CSE403 ● Software engineering ● sp12

Week 3				
Monday	Tuesday	Wednesday	Thursday	Friday
• Design • Reading II due	• Group meetings • SRS due	• Design	• UML	• Design • Progress report due

Today's pre-apology:
a couple of far too busy slides

(De)composition

- Functional decomposition vs. information hiding
- Architectural composition – pipes & filters, layers, etc.
- Object-oriented – aggregation, inheritance, etc.

KWIC

The KWIC index system accepts an ordered set of lines; …Any line may be "circularly shifted" by repeatedly removing the first word and appending it at the end of the line. The KWIC index system outputs a list of all circular shifts of all lines in alphabetical order.

```
Jokes are silly!
Mozart jokes are really silly.
Notkin is not a comic.
```

```
a comic notkin is not
are really silly mozart jokes
are silly jokes
comic notkin is not a
is not a comic notkin
jokes are really silly mozart
jokes are silly
mozart jokes are really silly
not a comic notkin is
notkin is not a comic
really silly mozart jokes are
silly jokes are
silly mozart jokes are really
```

```
      Notkin is not a comic!
     Mozart jokes are really silly.
            Jokes are silly!
    Notkin is not a comic!
             Notkin is not a comic!
             Mozart jokes are really silly.
                    Jokes are silly!
                    Mozart jokes are …
          Notkin is not a comic!
                    Notkin is not a comic!
   Mozart jokes are really silly.
          Jokes are silly!
 … jokes are really silly!
```

Note: may not produce identical output to previous slide

KWIC script: (de)composition

```
awk '{print $0
for (i = length($0); i > 0; i--)
  if (substr($0,i,1) == " ")
        print substr($0,i+1) "\t" substr($0,1,i-1)
}' $1 | sort -f | awk '
BEGIN {FS = "\t"; WID = 30}
{printf("%" WID "s      %s\n",
        substr($2,length($2)-WID+1),substr($1,1,WID))
}'
```

composition: Unix pipe

What can change easily? What can't?

`input`

`generate shifts`

`sort`

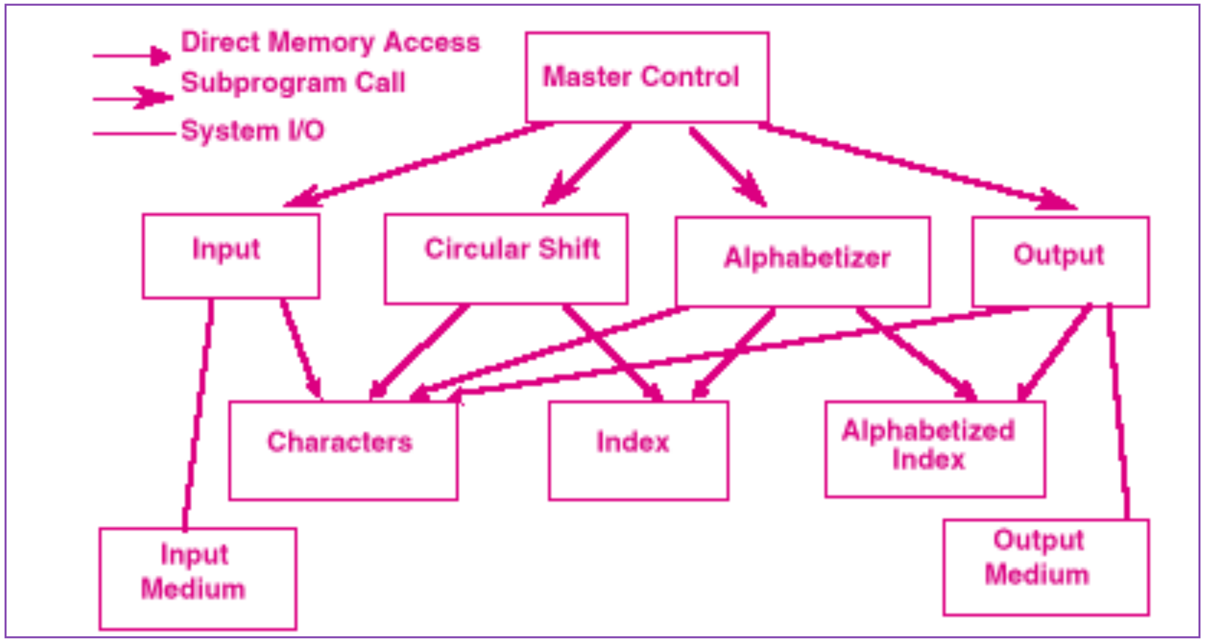
`output`

Generate k copies of each line of k words. Split each copy at a different word to produce all rotations of the line; mark the split word with a tab

Split each line at the tab, order the two pieces, truncate each piece to ≤30 characters, and output

