

# طراحی سیستم‌های تعبیه شده Embedded System Design

فصل دوم - قسمت چهارم

## مشخص سازی Specifications

کاظم فولادی  
دانشکده‌ی مهندسی برق و کامپیوتر  
دانشگاه تهران

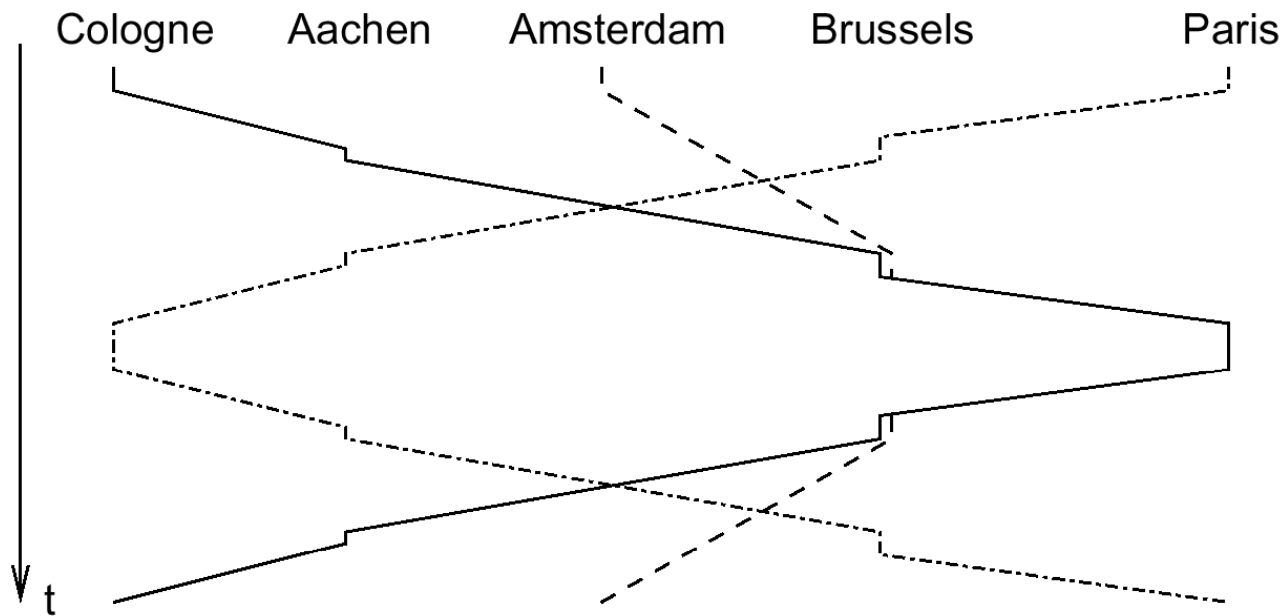
kazim@fouladi.ir



# نمودارهای دنباله‌ی پیام

## Message sequence charts (MSC)

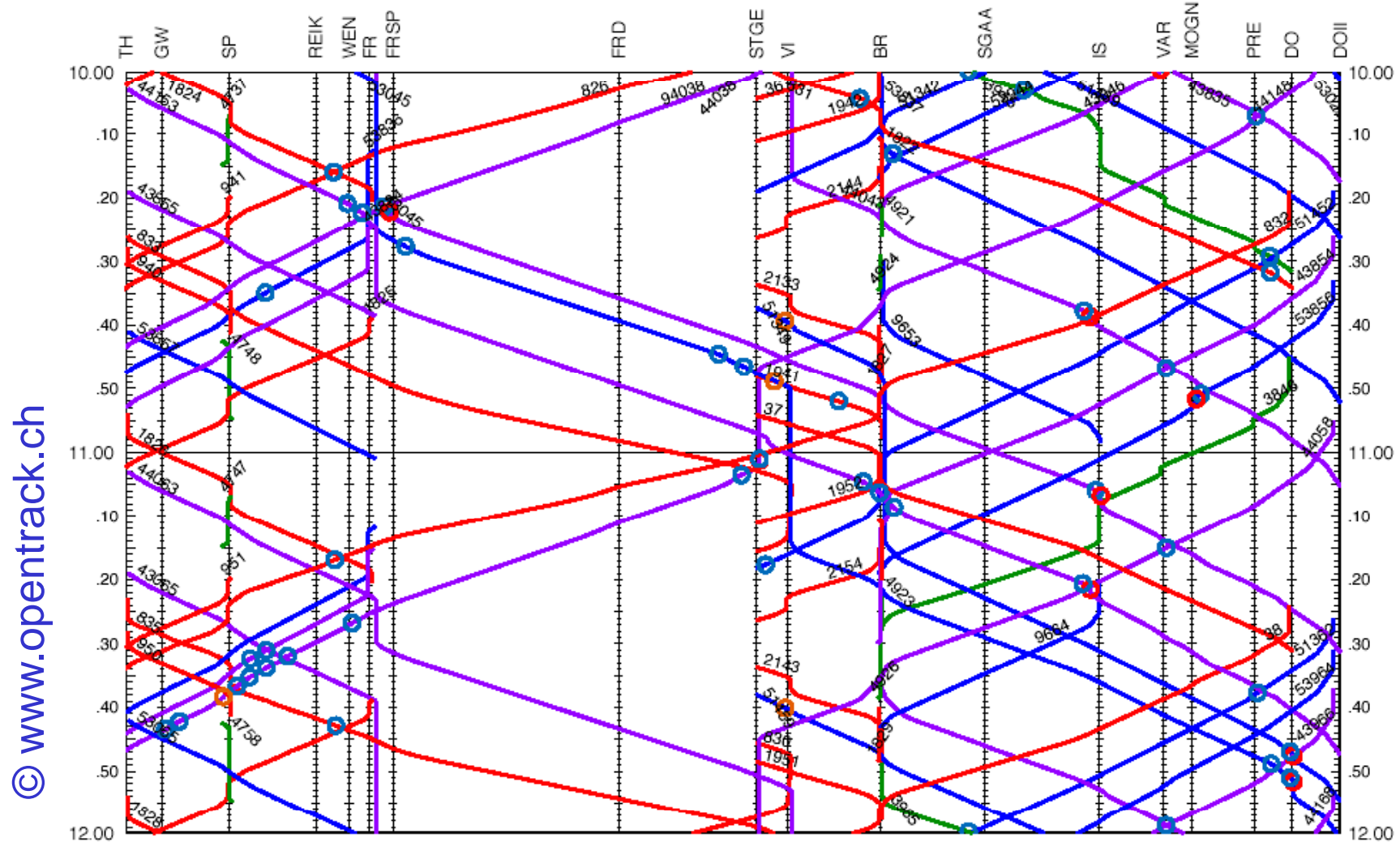
ابزاری گرافیکی برای نمایش دادن زمان‌بندی‌ها؛  
 زمان به صورت عمودی، توزیع جغرافیایی به صورت افقی



