نمایش می گذارند.

Class: FloorPlan	
Description	
Responsibility:	Collaborator:
Defines floor plan name/type	
Manages floor plan positioning	
Scales floor plan for display	
Scales floor plan for display	
Incorporates walls, doors, and windows	Wall
Shows position of video cameras	Camera

این کارت ها به سه بخش تقسیم می شوند:

- بالای کارت : نام کلاس

- سمت راست بدن کارت : فهرست مسئولیت های کلاس

- سمت چپ بدن کارت : همکاران

برای مدل CRC از کارت های واقعی یا مجازی استفاده می شود.

کلاس ها

طبقه بندی انواع کلاس ها ارائه شده در قبل را می توان با در نظر گرفتن گروه های زیر بسط داد :

- **کلاس های موجودیت** (مدل یا تجاری) : مستقیما از بیان مسئله استخراج می شوند.
- **کلاس های مرزی** : در ایجاد واسط به کار می روند که کاربر به هنگام استفاده از نرم افزار و تعامل مشاهده می کند.
- **کلاس های کنترل گر** : یک واحد کار را از ابتدا تا انتهای مدیریت می کند.

کلاس های کنترلر گر را می توان طوری طراحی کرد که موارد زیر را مدیریت کنند:

- ۱- ایجاد یا به هنگام سازی اشیای موجودیت
- ۲- معرفی اشیای مرزی به هنگام کسب اطلاعات از اشیای موجودیت
- ۳- ارتباطات پیچیده میان مجموعه های اشیا
- ۴- اعتبار سنجی داده های ارتباطی میان اشیا یا میان کاربر و برنامه

مسئولیت ها

پنج دستورالعمل برای تخصیص مسئولیت ها به کلاس ها :

- ۱- هوشمندی سیستم باید طوری میان کلاس ها توزیع شود که به بهترین وجه پاسخ گوی نیازهای مسئله باشد.
- ۲- هر مسئولیتی باید تا حد امکان به صورت کلی بیان شود.
- ۳- اطلاعات و رفتار مرتبط با آن باید در یک کلاس قرار داده شوند.
- ۴- اطلاعات مربوط به یک چیز باید تنها در یک کلاس قرار داده شوند و نباید در میان چند کلاس توزیع شوند.
- ۵- مسئولیت ها را در صورت امکان باید در میان کلاس های مرتبط به اشتراک گذاشت.

همکاری ها

کلاس ها به یکی از دو شیوه، مسئولیت های خود را به انجام می رسانند:

- ۱- کلاس می تواند از عملیات های خودش برای دستکاری صفات استفاده کند.
- ۲- کلاس می تواند با سایر کلاس ها همکاری کند.

تعریف:

همکاری ها نشان گر درخواست های یک کلاینت از سرور برای به انجام رساندن مسئولیت یک کلاینت هستند.
همکاری ، تجسم هم پیمانی کلاینت و سرور است.