Parnas: functional decomposition

Partially *obsolete design*

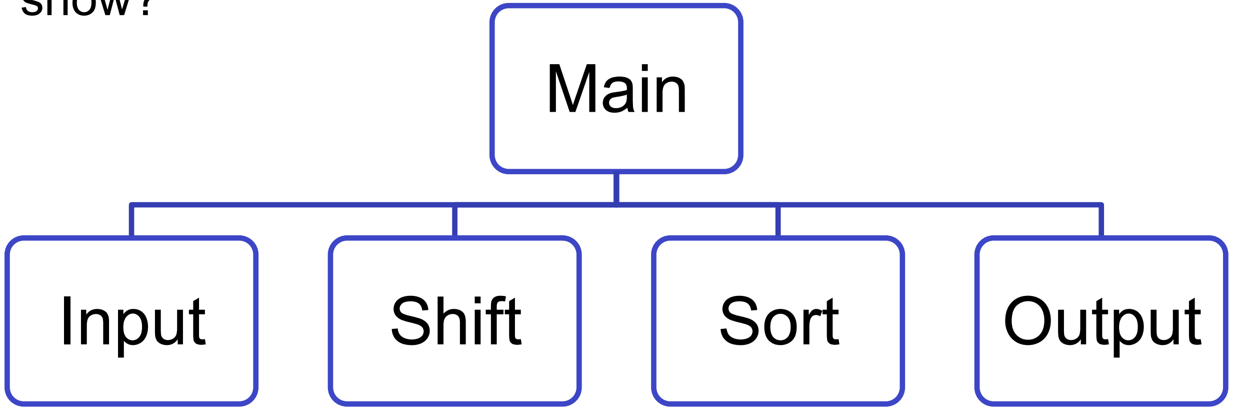


Partial list of possible changes

- Representation of lines, words, characters; storing on disk vs. in-memory – *basically obsolete*
- Incremental vs. monolithic sorting algorithms
  - `[shift; sort]` *vs.* `[for each shift insert into sorted list]`
- Eliminating noise words

Functional decomposition

- Top-down design – breaking each high-level function into more manageable parts
- Well-suited to program correctness, defining and proving pre- and post-conditions
- What does this diagram actually show?



- What would have to change?
  - Incremental vs. monolithic sorting algorithms
  - Eliminating noise words
  - Interactive UI
  - …

Connecting design and change

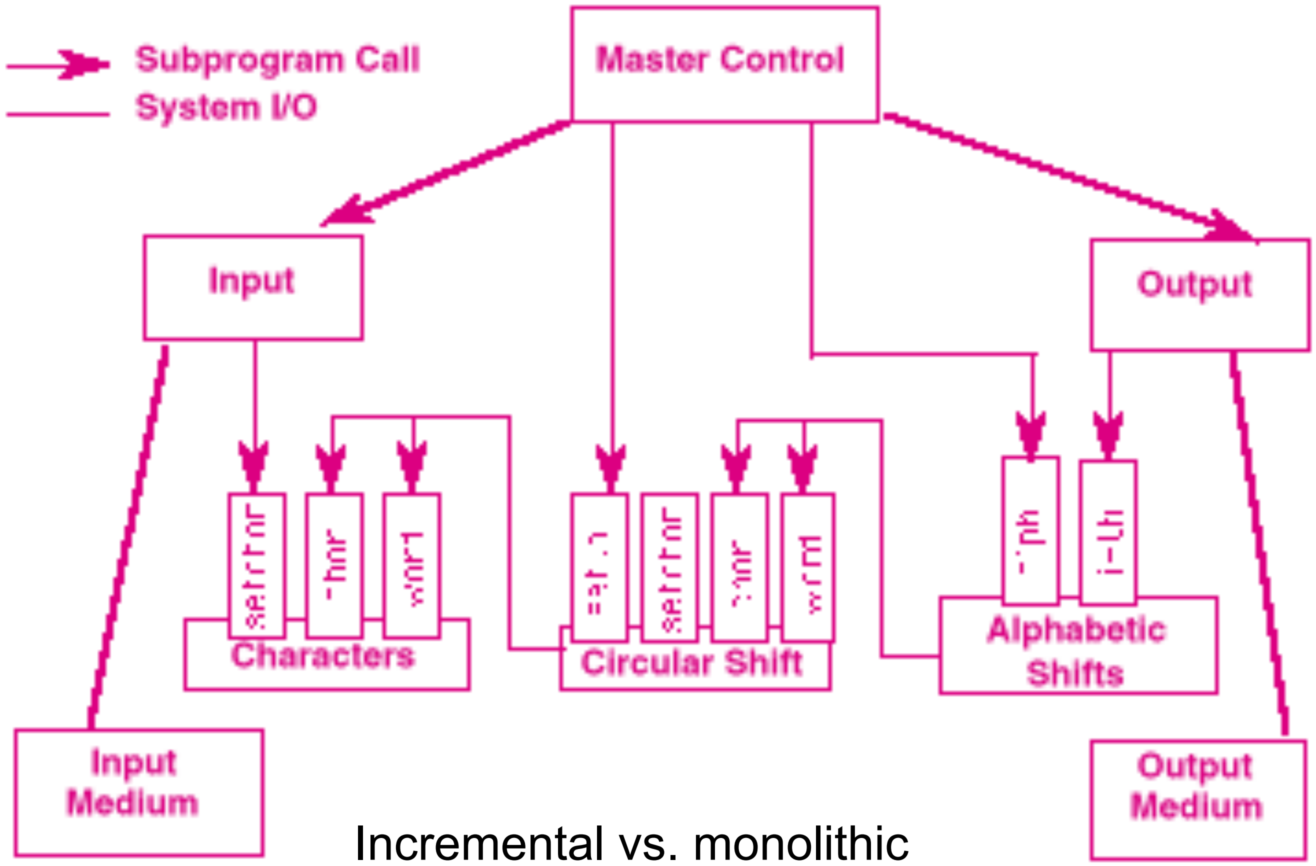
- The functional decomposition is based on breaking the computation down into more manageable parts
- But the questions are about, “What will likely change?”
- There is no *a priori* reason that a design based on functional decomposition will be suitable to support likely kinds of change
- What if we decomposed with anticipated change in mind?