عدم تمایز بین همپوشانی اتفاقی و همگام‌سازی



# دیاگرام‌های زمان/مکان به عنوان یک حالت خاص

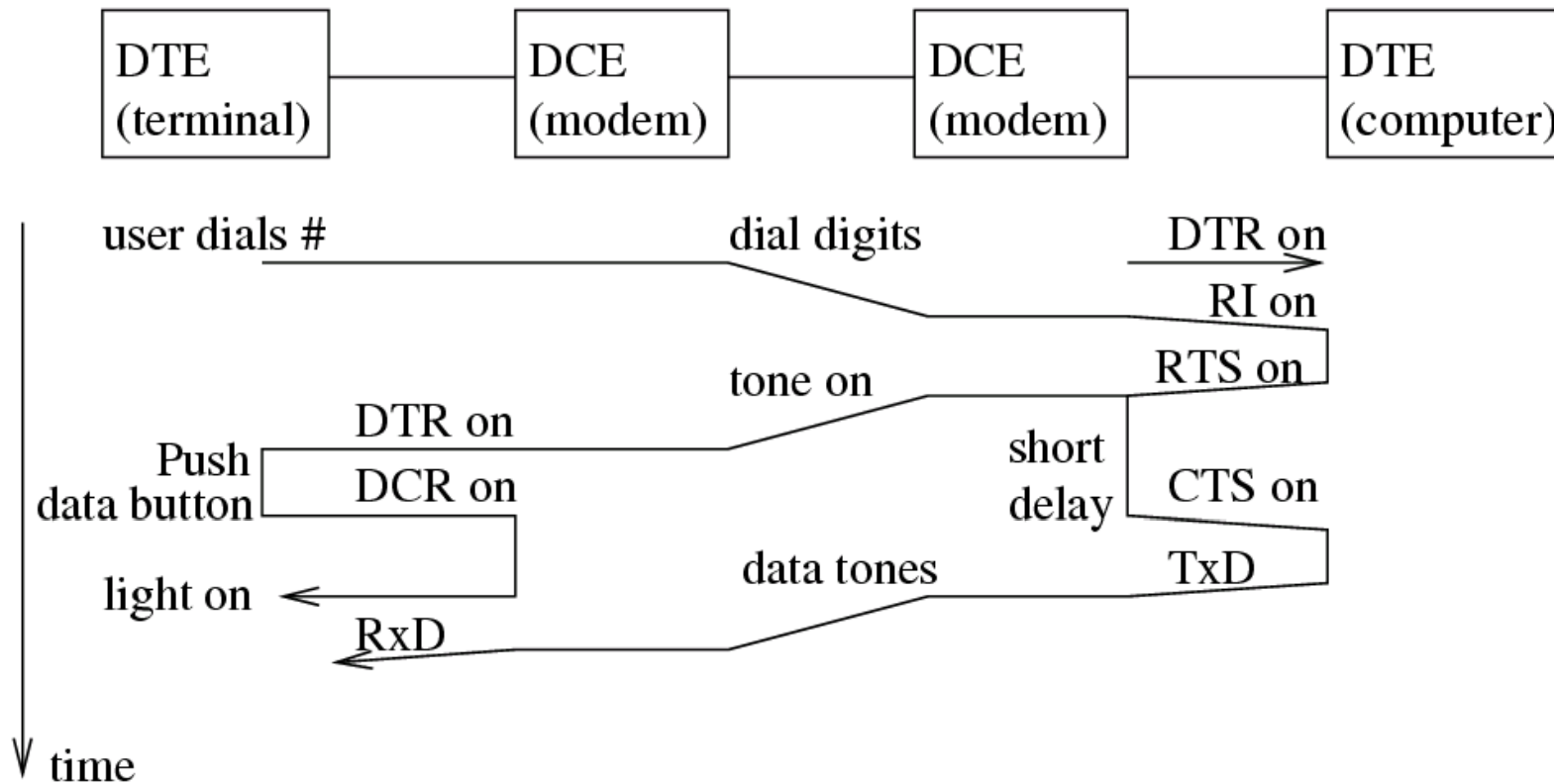


© www.opentrack.ch



# Example 1: establishing an (old-fashioned) modem connection

## مثال ۱: برقراری یک اتصال با مودم

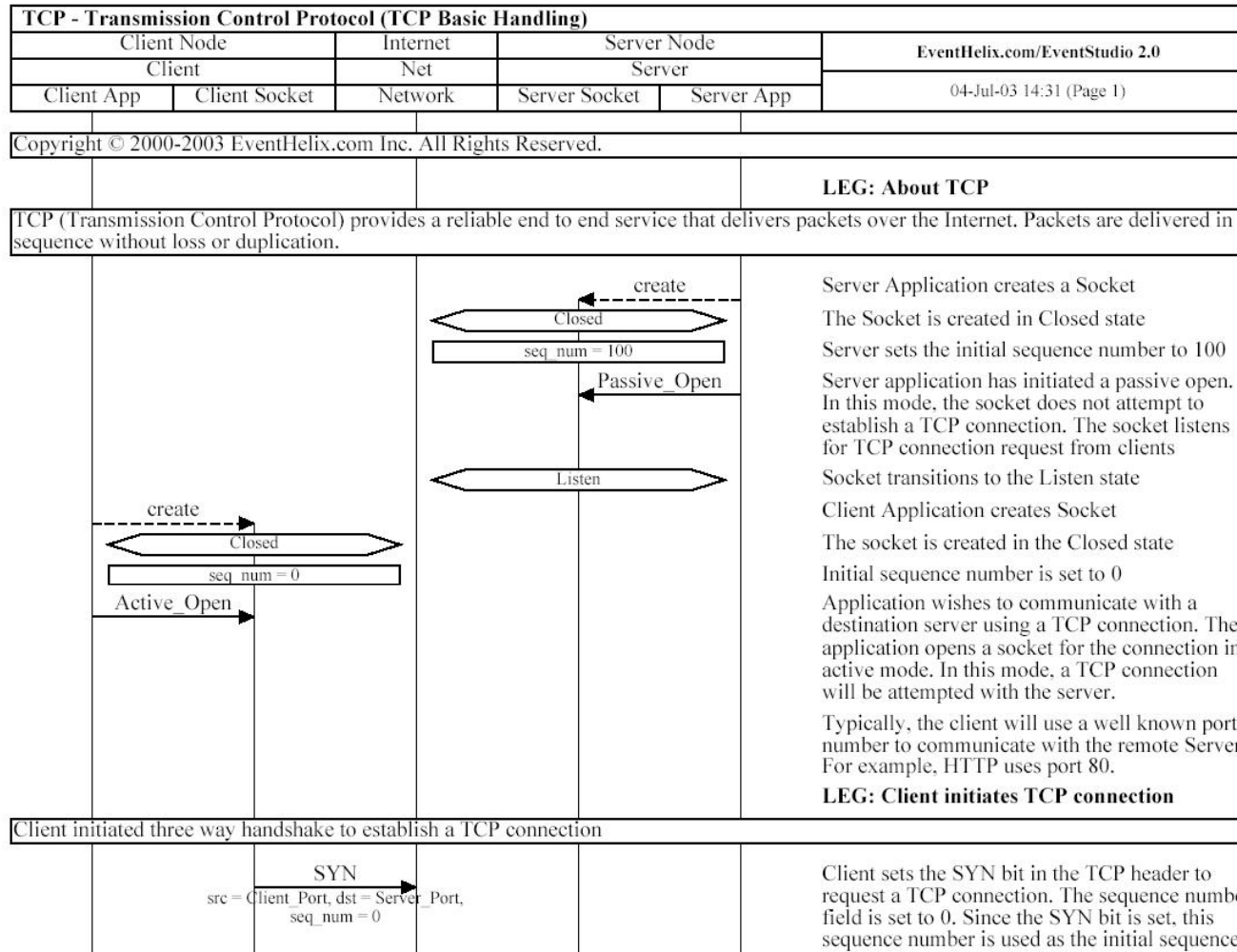


According to Stallings



## Example 2: setting up a TCP connection

### مثال ۲: تنظیم یک اتصال TCP



© Eventhelix.com



## نمودارهای دنباله‌ی پیام

### مزایا:

- مناسب برای به تصویر کشیدن زمان‌بندی‌ها
- روش پذیرفته شده برای بازنمایی زمان‌بندی‌ها در حمل و نقل
- استانداردها:

ITU-TS Recommendation Z.120: Message Sequence Chart (MSC), ITU-TS, Geneva, '96.

Semantics also defined:

*ITU-TS Recommendation Z.120: Message Sequence Chart (MSC) —*

*Annex B: Algebraic Semantics of Message Sequence Charts*, ITU-TS, Geneva.

### معایب:

- تنها یک مورد را تشریح می‌کند، بدون در نظر گرفتن تلورانس زمانی:  
«معنی یک مشخص‌سازی با MSC چیست: آیا همه‌ی رفتارهای یک سیستم را تشریح می‌کند، یا مجموعه‌ای از رفتارهای نمونه‌ی یک سیستم را؟»\*

\* H. Ben-Abdallah and S. Leue, "Timing constraints in message sequence chart specifications," in *Proc. 10th International Conference on Formal Description Techniques FORTE/PSTV'97*, Chapman and Hall, 1997.



*More  
in-depth:*

## Life Sequence Charts\* (LSCs)

Key problems observed with standard MSCs:

During the design process, MSC are initially interpreted as

**“what could happen”**

(existential interpretation, still allowing other behaviors).

Later, they are frequently assumed to describe

**“what must happen”**

(referring to what happens in the implementation).

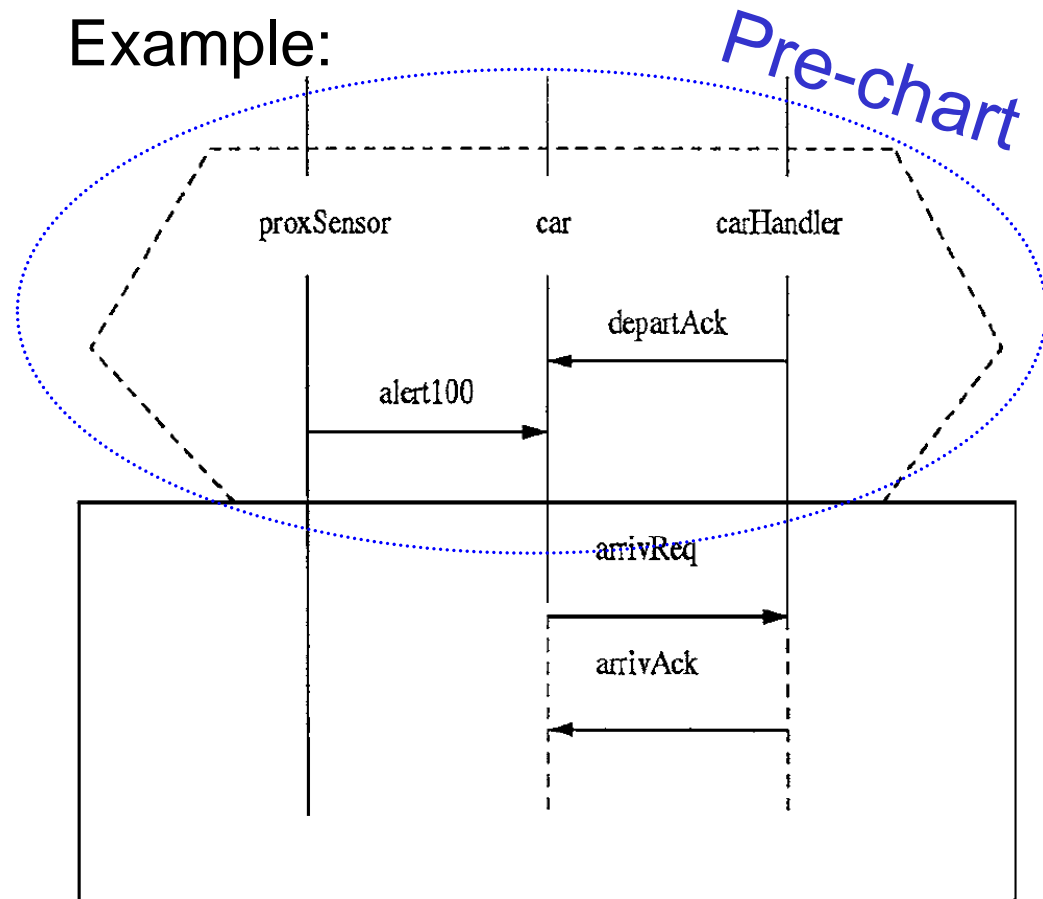
\* W. Damm, D. Harel: LSCs: Breathing Life into Message Sequence Charts,  
*Formal Methods in System Design*, 19, 45–80, 2001



# Extensions for LSCs (1)

Extension 1:  
 Introduction of **pre-charts**:  
 Pre-charts describe conditions that must hold for the main chart to apply.