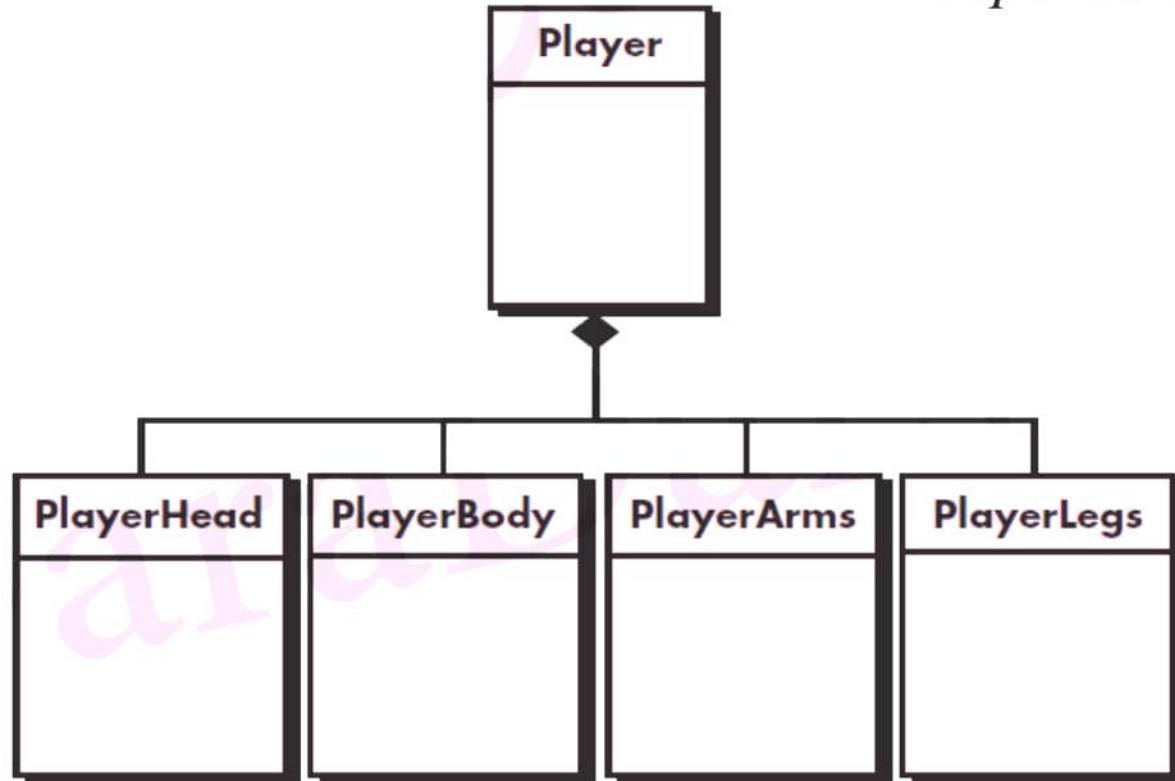
روابط میان کلاس ها

برای کمک به شناسایی همکارها سه رابطه کلی میان کلاس ها را بررسی می کنیم :

۱- رابطه شمول: *is-part-of*:

۲- رابطه اگاهی داشتن از : *has-knowledge-of* :

۳- بستگی داشتن به : *depends-upon* :



**CRC Models**

The scene: Ed's cubicle, as requirements modeling begins.

The players: Vinod and Ed—members of the SafeHome software engineering team.



Use case: *SafeHome* home management function.

Narrative:

We want to use the home management interface on a PC or an Internet connection to control electronic devices that have wireless interface controllers.

The system should allow me to turn specific lights on and off, to control appliances that are connected to a wireless interface, to set my heating and air conditioning system to temperatures that I define. To do this, I want to select the devices from a floor plan of the house. Each device must be identified on the floor plan. As an optional feature, I want to control all audiovisual devices—audio, television, DVD, digital recorders, and so forth.

With a single selection, I want to be able to set the entire house for various situations. One is *home*, another is *away*, a third is *overnight travel*, and a fourth is *extended travel*. All of these situations will have settings that will be applied to all devices. In the *overnight travel* and *extended travel* states, the system should turn lights on and off at random intervals (to make it look like someone is home) and control the heating and air conditioning system. I should be able to override these setting via the Internet with appropriate password protection . . .



Attributes:

optionsPanel
situationPanel
Floorplan
deviceIcons
devicePanels

Operations:

displayControl(),
selectControl(),
displaySituation(),
select situation(),
accessFloorplan(),
selectDeviceIcon(),
displayDevicePanel(),
accessDevicePanel(), ...

Class: HomeManagementInterface

Responsibility

displayControl()
selectControl()
displaySituation()
selectSituation()
accessFloorplan()

