CSE403 ● Software engineering ● sp12

Week 7-10				
Monday	Tuesday	Wednesday	Thursday	Friday
• Joel test & interviewing • No reading	• Groups	• Reviews	• Section	• Progress report due • Readings out
• Reading due	• Groups • Beta due		• Section	• Progress report due • Readings out
• No reading due	• Groups	• Midterm II • Reading covered [Notkin gone]	• No section	• Progress report due
Memorial Day Holiday	• Groups	• Final release due • **Project Pres. I**	• **Project Pres. II**	• **Project Pres. III**

Recall information hiding

- The core notion of Parnas' information hiding is connecting design with anticipated change
- Decide on likely changes
- 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

The information hiding "contract"

- Interfaces define the contract between the client and the implementation
  - The client doesn't care how the contract is implemented
  - The implementation doesn't care how the module is used
- Anticipated changes are supported as they are hidden by the interface

What defines the contract?

- What is included in the contract?
- What is excluded from the contract?

Implied-in-fact contract (wikipedia)

- "Although the parties may not have exchanged words of agreement, their actions may indicate that an agreement existed anyway.
  - *"For example*, when a patient goes to a doctor's appointment, his actions indicate he intends to receive treatment in exchange for paying reasonable/fair doctor's fees. Likewise, by seeing the patient, the doctor's actions indicate he intends to treat the patient in exchange for payment of the bill. Therefore, it seems that a contract actually existed between the doctor and the patient, even though nobody spoke any words of agreement. … In such a case, the court will probably find that … the parties had an implied contract. If the patient refuses to pay after being examined, he will have breached the implied contract."

Can or does such an implied contract hold between a client or clients and an implementation?

Consider

- Oracle produces and publishes an API
- Notkin LLC uses that API and, furthermore, publicly and frequently asks questions about its use of the API
- Oracle later modifies the API
- Notkin LLC's application breaks due to those modifications, which were not mentioned as possibilities during the public interactions
- Was there an implied-in-fact contract?

I have no idea at all! I'm not a lawyer, and I don't even play one on TV.

♥"It's complicated"

- Is it the documentation and only the documentation?
- What if the documentation is silent on a point?
- Can the client infer parts of the contract?
- Isn't the client always right?
- What if multiple clients end up with different expectations for the same implementation?

Clients sometimes get some power

- Compiler directives
  - Implementation has several choices
  - Client can direct the implementation to use a specific choice

    ```
    register int i;
    ```
    change a cache size
    configure which backend database to use
    …
- This somewhat begs the question: the above concerns hold for any specific implementation chosen by the client

This reasoning…

- …and identification of these problems as arising in programming languages, operating systems, and many more domains
- …led to the identification (by Kiczales et al.) of the notion of "open implementations"
- An open implementation has a standard "base" interface and a second "meta" interface
  - The meta interface helps the client control aspects of the implementation in explicit ways
- Further work led to the notion of aspect-oriented design and programming – I'll introduce this with a fly-through of some slides from a Kiczales keynote