This is the core notion of Parnas’ ***information hiding –*** *connecting design with anticipated change*

Information hiding

- Decide on likely changes
  - [Truth in advertising: this is hard and perhaps not doable well]
- Design interfaces that protect clients from those changes – the interfaces define a contract that the clients can rely on
- Implement the interface based on a best decision
- If the decision later changes, re-implement the interface but continue to satisfy the contract

KWIC: information hiding



***Partially** obsolete design*

Incremental vs. monolithic sorting? Eliminating noise words? Interactive UI? …?

Other examples of

- A software system based on a physical sensor – maybe an accelerometer on a mobile phone
  - Design to more easily accommodate improvements in the accuracy of the sensor in later versions of the phone?
- Tax software (TurboTax, TaxCut, etc.)
  - Ever notice how Congress changes tax laws every year?
  - These companies can't be late to market with their products
  - Allowing tax rates to change may be easy, but what about eligibility, etc.?

Tax Credit of up to $8,000 for First-Time Homebuyers …

…Existing homebuyers are eligible to receive a tax credit of 10% of the purchase price up to $6,500 if they bought and closed on a replacement home by September 30, 2010. In order to be eligible for the credit, homeowners must have lived in the same principal residence for any five-consecutive-year period during the past eight years. They are not required to sell or dispose of their current home, but the new home must become their principal residence.

If you purchased and closed on a primary residence before September 30, 2010, and are a "first-time" homebuyer, you can qualify for a tax credit of 10% of the purchase price up to $8,000. To be eligible, you must not have owned a residence in the United States in the previous three years.

To qualify for either credit, you must have signed a binding contract to buy the house by April 30, 2010, and closed on it by September 30, 2010.

Members of the armed forces who were on official extended duty outside of the United States for at least 90 days between Jan .1, 2009, and May 1, 2010, may qualify for a one-year extension.

The credit is refundable to the extent it exceeds your regular tax liability, which means that if it more than offsets your tax liability, you'll get a refund check. But it does not offset the Alternative Minimum Tax.

In addition, income limits were expanded from earlier versions of the credit. Homebuyers who file as single or head-of-household taxpayers can claim the full credit if their modified adjusted gross income (MAGI) is less than $125,000. For married couples filing a joint return, the combined income limit is $225,000.

Single or head-of-household taxpayers who earn between $125,000 and $145,000, and married couples who earn between $225,000 and $245,000 are eligible to receive a partial credit. The credit is not available for single taxpayers whose MAGI is greater than $145,000 and married couples with a MAGI over $245,000. Also, homes costing more than $800,000 are not eligible for the credit.

Some "last" things on information hiding

- Encapsulation – defining an interface that may keep some elements of the corresponding implementation private – is not always information hiding
  - Encapsulation by itself does not focus on change as a design principle – separates public from private components
- Abstract data types are a form of information hiding that focuses on hiding concrete data representations – that might change – and the implementation of the operations defined on the abstractions
- What defines the "contract" by an information hiding interface? The documentation? Performance – promised or inferred? Is the client always right?
  - **These are really complicated questions!**… that led in part to the aspect-oriented design and programming paradigm

Software architecture

Mary Shaw and David Garlan … have been named co-recipients of the Outstanding Research Award for 2011 presented by the Association for Computing Machinery's Special Interest Group on Software Engineering (SIGSOFT).

Shaw, the Alan J. Perlis Professor…, and Garlan, professor of computer science … in the [Carnegie Mellon] School of Computer Science, were recognized by SIGSOFT for their "significant and lasting software engineering research contributions through the development and promotion of software architecture."