Collaborator

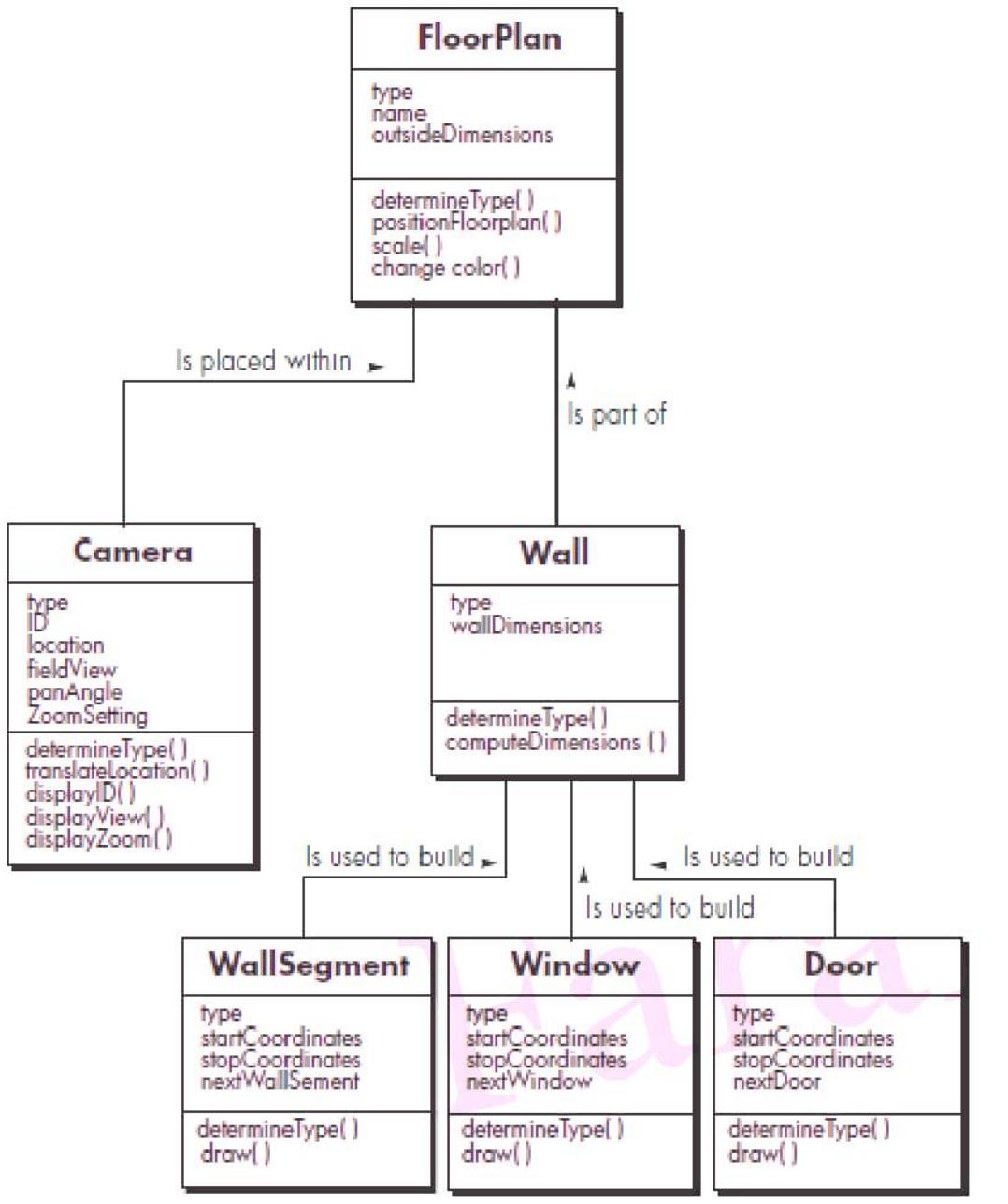
OptionsPanel (class)
OptionsPanel (class)
SituationPanel (class)
SituationPanel (class)
FloorPlan (class) . . .

