

عصبه فنی تله از انفجار است. عصاره کرم خنثی از انفجار است. *Umbrellachytry* است و
عصاره تله از انفجار است. عصاره کرم خنثی از انفجار است. عصاره کرم خنثی از انفجار است.

فصل هفتم

- الگوهای مدلسازی نیازمندیها - مهندسی نیازمندی برنامه های تحت وب

درس : طراحی نرم افزار پیشرفته
- مدرس : پروانه مرادی
<http://eng.uok.ac.ir/moradi>
- مدرس : باقر - خانم سمانه حسین مرادی - خانم فاطمه پور باقی

لیسه سال دوم ۹۴-۹۳
دانشکده ریاضی و علوم کامپیوتر
دانشگاه صنعتی امیرکبیر

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7th Edition (McGraw-Hill, 2008). Slides copyright 2009 by Rexy P. Dierckx.

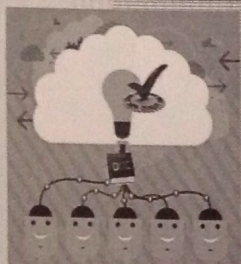
جلسه هفتم

- الگوهای مدل‌سازی نیازمندیها
- تحلیل نیازمندیهای برنامه های تحت وب

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill, 2009). Slides copyright 2009 by Robert Pressman.

کتابت صحیح و نورید اشترای کنیم!! میخوانیم بنیم در حق خودمان انانیم و انانیم؟ در حق خودمان

الگوهای مدلسازی نیازمندیها



- گواهی عدم ازاری: حکایتی های برای استفاده از دانش و همچنین نشان دانش این متخصص در زمینه های مختلف عدم ازاری است. این نشان به معنوی است که از پروژه های قبلی به پروژه های جدید قابل اعیال خواهد بود.
- نشان عدم ازاری: در یک دوره (مثلاً دو ساله) به معنوی است (نشان عدم ازاری) از همان دوره های جدید اعیال خواهد بود.
- یک اکتیو اعیال از همان دوره اعیال: اینکه یک اکتیو اعیال به معنوی است که برای یک معالیه یا یک متخلی حاصل ازیه نشود و این به معنوی است که استفاده از ازیانی میشود تا دیگران نتوانند از آن استفاده کنند.
- بعد از اینکه راه حل یک مشکل پیدا شد این راه حل برای استفاده به معنوی دیگران باید به معنوی یک مستند اعیال شود.
- این مستند همان ازیانی است.

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7th Edition* (McGraw-Hill 2006). Slides copyright 2006 by Roger Pressman.



شناسایی الگوهای مهندسی نیازمندیها

- نمودار مورد کاربرد یکی از عناصر پایه ای برای تشریح نیازمندی های سیستم است
- می توان برای مجموعه ای از نمودار های مورد کاربرد مرتبط با هم (یک مجموعه تعالیت مرتبط با هم) یک الگو طراحی کرد
- **semantic analysis pattern (SAP)**
- یک الگو است که برای مجموعه کوچکی از نمودار های مورد کاربرد که یک کاربرد عمومی و پایه ای از ترم افزار را در بر میگیرد

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7/e* (McGraw-Hill 2008). Slides copyright 2008 by Roger Pressman.

ادامه مثال:

- بر اساس این مورد کاربرد می توان یک الگوی کلی ایجاد کرد که این الگو در عین سادگی می تواند در کاربردهای مختلفی که یک سخت افزار اطلاعات ورودی را از سنسور میگیرد و یک سری اعمال را انجام میدهد به کار گرفته شود (یک الگوی SAP)
- در این الگو سخت افزار اطلاعات ورودی خود را از سنسورها (مثل دوربین) میگیرد و یک نرم افزار کنترلی (مثل ای سی یو ماشین) اطلاعات را پردازش کرده و بر اساس این اطلاعات یک سری تصمیمات گرفته می شود (مثلا متوقف کردن ماشین در صورت نزدیک شدن به مانع) و این تصمیم را به عملگرها (مثل خلاص کردن دنده) میدهد.
- این الگوی کلی با نام Actuator-Sensor شناخته میشود.