A. Capturing, cataloguing, and exploiting experience in software designs
B. Allowing reasoning on classes of designs

- By adopting software architectures with known properties, one can increase confidence in the properties of your software system
  - [Think about Feynman's observations]

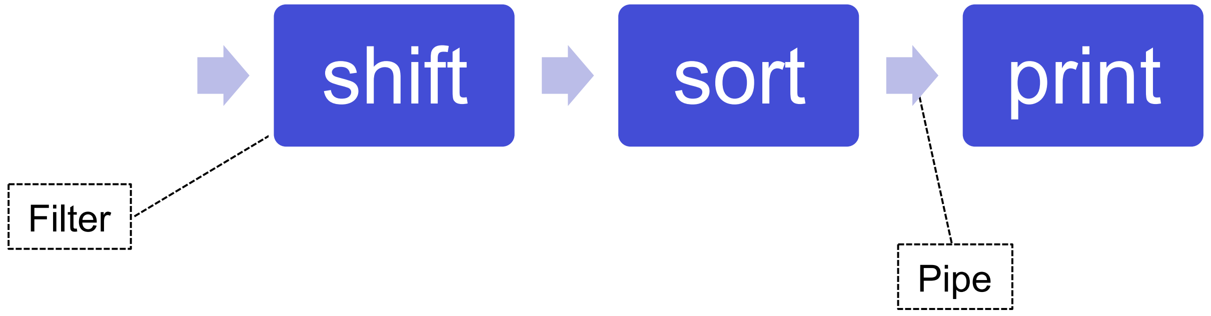
Architecture: two parts

- *Components* define the basic computations comprising a software system
  - Abstract data types, filters, etc.
- *Connectors* define the interconnections between components
  - Procedure call, event announcement, etc.

Architectural style

- Defines the vocabulary of components and connectors for a family (style) of software systems
- Constraints on the components and connectors and on their combination
  - Topological constraints (no cycles, register/ announce relationships, etc.)
  - Execution constraints (timing, etc.)
- By choosing a style, one gets all the known properties of that style

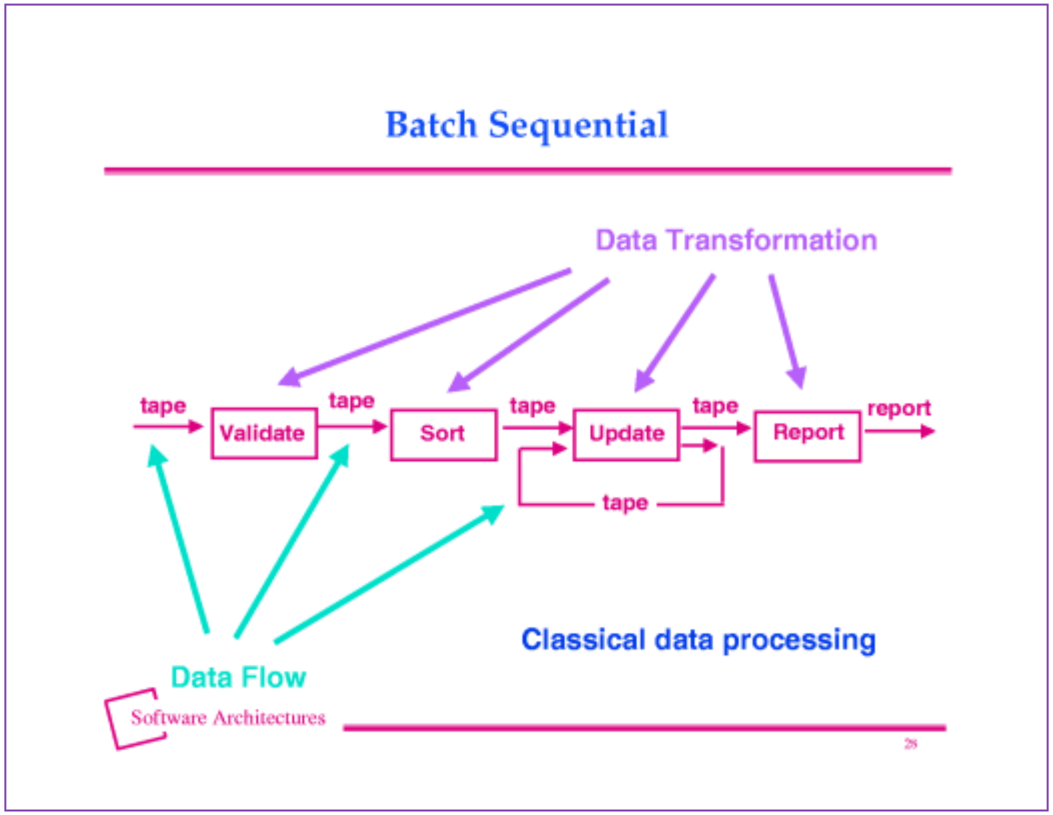
Classic style example: pipes & filters



a) Pipes must compute local transformations
b) Filters must not share state with other filters
c) There must be no cycles

- If these constraints are satisfied, it is a pipe & filter system, and some properties are ensured – for example, lack of deadlock
- But if they are not satisfied (even a “little” bit) no guarantees are provided
- One can think of the constraints as obligations (pre-conditions of a sort) on the designer

Pipe & filter?