Example:





## Extensions (2)

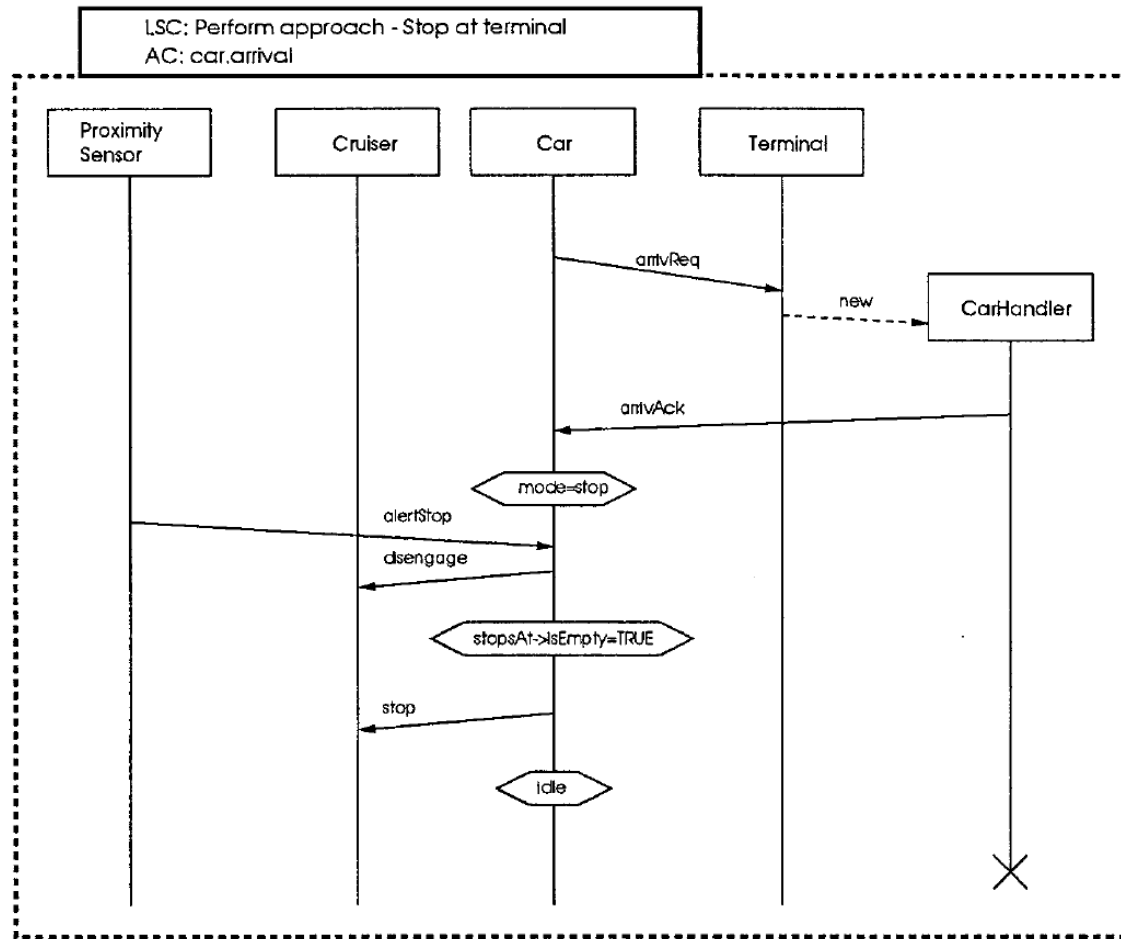
### Extension 2: Mandatory vs. provisional behavior

Level	Mandatory (solid lines)	Provisional (dashed lines)
<b>Chart</b>	All runs of the system satisfy the chart	At least one run of the system satisfies the chart
<b>Location</b>	Instance must move beyond location/time	Instance run need not move beyond loc/time
<b>Message</b>	If message is sent, it will be received	Receipt of message is not guaranteed
<b>Condition</b>	Condition must be met; otherwise abort	If condition is not met, exit subchart

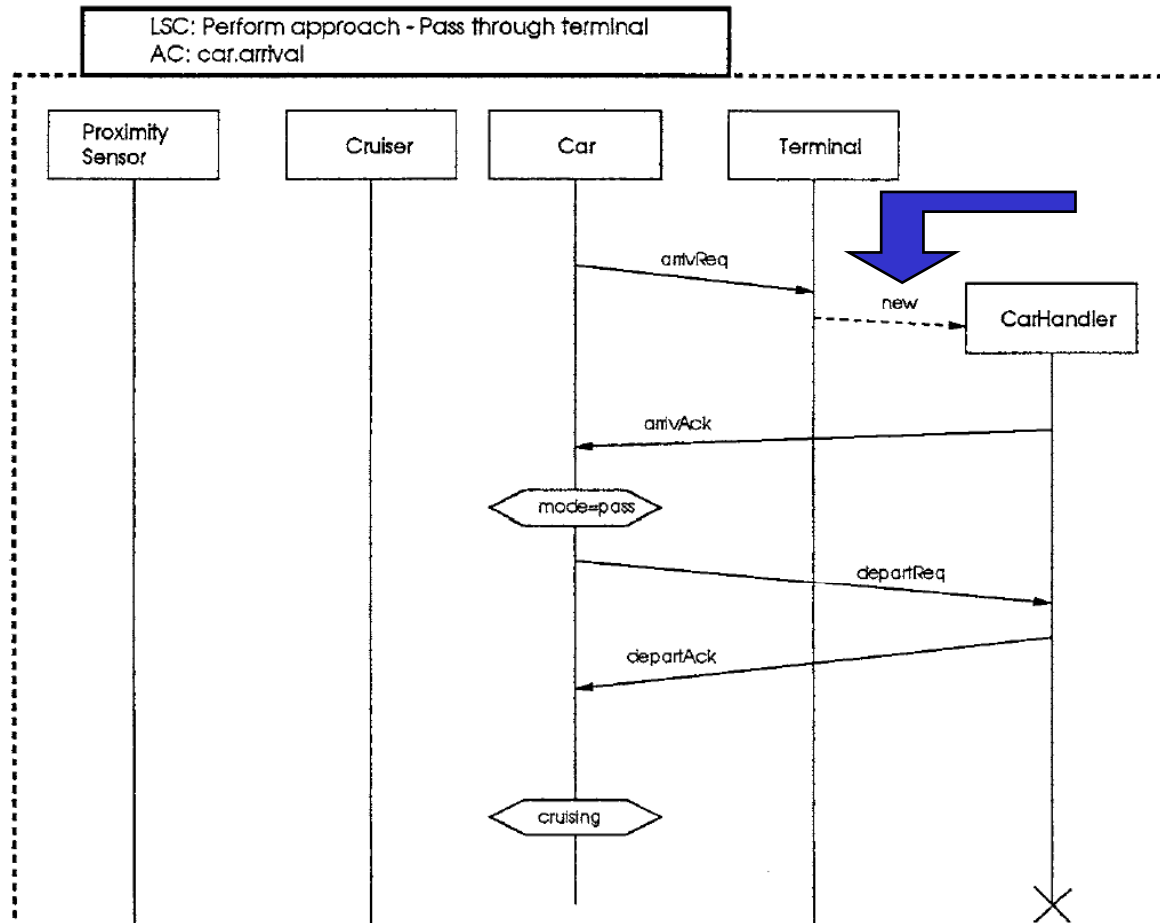


# Provisional charts: Behavior may be this one:

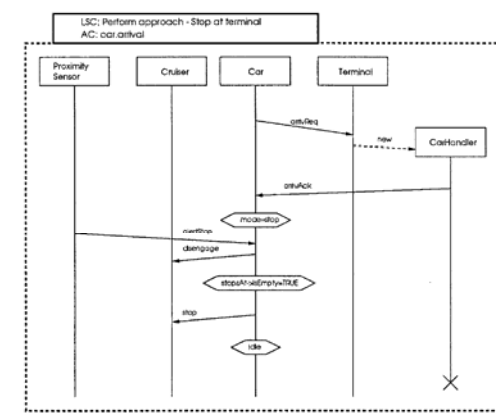
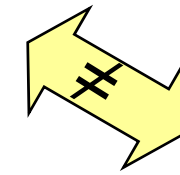
Dashed charts