اجتماع

Associations

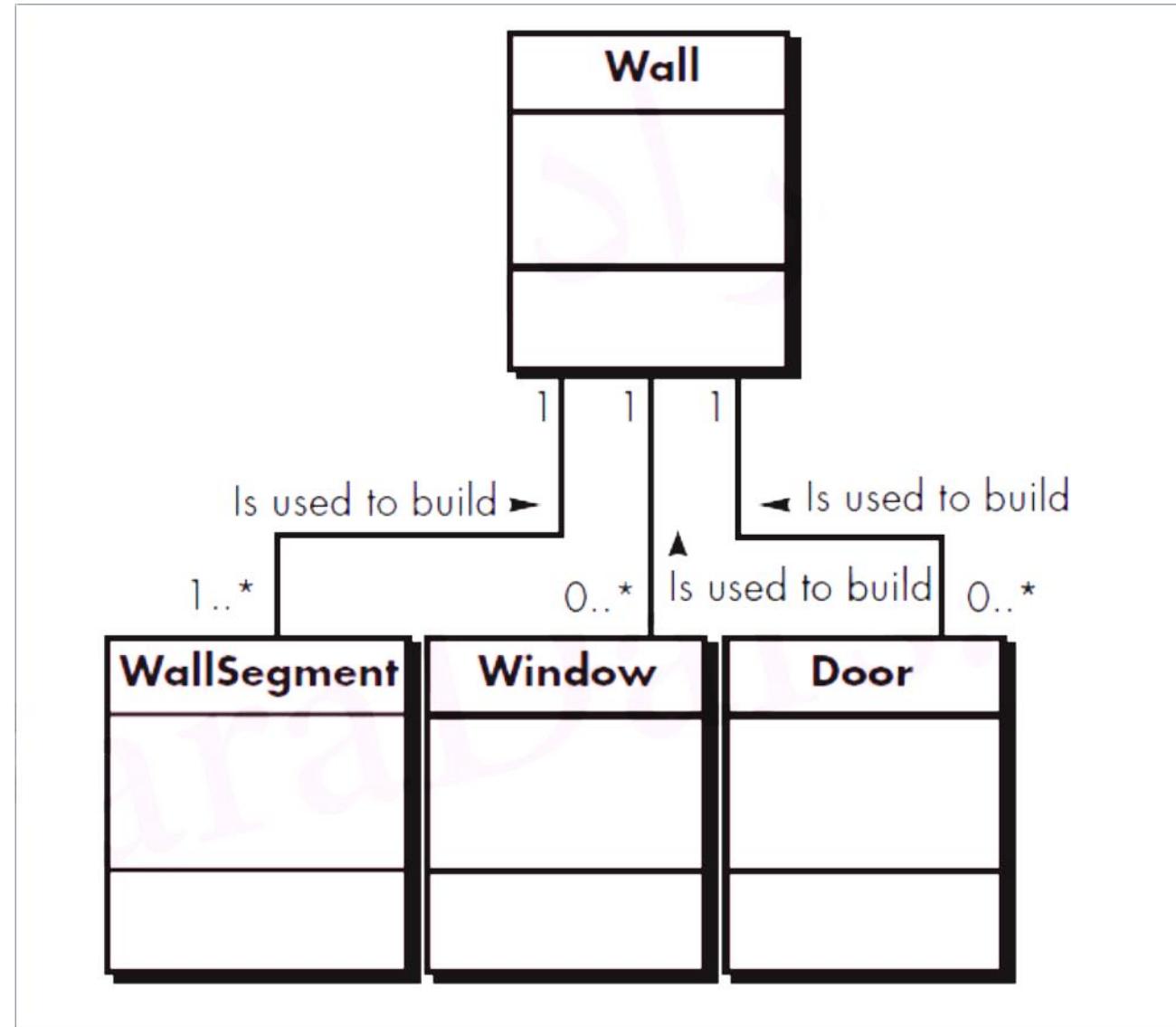
در بسیاری از موارد، دو کلاس تحلیل، به شیوه ای بسیار مشابه با ارتباط دوشیء داده ای، با هم ارتباط دارند.

در UML این روابط را اجتماع می نامند.