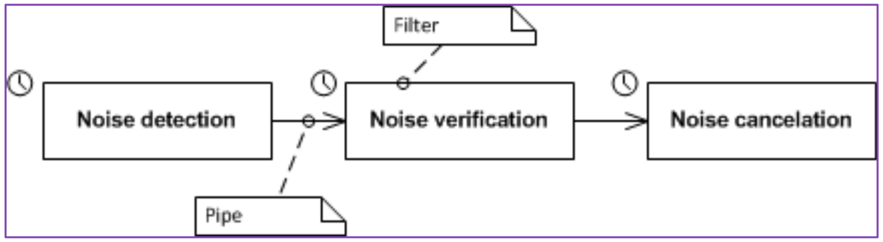


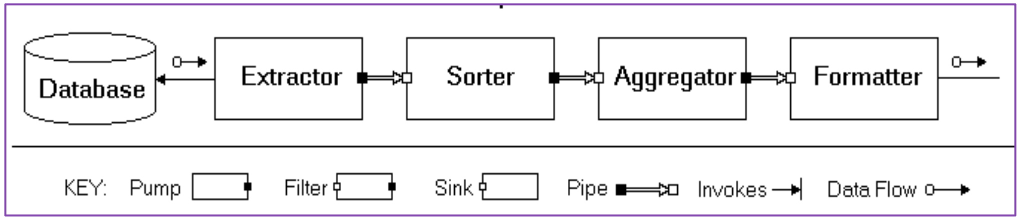
- Processing steps are independent
- Each runs to completion before moving to next step
- Data transmitted as a whole between steps

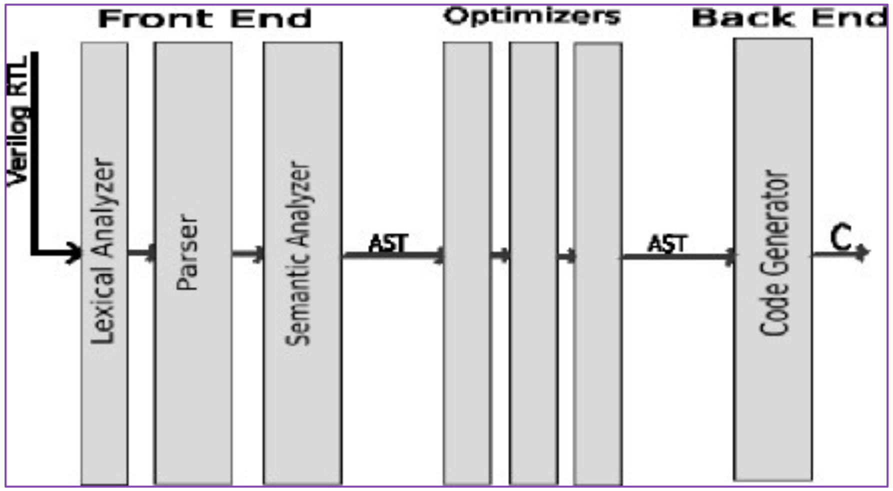
a) Pipes must compute local transformations

b) Filters must not share state with other filters

c) There must be no cycles


Filter
Noise detection
Noise verification
Noise cancelation
Pipe

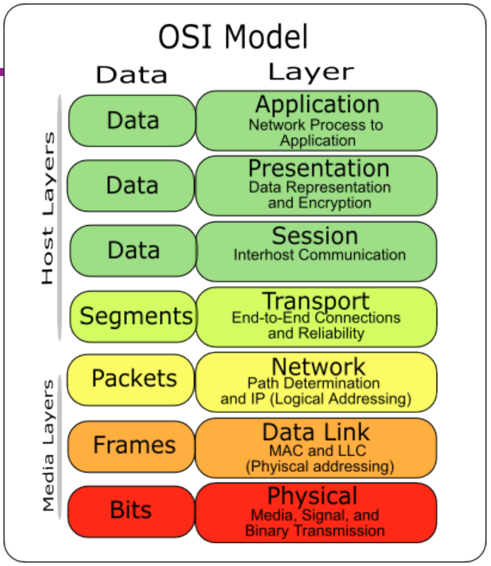

Database
Extractor
Sorter
Aggregator
Formatter
KEY: Pump
Filter
Sink
Pipe
Invokes
Data Flow


Front End
Optimizers
Back End
Verilog RTL
Lexical Analyzer
Parser
Semantic Analyzer
AST
AST
Code Generator
C