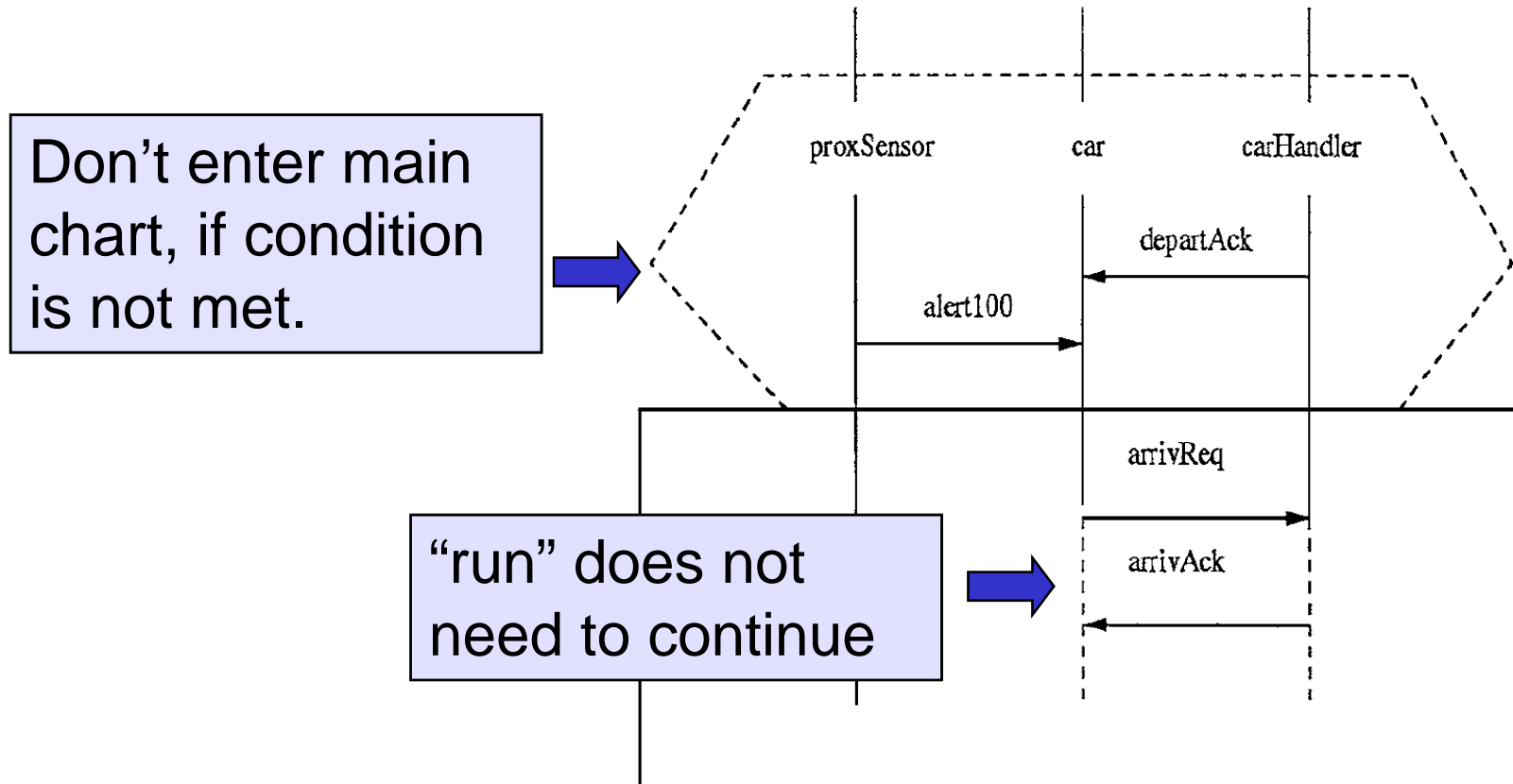
# ... or this one:



Message does not need to arrive



# Extensions on other levels

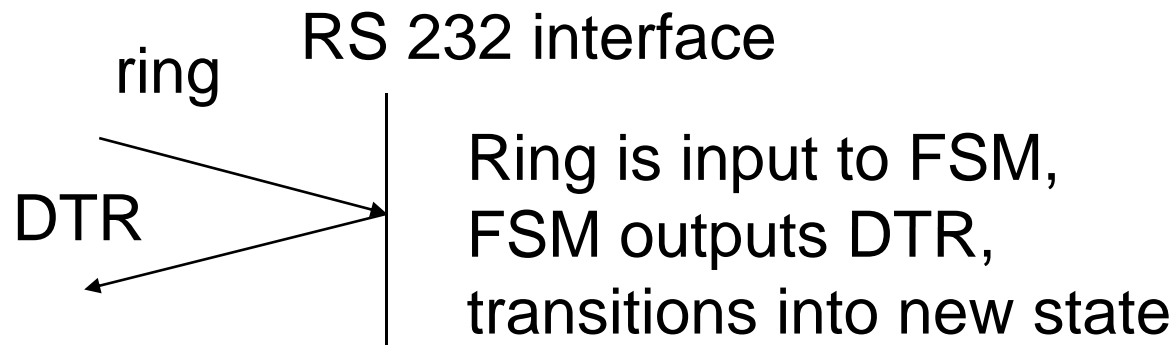




## Exploiting distinction in tools

Mandatory behavior can be used to generate StateCharts models from LSC models.

- ➔ LSCs enable link between “timing” spec and FSMs (no real link possible with standard MSCs).



No precise timing!

Provisional behavior can be checked against mandatory. Error messages can be generated if the two are not consistent.

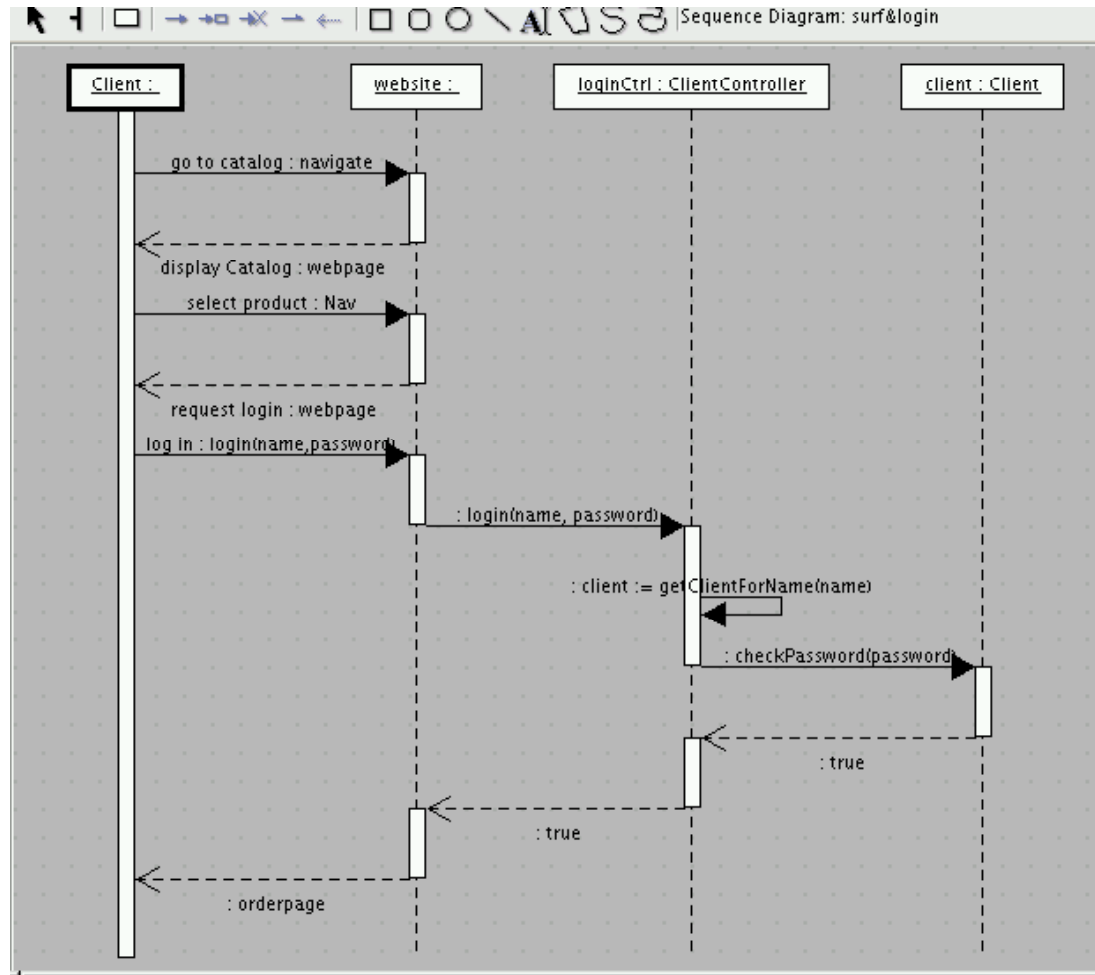


# کاربرد در UML

کاربردهای فراوان  
 MSCها در UML.  
 (با عنوان دیاگرام  
 دنباله شناخته  
 می شود)