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill 2008). Slides copyright 2008 by Roger Pressman.

مثال:

- مورد کاربرد زیر که برای کنترل سنسورهای یک وسیله نقلیه به کار می رود را در نظر بگیرید:
- مورد کاربرد:
- ماشین کردن ماشین در هنگام حرکت به صورت دنده عقب
- توضیحات:
- وقتی که ماشین در دنده عقب قرار داده می شود و به عقب حرکت می کند نرم افزار کنترلی دارای سی یو ماشین دوربین عکس ماشین روشن شده و تصاویر را در مانیتور جلو ماشین نمایش میدهد نرم افزار کنترلی تصاویر داخل تصویر را تشخیص داده و دور آنها را با یک خط مشخص میکند و همچنین فاصله آنها را با ماشین مشخص میکند و در صورت لزوم وقتی که فاصله تصاویر در حدود چند متری ماشین باشند آژیر و اختلالاتی را از راه راننده میدهد در ضمن ممکن است که اگر ماشین به مانع خیلی نزدیک شود به صورت آهسته آهسته ماشین را متوقف میکند.

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill 2008). Slides copyright 2008 by Roger Pressman.

بررسی الگوی Actuator-Sensor ادامه:

نام الگو: Actuator-Sensor

محدودیتها (constraints):

- هر سنسور باید روشهایی برای خواندن اطلاعات سنسور و بازتابی آن داشته باشد
- هر سنسور (اتو) (عملگر) باید قابلیتایی برای انتقال پیامهای کنترلی را به عملگرها داشته باشد
- هر سنسور باید قابلیت این را داشته باشد تا از صحت عملکرد عملگرها مطمئن شود برای این کار باید بتواند پیامهایی را در فواصل زمانی منظم فرستاده تا در صورت دریافت پاسخ از عملگر از صحت عملکرد آن مطمئن شود
- هر عملگر باید قابلیت انجام پیامهای کنترلی مرکز کنترل را داشته باشد
- هر سنسور و عملگر باید قابلیت این را داشته باشد تا بتواند وضعیت عملکرد خود را بررسی نموده و نتیجه را برای مرکز کنترل ارسال نماید

کاربرد:

راه حل ارائه شده برای این الگو برای هر سیستمی که دارای چندین سنسور و چندین عملگر باشد مناسب است

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill 2008). Slides copyright 2008 by Roger Pressman.

بررسی الگوی Actuator-Sensor

نام الگو: Actuator-Sensor

هدف (intent):

برای ارائه در کلیه سیستم های سخت افزاری و تعبیه شده که اطلاعات را از ورودی خوانده و تصمیمات را از طریق عملگرها اعمال نمایند.

انگیزه (motivation):

در سیستم های تعبیه شده (همچون نرم افزار ای سی یو ماشین و یا نرم افزارهای کنترلی میکروکنترلرها و یا هوابتها و یا دستگاههای صنعتی) توانمندی کماکان از سنسورها وجود دارد که اطلاعات معتبره را برای سیستم کنترل مرکزی فراهم می کنند.

این سنسورها ممکن است مستقیم و یا غیر مستقیم اطلاعات را به مرکز کنترل ارسال نمایند در ضمن تصمیمات مرکز کنترل از طریق عملگرها اعمال می شود.

ممکن است در سخت افزارهای معتبره سنسورها و عملگرهای معتبره وجود داشته باشد ولی رفتار آنها مشابه است. لذا میتوان یک الگو برای آنها در نظر گرفت.

این الگو شخصی می کند که چگونه سنسورها و عملگرها را با سیستم به کار گرفته نحوه عمل و رفتار این سنسورها و عملگرها چگونه باشد در این الگو در این الگو مکانیزم pull برای سنسورهای پسیو (PassiveSensors) و مکانیزم push برای سنسورهای فعال (ActiveSensors) مورد استفاده قرار می گیرد.

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill 2008). Slides copyright 2008 by Roger Pressman.

ویژگیهای نرم افزارهای تحت وب می تواند در ۳

ویژگیهای نرم افزارهای مبتنی بر وب

Network intensiveness.