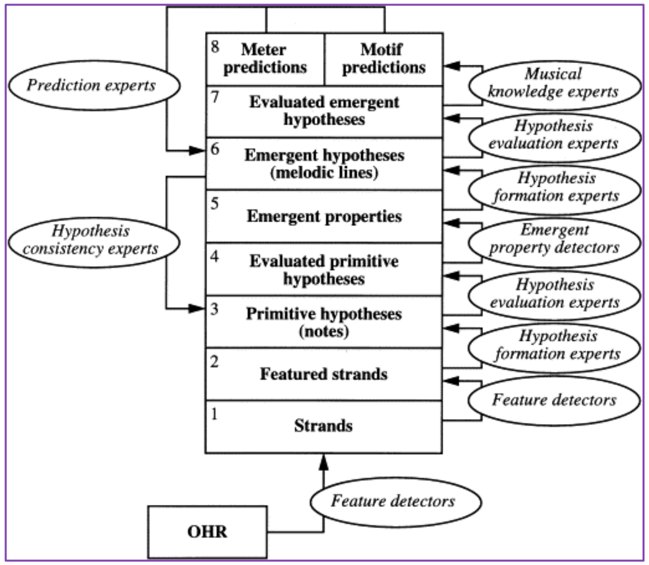
Other common styles

- Layered systems
- Blackboard systems
- …many others

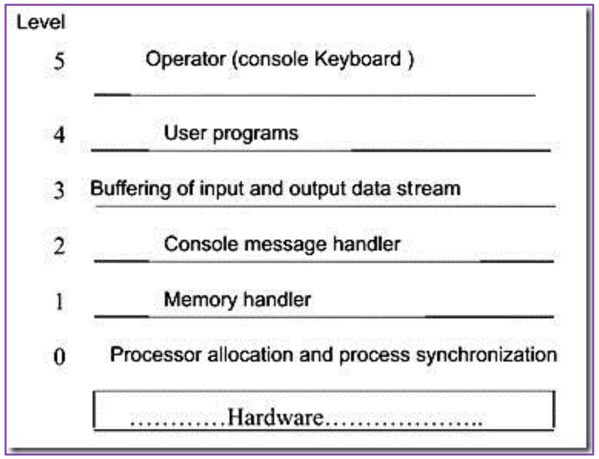
Key: by constraining your designs you can increase your confidence in the software's properties



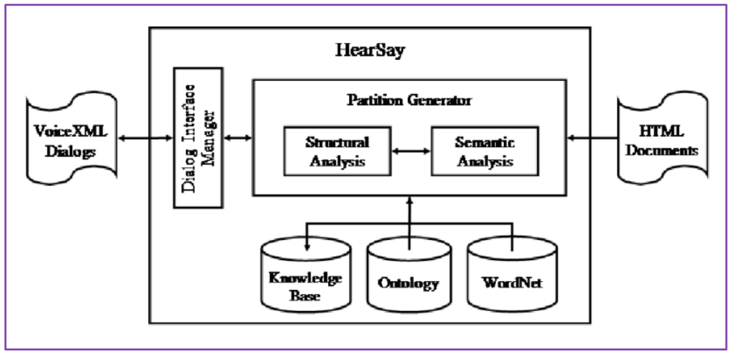
OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission



Prediction experts
Hypothesis consistency experts
8 Meter predictions
Motif predictions
7 Evaluated emergent hypotheses
6 Emergent hypotheses (melodic lines)
5 Emergent properties
4 Evaluated primitive hypotheses
3 Primitive hypotheses (notes)
2 Featured strands
1 Strands
Musical knowledge experts
Hypothesis evaluation experts
Hypothesis formation experts
Emergent property detectors
Hypothesis evaluation experts
Hypothesis formation experts
Feature detectors
Feature detectors
OHR