عدم وجود  
 زمانهای دقیق

انواع بسیاری از  
 عناصر اضافی

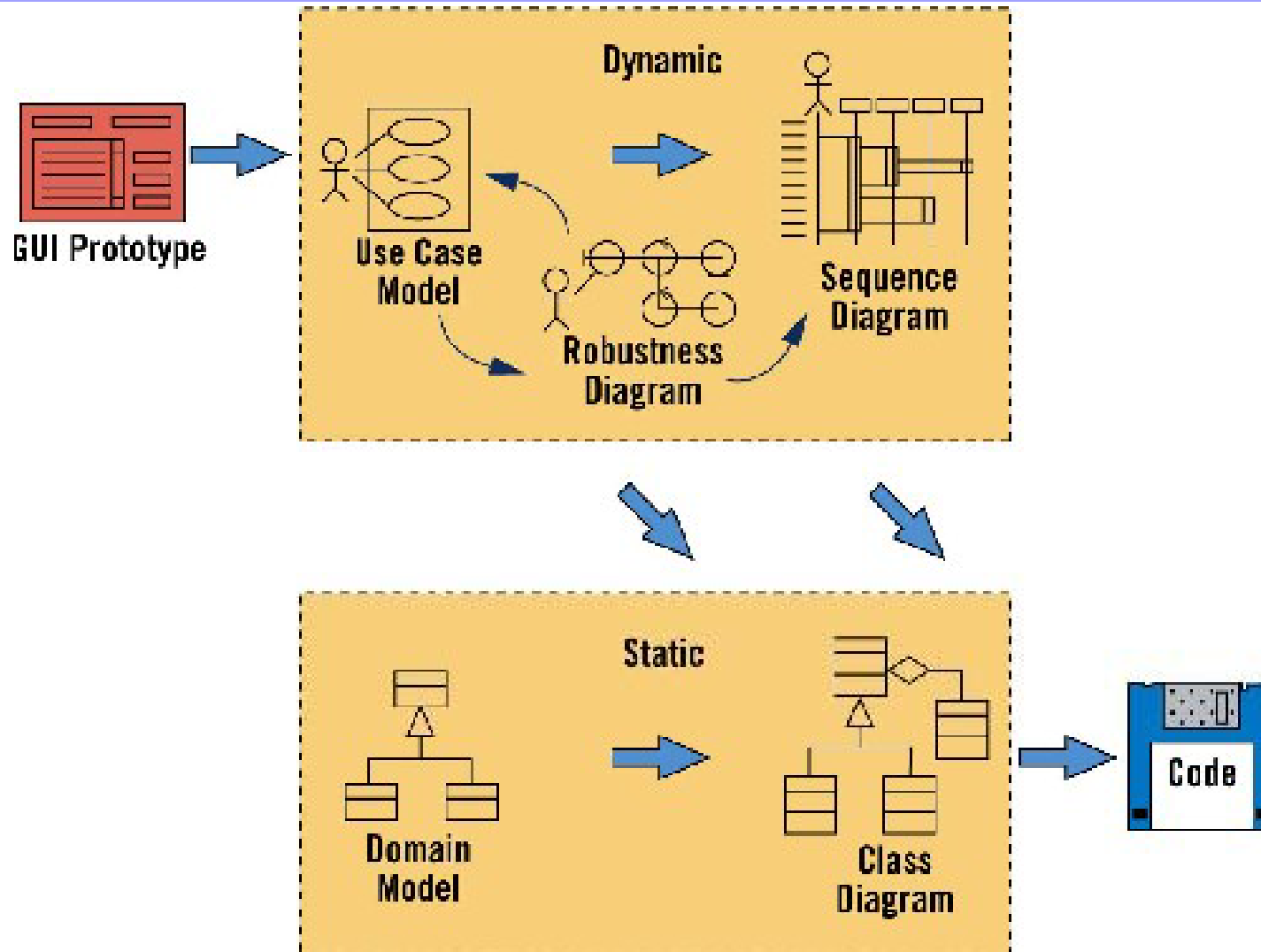


from: <http://www.gentleware.com/products/documentation/PoseidonUsersGuide/x1462.html>



# UML (Unified modeling language)

## - designing with a sequence of refinements -

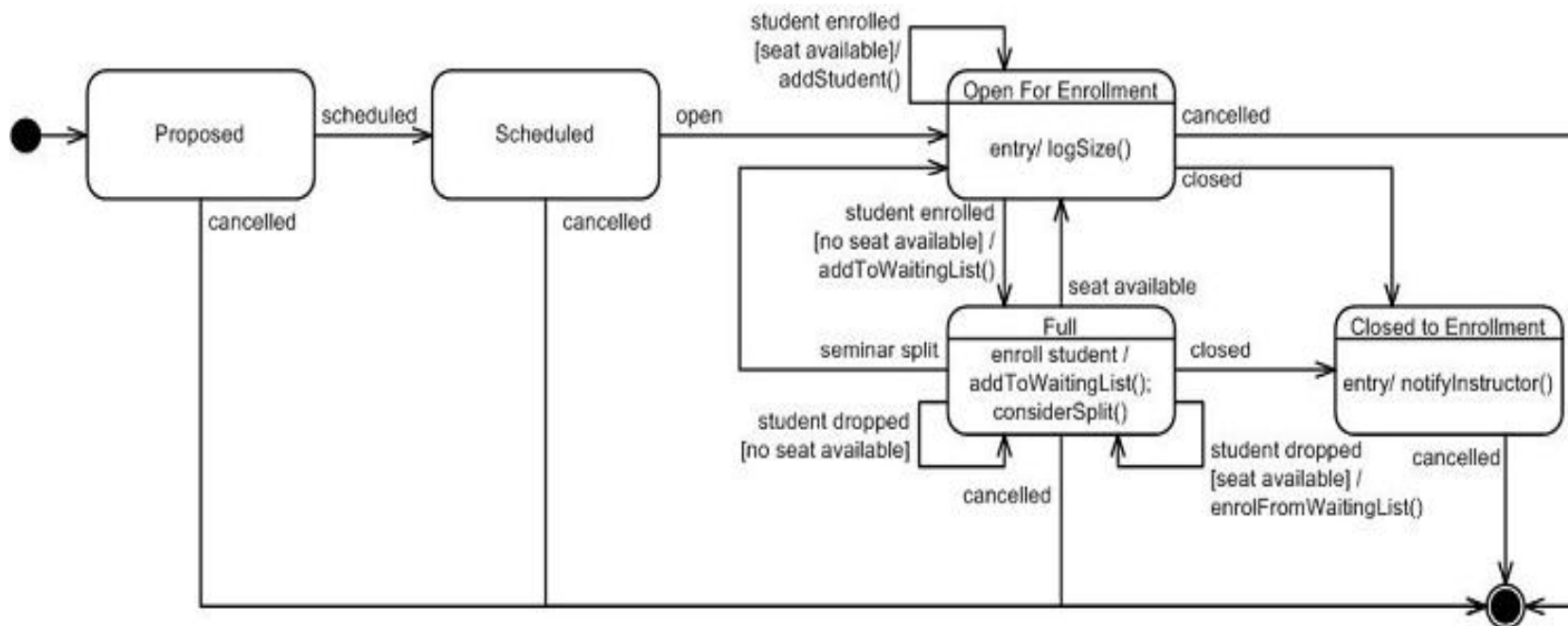


From: [www.sdmagazine.com/documents/s=815/sdm0012c/](http://www.sdmagazine.com/documents/s=815/sdm0012c/)



# State machine diagrams (UML 2.x) State diagrams (UML 1.x)

**:State machine diagrams/State diagrams**  
UML حاوی انواع گوناگونی از StateCharts است.



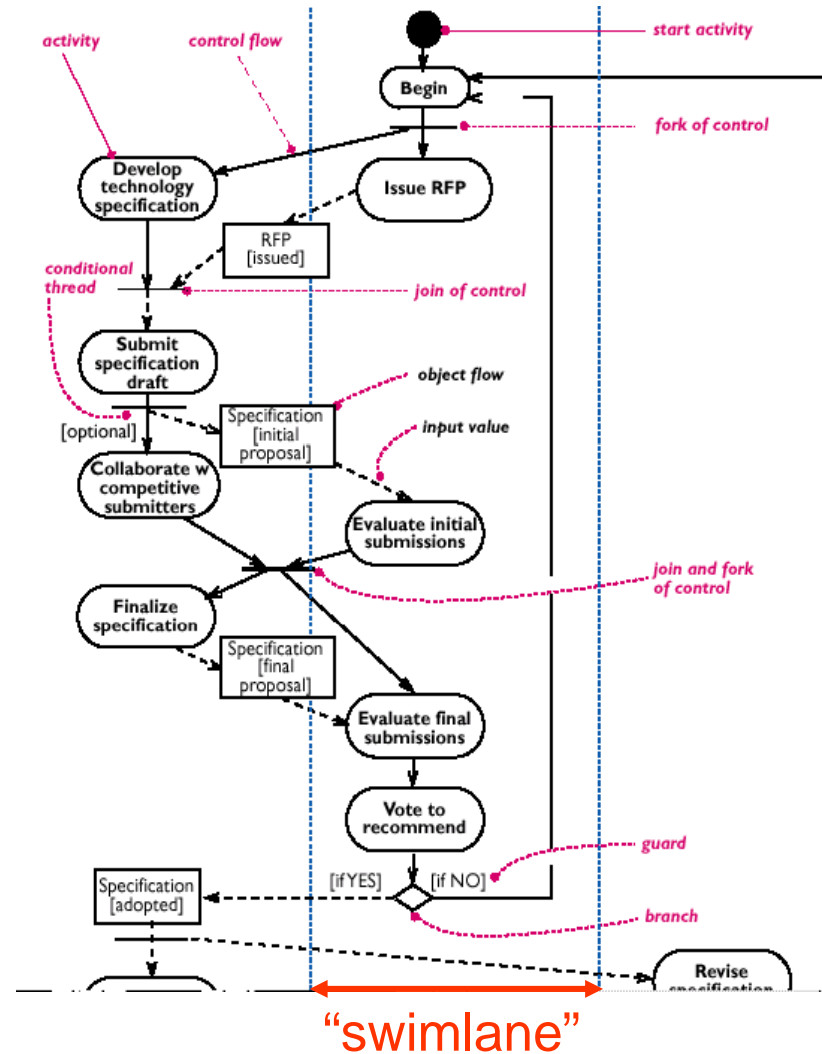
© Scott Ambler, Agile Modeling, //www.agilemodeling.com, 2003