- نرم افزارهای تحت وب عموماً بر روی شبکه هستند.

Concurrency (همزمانی).

- نرم افزارهای تحت وب توسط کاربران زیادی به صورت همزمان مورد استفاده قرار می گیرند.

Unpredictable load.

- تعداد کاربران استفاده کننده از یک نرم افزار تحت وب قابل پیش بینی نیست و ممکن است تعداد این کاربران در یک روز با تعداد کاربران یک برنامه در یک روز دیگر متفاوت باشد.

Performance.

- نرم افزار تحت وب باید کارایی زیادی داشته باشد تا کاربران به صورت آنلاین (یا تاخیر کم) بتوانند از آن استفاده کنند.

Availability.

- نرم افزار تحت وب باید ۲۴ ساعت در روز، هفت روز در هفته و ۳۶۵ روز در سال در دسترس باشد.

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill 2006). Slides copyright 2006 by Roger Pressman.

28

ویژگیهای نرم افزارهای مبتنی بر وب - ادامه

Data driven.

- نرم افزارهای تحت وب با انواع داده ای مختلف مثل صوت، تصویر، متن، فیلم و ... کار میکنند.

Content sensitive.

- کیفیت و ظاهر داده های نرم افزارهای تحت وب باید مناسب باشد.

Continuous evolution.

- نرم افزارهای معمولی به صورت دوره ای بروز رسانی می شوند ولی نرم افزارهای تحت وب نیاز است که به صورت پیوسته و مداوم تحت بروز رسانی و نگهداری قرار بگیرند.

Immediacy.

- نرم افزارهای تحت وب به دلیل نیاز بازار باید در زمان بسیار کمتری نسبت به نرم افزارهای دیگر نوشته شوند.

Security.

- نرم افزارهای تحت وب در شبکه در یک محیط عمومی مورد دسترسی قرار می گیرند لذا لازم است که امنیت بالایی داشته باشند.

Aesthetics.

- ظاهر برنامه های تحت وب باید مناسب و کاربر پسند باشد.

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill 2006). Slides copyright 2006 by Roger Pressman.

29

تحلیل برنامه های تحت وب

Content Analysis (تحلیل محتوی).

- باید تمام اطلاعاتی را که لازم است در صفحات وب نمایش داده شود را مشخص کرد این اطلاعات میتواند شامل متن، فیلم، عکس و یا صدا باشد. لذا نیاز به مدخل سازی داده و مشخص نمودن اشیاء داده ای وجود دارد.

Interaction Analysis (تحلیل تعاملات با کاربر).

- روش تعامل کاربر با صفحات وب باید کاملاً مشخص شود برای این کار میتوان از نمودار مورد کاربرد استفاده کرد.

Functional Analysis (تحلیل عملکرد).

- تمام عملیاتی که یک وب سایت، نحوه درخواستهای کاربر و چگونگی رفتار وب سایت در برخورد با رفتارهای کاربر باید تحلیل شود. لذا برای تحلیل رفتار یک وب سایت میتوان از نمودار مورد کاربرد استفاده کرد.

Configuration Analysis (تحلیل پیکربندی).

- محیط و زیر ساختی که یک برنامه تحت وب قرار است بر روی آن اجرا شود باید با جزئیات کامل تحلیل شود (سرور مورد استفاده، بهای باید در اختیار و ...)

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill 2006). Slides copyright 2006 by Roger Pressman.

31

در چه زمانی باید تحلیل را باید انجام داد؟

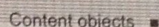
- در بسیاری از موارد که وب سایت طراحی می کنیم بخش طراحی و تحلیل با هم ادغام می شوند اما در شرایط زیر بخش تحلیل برنامه کاربردی تحت وب به صورت جداگانه انجام می پذیرد:

- برنامه کاربردی بزرگ و یا پیچیده باشد.
- تعداد دستمیل و استفاده کنندگان از این برنامه خیلی زیاد باشد.
- برنامه کاربردی تحت وب یکی از برنامه های اساسی سازمان باشد.
- موفقیت سازمان در گرو موفقیت برنامه تحت وب باشد.
- مثل فروشگاههای تجارت الکترونیک و ...