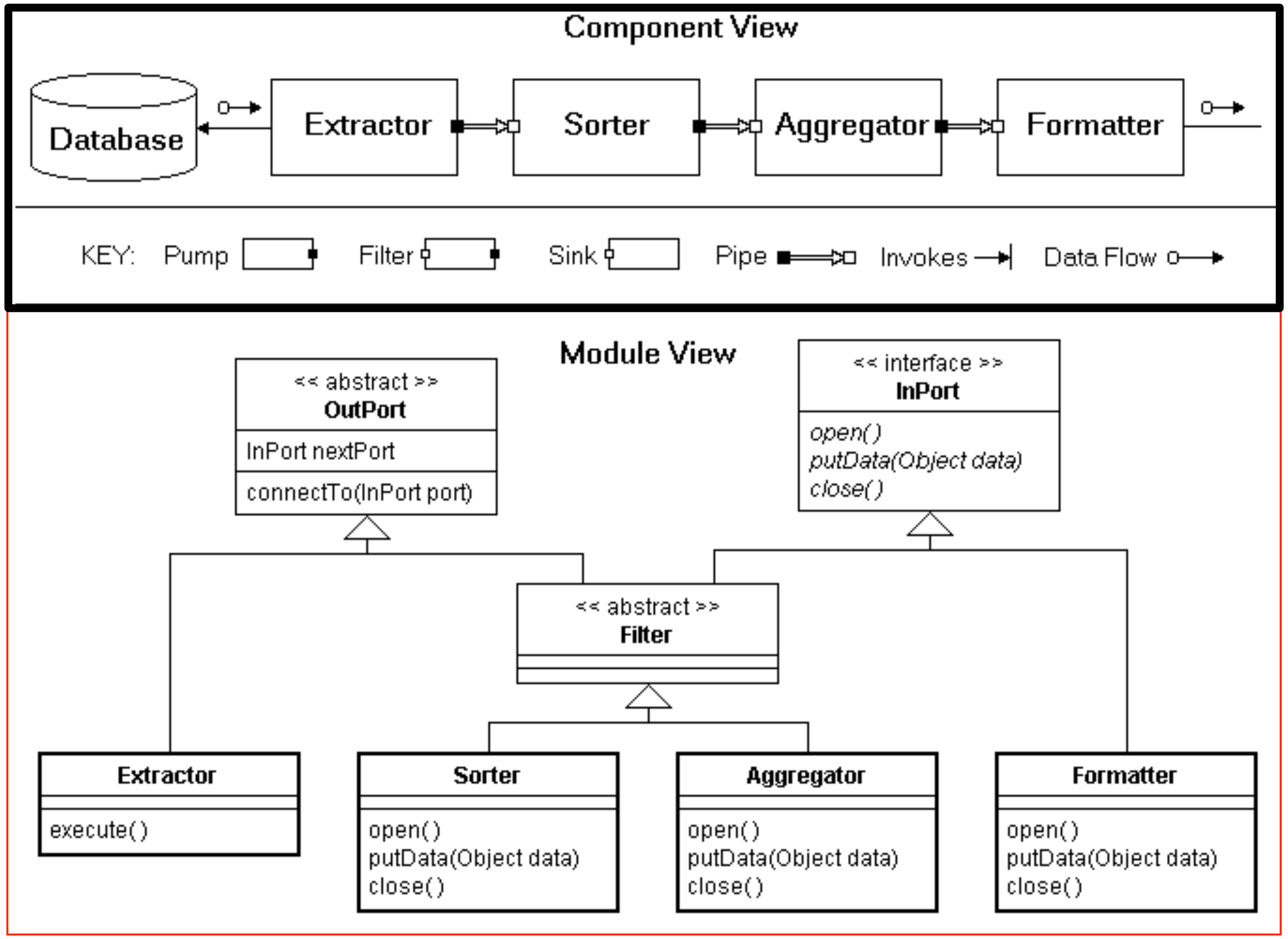


Level
5 Operator (console Keyboard)
4 User programs
3 Buffering of input and output data stream
2 Console message handler
1 Memory handler
0 Processor allocation and process synchronization
…………Hardware……………….



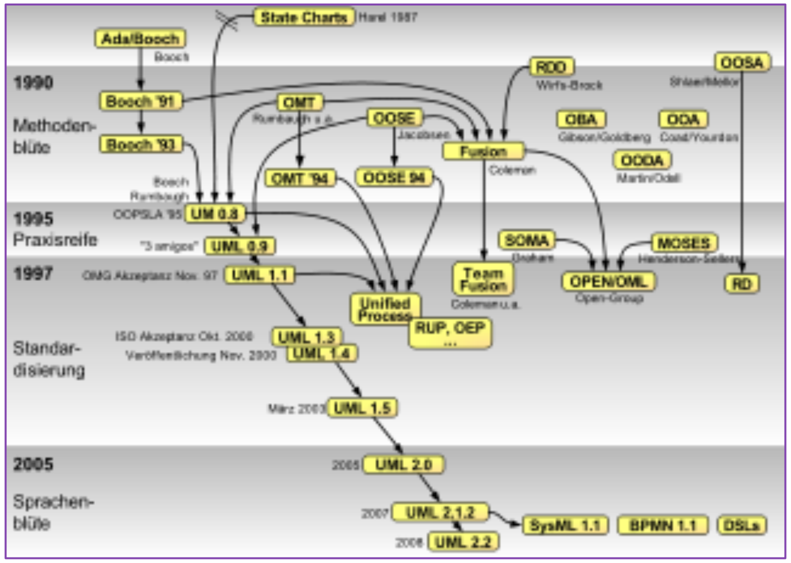
More to design? Of course

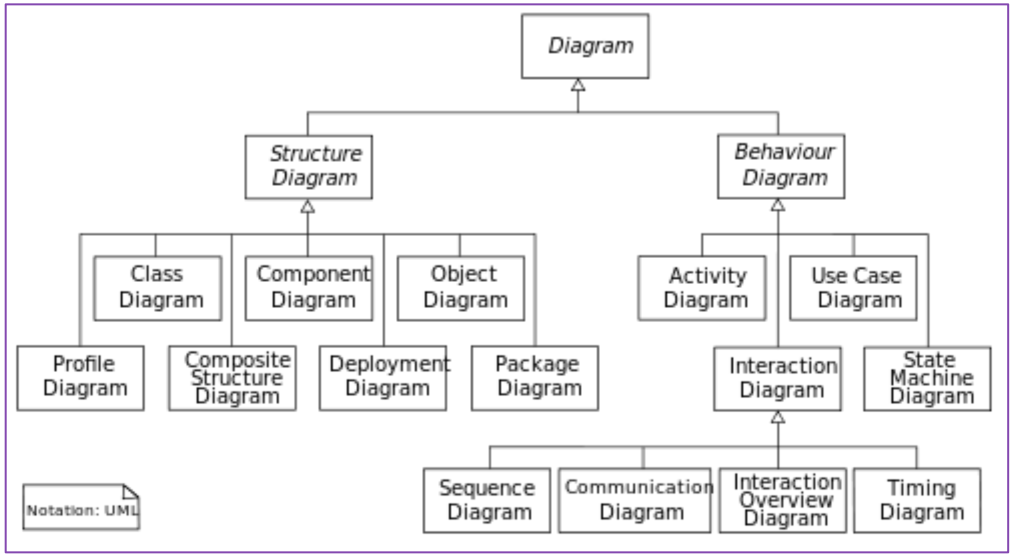
- Pipe & filter from earlier slide
- Now with more design information
- Classes, interfaces, methods, …