# دیاگرام فعالیت (Activity diagram)

شبکه‌های پتری  
توسعه‌یافته.  
حاوی ساختار تصمیم  
(مانند فلوچارت‌ها)  
نمادگذاری گرافیکی  
مشابه با SDL

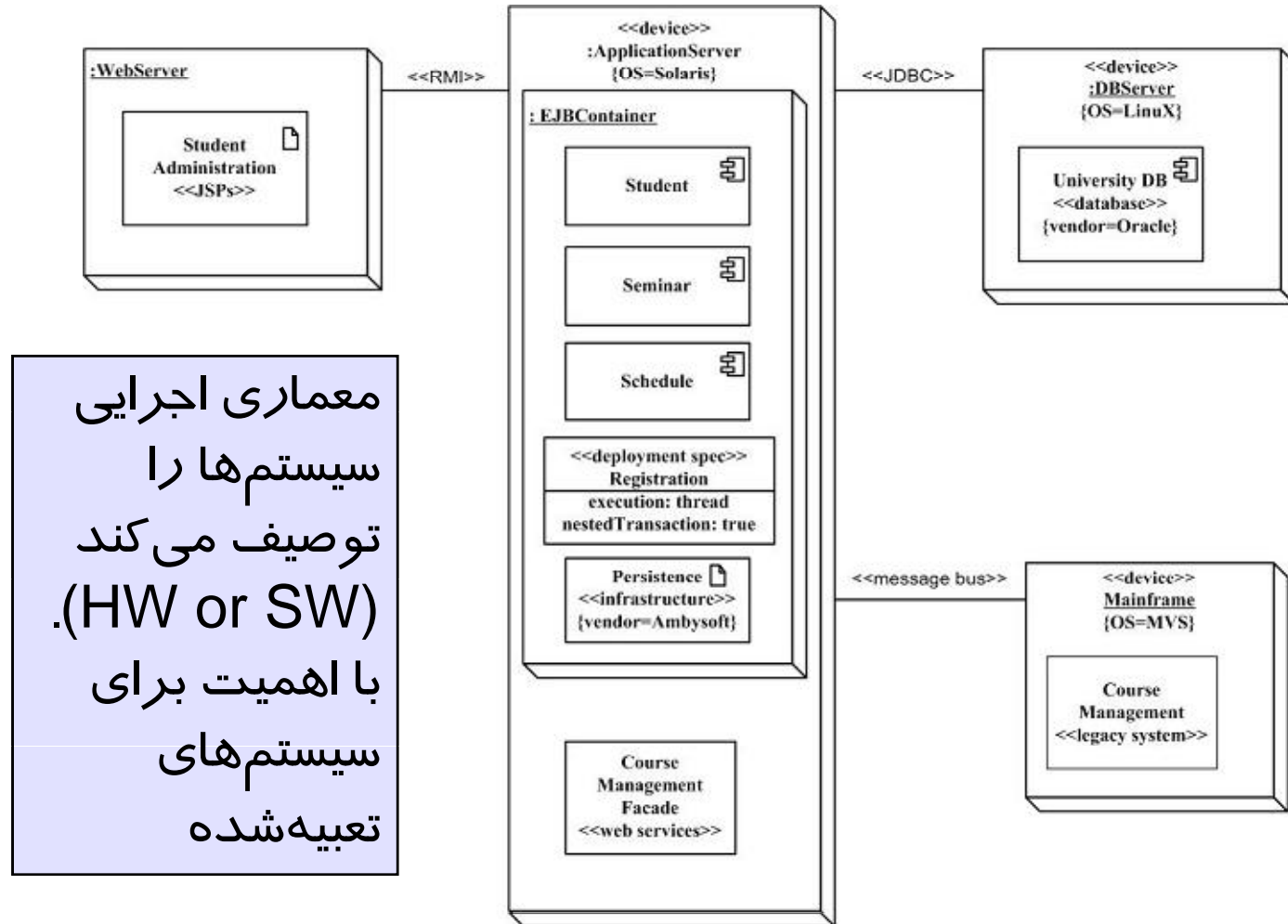


© Cris Kobryn: UML 2001: A Standardization Odyssey, CACM, October, 1999



# دیagram بکارگیری (Deployment diagram)

Example including some details:



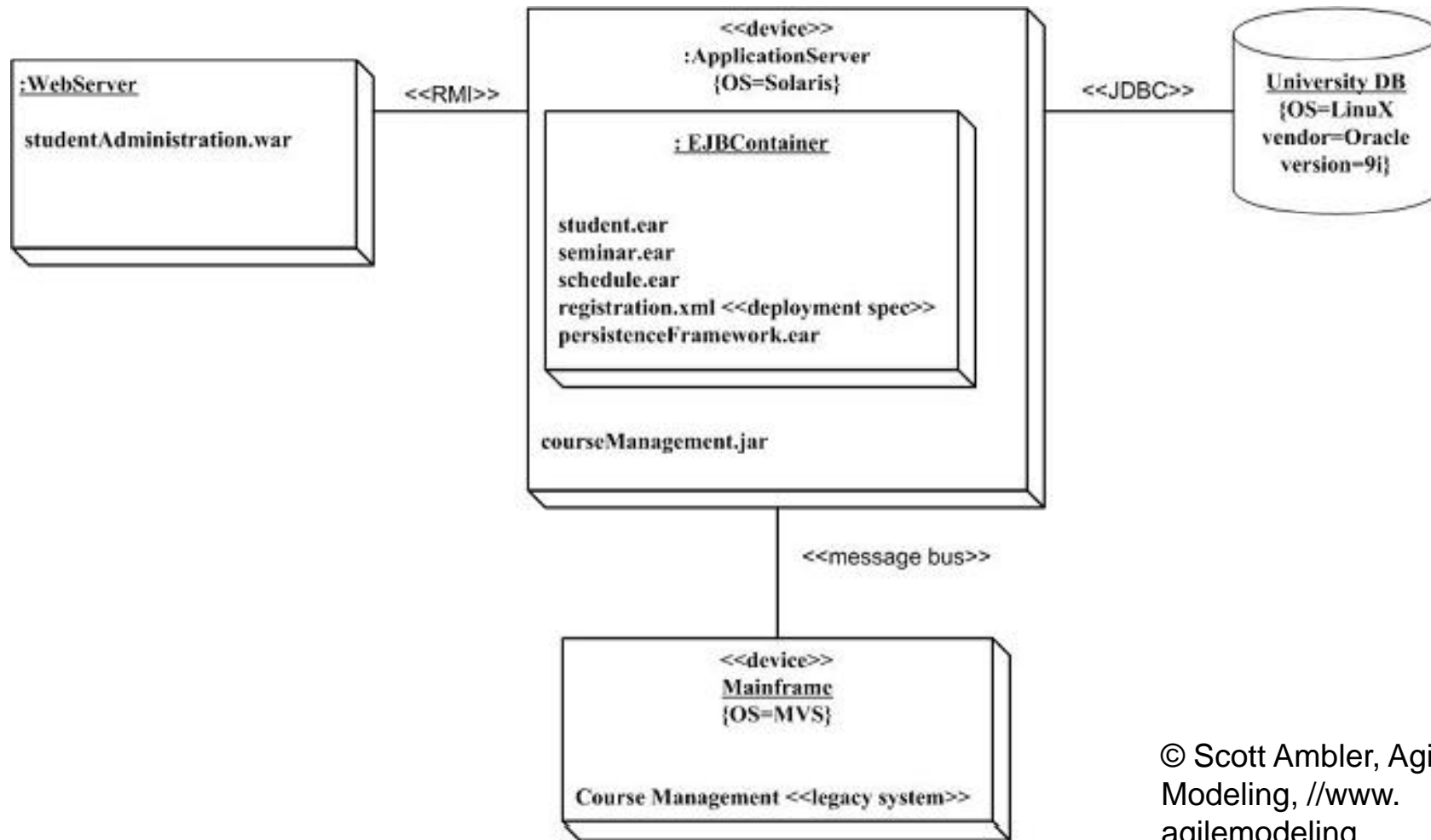
معماری اجرایی سیستم‌ها را توصیف می‌کند (HW or SW) با اهمیت برای سیستم‌های تعبیه شده