چند گانگی

Multiplicity



وابستگی

Dependencies

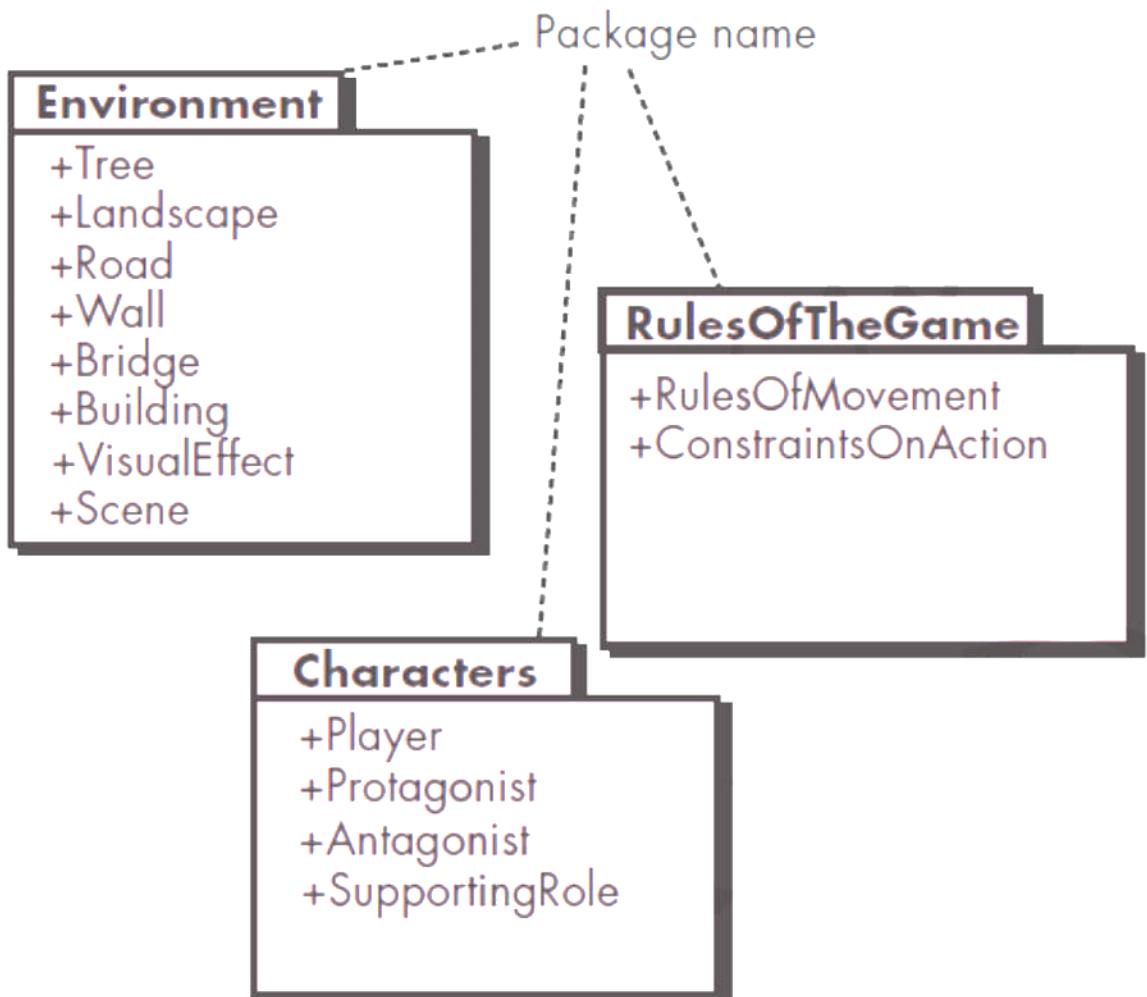
وابستگی ها توسط یک کلیشه (stereotype) تعریف می شوند.



`<<access>>` به این معنی است که استفاده از خروجی دوربین توسط یک کلمه عبور خاص کنترل می شود.

پکیج های تحلیل

Analysis Packages



بخش مهمی از مدل سازی تحلیل گروه بندی است.

یعنی عناصر گوناگون مدل تحلیل طوری گروه بندی می شوند که یک پکیج از آنها تشکیل شود و به آنها پکیج تحلیل گفته می شود. به هر پکیج یک نام داده می شود.

پکیج برای بسته بندی مجموعه ای از کلاس های مرتبط به کار می رود.

علامت مثبت، قبل از نام کلاس ها تحلیل در هر پکیج نشانگر ان است که این کلاس ها در معرض دید عموم هستند و بنابراین از طریق سایر پکیج ها قابل دسترسی اند.

علامت منفی، نشان می دهد که عنصر از تمام پکیج های دیگر پنهان است.

پایان فصل ۶