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill 2006). Slides copyright 2006 by Roger Pressman.

32

مدل سازی محتوای برنامه تحت وب



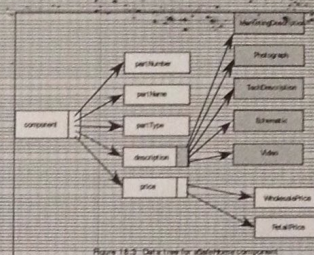
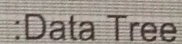
- Attributes ■

- ارتباط بین اشیاء داده ای باید مشخص شود.
- بهتر است ساختار داده ای به صورت سلسله مراتبی نشان داده شود.
- برای اینکه می توان از نمودار ERD و یا Data Tree استفاده کرد.

Entity - ¹⁰موجودات

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7th* (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

Entity

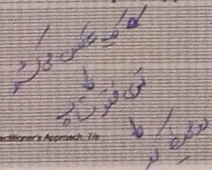
Figure 18.2 Data tree for *staphylococcus* comparison

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

100

■ مدل تعاملی برای برنامه های کاربردی تحت وب شامل چهار مورد زیر است:

- use-cases ■
- sequence diagrams ■
- state diagrams ■
- a user interface prototype ■



These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th Edition (McGraw-Hill, 2006). Slides copyright 2006 by Roger Pressman.

24

```

sequenceDiagram
    participant User
    participant Room
    participant RoomType
    participant ProductComponent
    participant GlobalAttributes
    participant RoomPlanRepository
    participant GlobalRepository

    User->>Room: createRoom()
    activate Room
    Room->>RoomType: getRoomType()
    activate RoomType
    RoomType->>Room: getRoomType()
    deactivate RoomType
    Room->>RoomPlanRepository: getRoomPlan()
    activate RoomPlanRepository
    RoomPlanRepository->>Room: getRoomPlan()
    deactivate RoomPlanRepository
    Room->>GlobalAttributes: getRoomAttributes()
    activate GlobalAttributes
    GlobalAttributes->>Room: getRoomAttributes()
    deactivate GlobalAttributes
    Room->>GlobalRepository: getRoomGlobalAttributes()
    activate GlobalRepository
    GlobalRepository->>Room: getRoomGlobalAttributes()
    deactivate GlobalRepository
    Room->>RoomPlanRepository: getRoomPlan()
    activate RoomPlanRepository
    RoomPlanRepository->>Room: getRoomPlan()
    deactivate RoomPlanRepository
    Room->>GlobalAttributes: getRoomAttributes()
    activate GlobalAttributes
    GlobalAttributes->>Room: getRoomAttributes()
    deactivate GlobalAttributes
    Room->>GlobalRepository: getRoomGlobalAttributes()
    activate GlobalRepository
    GlobalRepository->>Room: getRoomGlobalAttributes()
    deactivate GlobalRepository
    Room->>RoomPlanRepository: getRoomPlan()
    activate RoomPlanRepository
    RoomPlanRepository->>Room: getRoomPlan()
    deactivate RoomPlanRepository
    Room->>GlobalAttributes: getRoomAttributes()
    activate GlobalAttributes
    GlobalAttributes->>Room: getRoomAttributes()
    deactivate GlobalAttributes
    Room->>GlobalRepository: getRoomGlobalAttributes()
    activate GlobalRepository
    GlobalRepository->>Room: getRoomGlobalAttributes()
    deactivate GlobalRepository
    deactivate Room
  
```

UML sequence diagram illustrating the interactions between a User and various system components (Room, RoomType, Product Component, Global Attributes, RoomPlan Repository, Global Repository).