© Scott Ambler, Agile Modeling, //www.gilemodeling.com, 2003



# Deployment diagram

- More concise example -

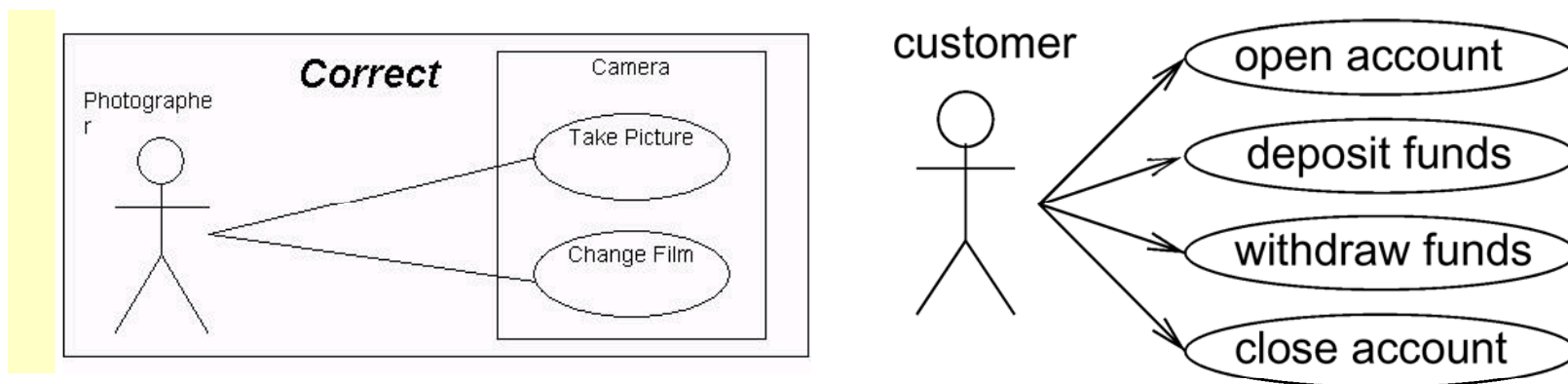


© Scott Ambler, Agile Modeling, //www.agilemodeling.com, 2003



# دیاگرام مورد کاربرد (Use case diagram)

Captures typical application scenarios



[//sds.hss.cmu.edu/courses/Syllabi/ids/271/umlfaq.asp](http://sds.hss.cmu.edu/courses/Syllabi/ids/271/umlfaq.asp)

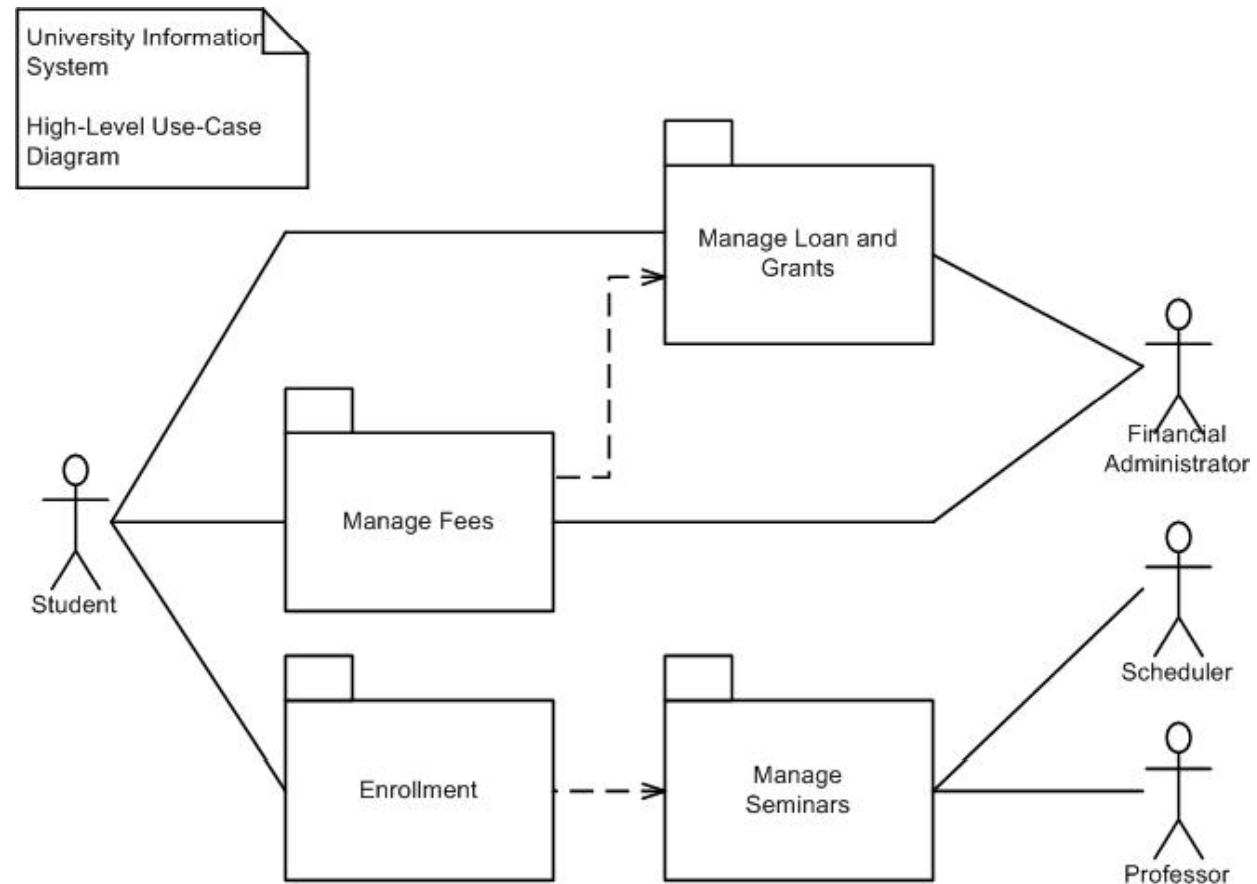
© Scott Ambler, Agile Modeling,  
[//www.agilemodeling.com](http://www.agilemodeling.com), 2003



## دیاگرام بسته (Package diagram)

افراز کردن به بسته‌ها را بازنمایی می‌کند. سلسله‌مراتب را تعریف می‌نماید.

**مثال: دیاگرام مورد کاربرد بسته**

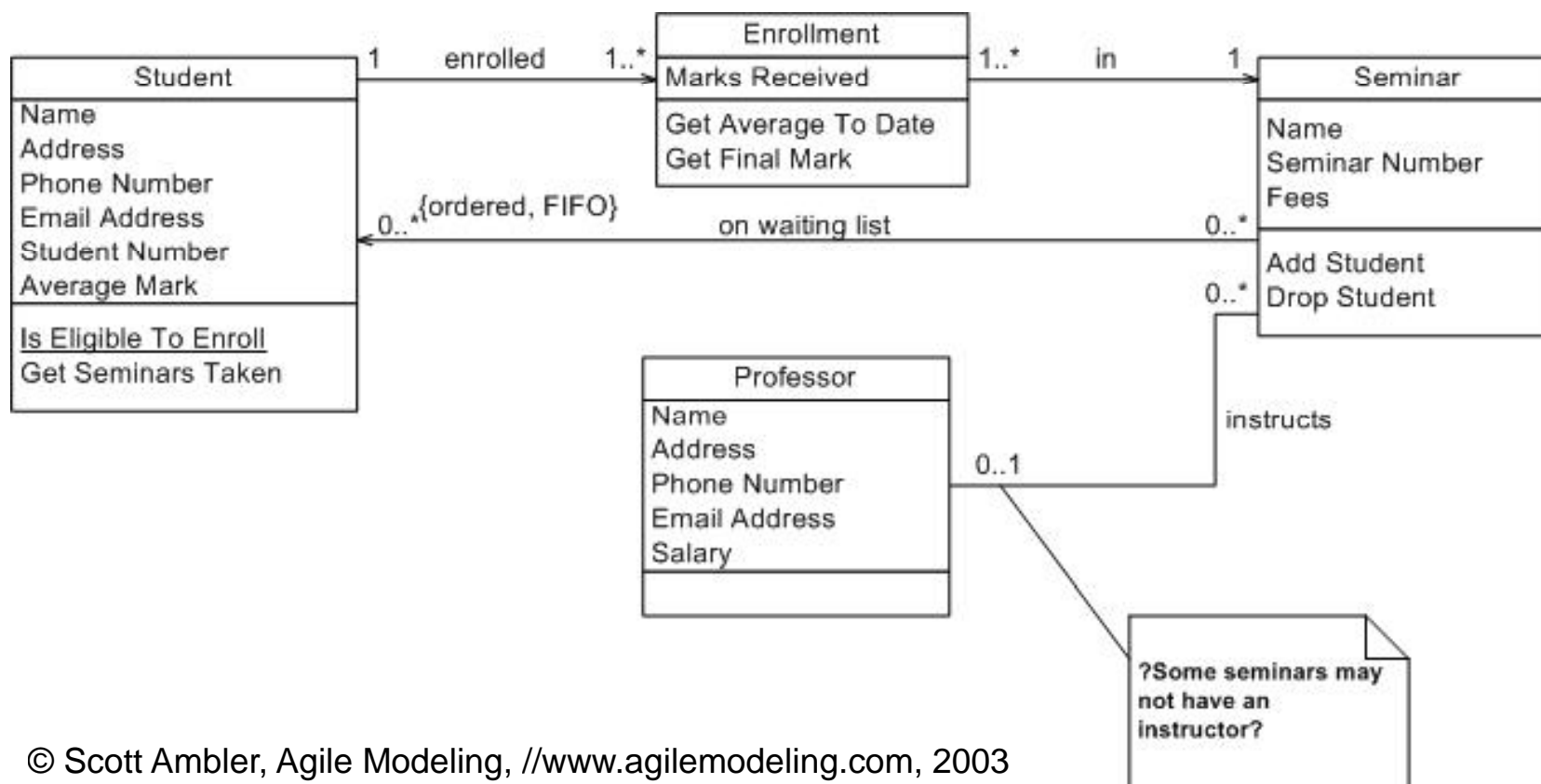


© Scott Ambler, Agile Modeling, //www.agilemodeling.com, 2003



# دیاگرام‌های کلاس (Class diagrams)

کلاس‌های اشیا را توصیف می‌کند.