Object-oriented design notations

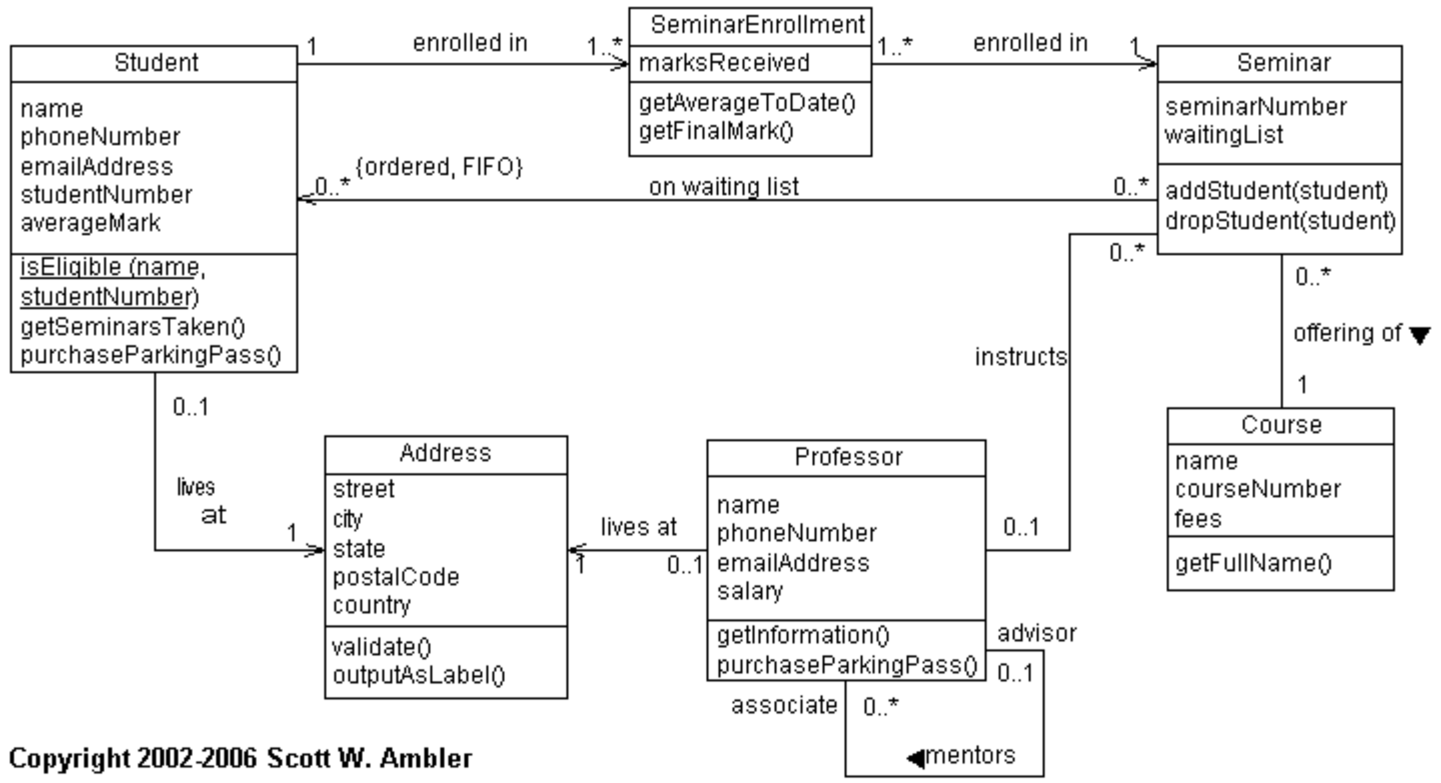
- “Modeling” languages
- UML is by far the most used (and misused), with 14 different kinds of diagrams
- Very wide-spectrum use (and misuse)





- Two most important diagrams are class diagrams (structure, attributes, and relationships) and sequence diagrams (how objects communicate via a sequence of messages)

UML class diagrams



association (one- and two-way), aggregation, inheritance, multiplicity, interfaces, abstract, visibility, …

More in section tomorrow (Anton)

- Class and sequence diagrams
- Violet (free drawing package for UML)

- **Use UML to the degree it is a help for your team – if it's getting in the way, think hard about what aspects you want to use**

CSE403 ● Software engineering ● sp12

Week 3				
Monday	Tuesday	Wednesday	Thursday	Friday
• Design • Reading II due	• Group meetings • SRS due	• Design	• UML	• Design • Progress report due