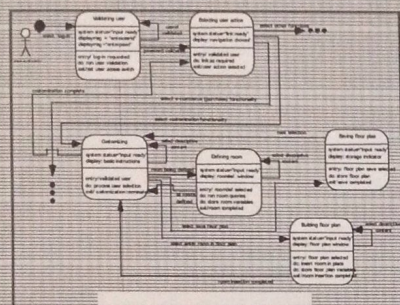
The sequence of messages is as follows:

- User to Room: createRoom()
- Room to RoomType: getRoomType()
- RoomType to Room: getRoomType()
- Room to RoomPlan Repository: getRoomPlan()
- RoomPlan Repository to Room: getRoomPlan()
- Room to Global Attributes: getRoomAttributes()
- Global Attributes to Room: getRoomAttributes()
- Room to Global Repository: getRoomGlobalAttributes()
- Global Repository to Room: getRoomGlobalAttributes()
- Room to RoomPlan Repository: getRoomPlan()
- RoomPlan Repository to Room: getRoomPlan()
- Room to Global Attributes: getRoomAttributes()
- Global Attributes to Room: getRoomAttributes()
- Room to Global Repository: getRoomGlobalAttributes()
- Global Repository to Room: getRoomGlobalAttributes()
- Room to RoomPlan Repository: getRoomPlan()
- RoomPlan Repository to Room: getRoomPlan()
- Room to Global Attributes: getRoomAttributes()
- Global Attributes to Room: getRoomAttributes()
- Room to Global Repository: getRoomGlobalAttributes()
- Global Repository to Room: getRoomGlobalAttributes()

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7th (McGraw-Hill 2004). Slides copyright 2009 by Roger Pressman.

10

عاشقوں کی مختلف نسلوں کا شمار



These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

37

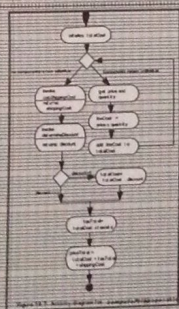
- مدل نامی برای یک برنامه کاربردی تحت وب شامل دو مورد زیر است:
 - فعالیت‌های که توسط کاربر قابل مشاهده است.
 - فعالیت‌های که در کلاس‌های سیستم پیاده سازی شده است.
- از نمودار فعالیت (activity diagram) برای بازتابی پردازش‌های سیستم استفاده می‌شود.

۲
کتابخانه
موزه و مرکز اسناد
سازمان اسناد و کتابخانه ملی
جمهوری اسلامی ایران

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill 2009). Slides copyright 2009 by Roger Freeman.

1

Activity Diagram (نمودار فعالیت)

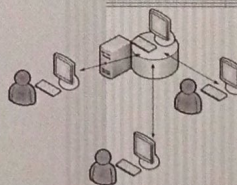


These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

30

مدل پیکربندی: Configuration

- **Server Side** - یک پیکربندی سیستم سرور
 - مشخص نمودن ساختار سخت افزاری سرورها
 - مشخص نمودن سیستم عامل های سرورها
 - قابلیت های که سرور می تواند برای برنامه داشته باشد
 - واسطه ها، پرتگاه های ارتباطی و موارد دیگر در این رابطه باید به صورت دقیق مشخص شود
- **Client Side** - یک پیکربندی سیستم کلاینت
 - اطلاعات یک پیکربندی مربوط به مرورگرها باید مشخص شود
 - ابزارهای نصب سیستم های مشخص شود

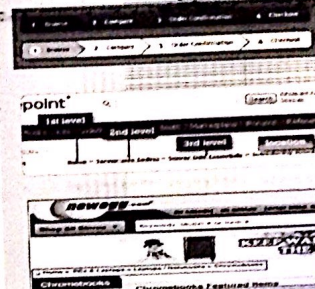


These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7th* (McGraw-Hill 2009). Slides copyright 2009 by Roger Prattman.

10

[illegible]

44

[illegible]

42

Team Foundation

مکتب و مدرسه اینست که در این مکتب و مدرسه و مدارس که تفاوتی را می بینند مدیریت نیز را می دانند

و این مکتب و مدرسه نیز دارای اهداف و مصادیق حقوقی و قراردادها می باشد

نمونه های مدرسه و مدارس را می توان به

✓ مدیریت بر مبنای برنامه های آموزشی

✓ تصمیم بر مبنای برنامه های آموزشی

✓ مدیریت بر مبنای

✓ تصمیم بر مبنای