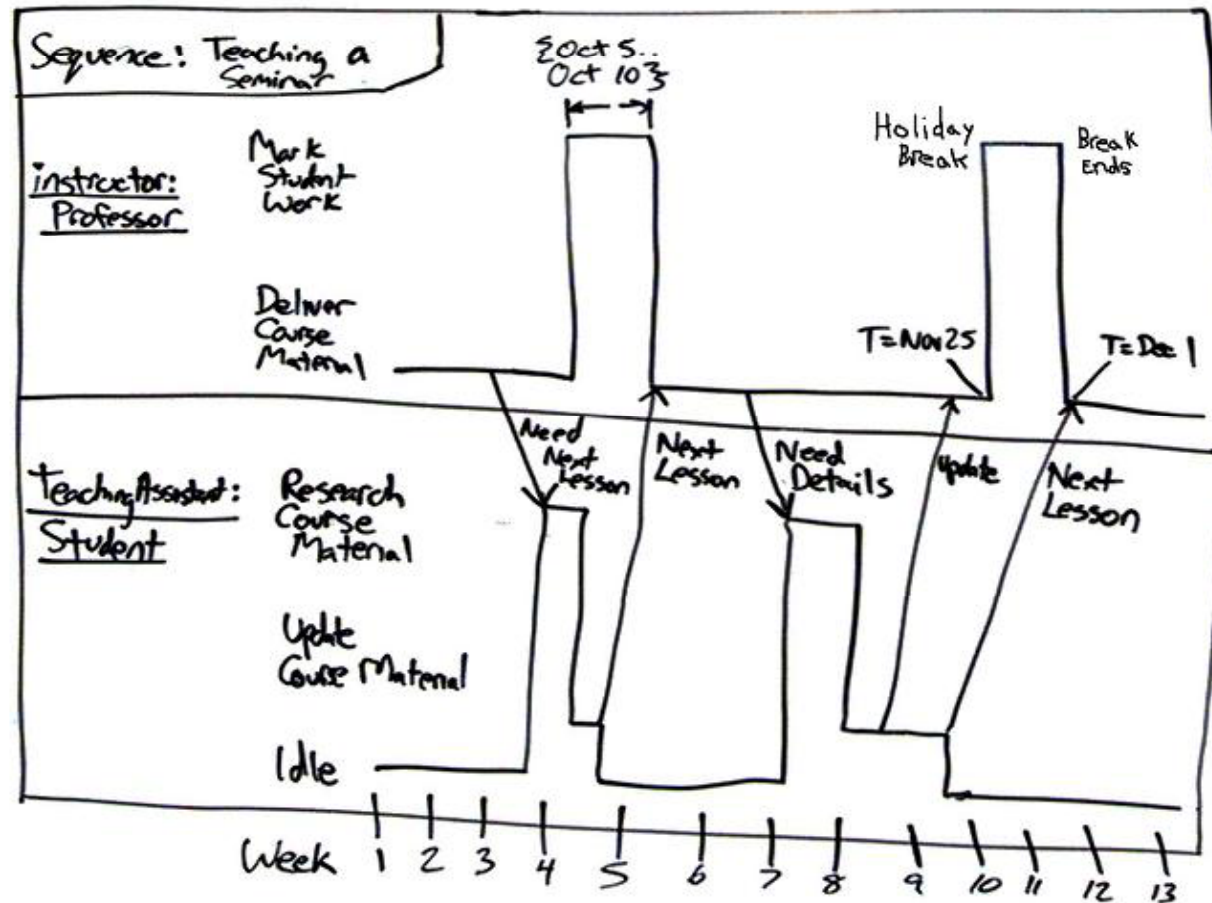


© Scott Ambler, Agile Modeling, //www.agilemodeling.com, 2003



# دیاگرام‌های زمانی (Timing diagrams)

می‌تواند  
برای نشان  
دادن تغییرات  
در یک شیء در  
طول زمان  
استفاده شود.



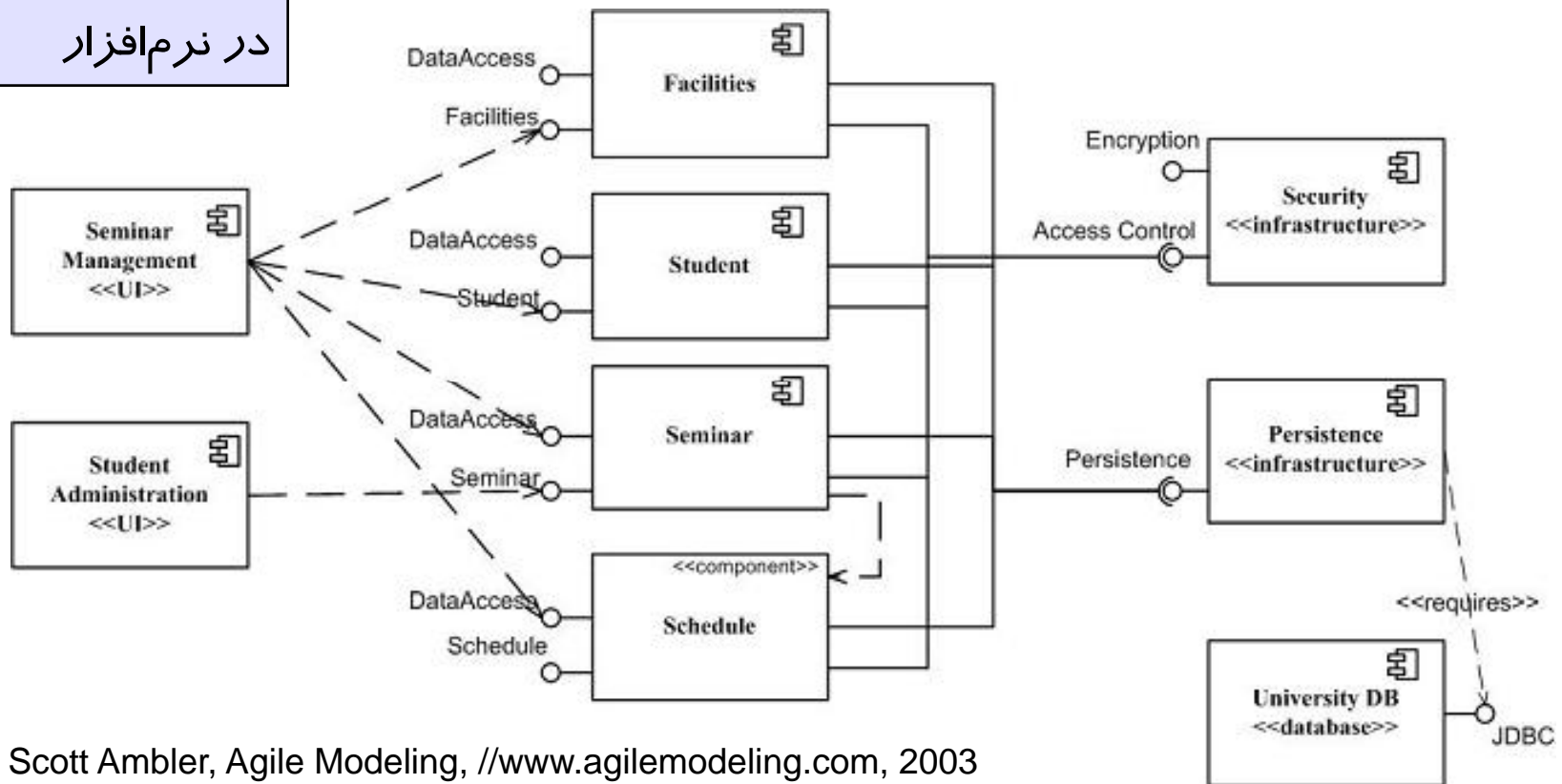
© Scott Ambler, Agile Modeling, //www.agilemodeling.com, 2003



# دیاگرام مؤلفه (Component diagram)

بازنمایی  
مؤلفه‌های  
استفاده شده  
در نرم‌افزار

.. model the business software architecture, the technical software architecture, ... . Physical architecture issues, in particular hardware issues, are better addressed via UML deployment diagrams ...



© Scott Ambler, Agile Modeling, //www.agilemodeling.com, 2003





## دیاگرام‌های دیگر

- **Communication diagram**  
(called collaboration diagram in UML 1.x)
  - **Object diagrams**
  - **Interaction overview diagrams**
  - **Composite structure diagrams**
- } Less frequently used



## ارزیابی

مشخص سازی دقیق مفاهیم؟  
معمولاً با SDL و C++ ترکیب می شود.



## UML برای سیستم‌های بی‌درنگ (Real-time)

ابتدائاً برای سیستم‌های بی‌درنگ طراحی نشده است.  
فاقد ویژگی‌های (1998):

- افراز نرم‌افزار به وظایف و پردازش‌ها
- مشخص کردن زمان
- مشخص‌سازی مؤلفه‌های سخت‌افزاری

پروژه‌هایی برای تعریف real-time UML بر اساس کارهای قبلی

- ROOM [Selic] is an object-oriented methodology for real-time systems developed originally at Bell-Northern Research.
- “UML profile for schedulability, performance and time”  
<http://www.rational.com/uml/resources/documentation>
- ...



## خلاصه

- **Message sequence charts**
  - Original (ITU) version, Time-Distance diagrams
  - Life sequence charts, Sequence diagram in UML
- **UML**
  - State machine diagram (StateChart-like)
  - Activity diagram (extended Petri nets)
  - Deployment diagram (exec. arch.), Use case diagram
  - Package diagram (hierarchy), Class diagrams,
  - Timing diagrams (UML 2.0), UML for real-time?

