1.264 Final Exam
Solutions
Fall, 2000

1. **Software process. (20 points)**

    a. **Is the prototyping development method appropriate in this case? Describe why or why not in 2-4 sentences. (5 points)**

Prototyping is not appropriate. A prototype already exists: the 1.264 class project, which was the first cycle in a spiral development process. There is little uncertainty about the functions, database or user interface that could be resolved by a prototype. If no software existed, or a massive change were required, building a prototype might be helpful, but not for incremental changes as in the current situation.

    b. **If prototyping is not appropriate, what method should be used? Name one or two methods that would be appropriate and briefly justify your choice in 2-4 sentences. If prototyping is appropriate, explain how it should be executed differently to avoid the problems in this case study in 2-4 sentences. (5 points)**

Continuing the spiral model is one reasonable method. The spiral model would immediately address the requirements changes needed by the users, have a short design and estimation cycle, build the code, test it and be ready for the inevitable set of further changes. The very short time frame for this project also supports the use of a spiral model. Other reasonable options include Design-to-schedule and evolutionary delivery are also reasonable choices: they are staged models with iterations of work

    c. **List at least 4 errors made by this development team in their execution of this project. For each error, describe what should have been done instead, in 1-2 sentences. (5 points)**

1.Wrong development process. Should have used spiral or similar model.
2. Unstructured requirements gathering process. Should have had defined requirements phase, followed by a design phase.
3. No resource estimation was done. Should have done time/effort estimation with function points or similar method when requirements changes were known.
4. Management (Pat) was clueless about software process and made unrealistic promises. Manager should know fundamentals of software process. Developers need to establish a software process even if the manager won't and try to educate the manager.
5. Manager should not make open-ended promises to end users ('give them what they want'). Use a requirements process to establish what will be delivered.
6. Management undermined morale with twice-daily cubicle checks, comments that the staff was not working hard enough, etc. Manager should listen and support staff.

d. **Outline the key steps in the development method that you proposed in step a above. List at least 4 steps; describe each in 1-2 sentences. (5 points)**

Spiral model:
1. Requirements. Gather business rules, data needs, user interface needs from users.
2. Design. Develop data model, user interface design, process descriptions based on requirements. Estimate resources required.
3. Coding. Develop to design.
4. Test. Unit test during coding; test plans; requirements, design, code reviews. System test.

2. **Data model (19 points)**

You must draw a data model that corresponds to this set of business rules. Follow these steps:

a. **Draw a box for each entity: give each an appropriate name (4 points)**

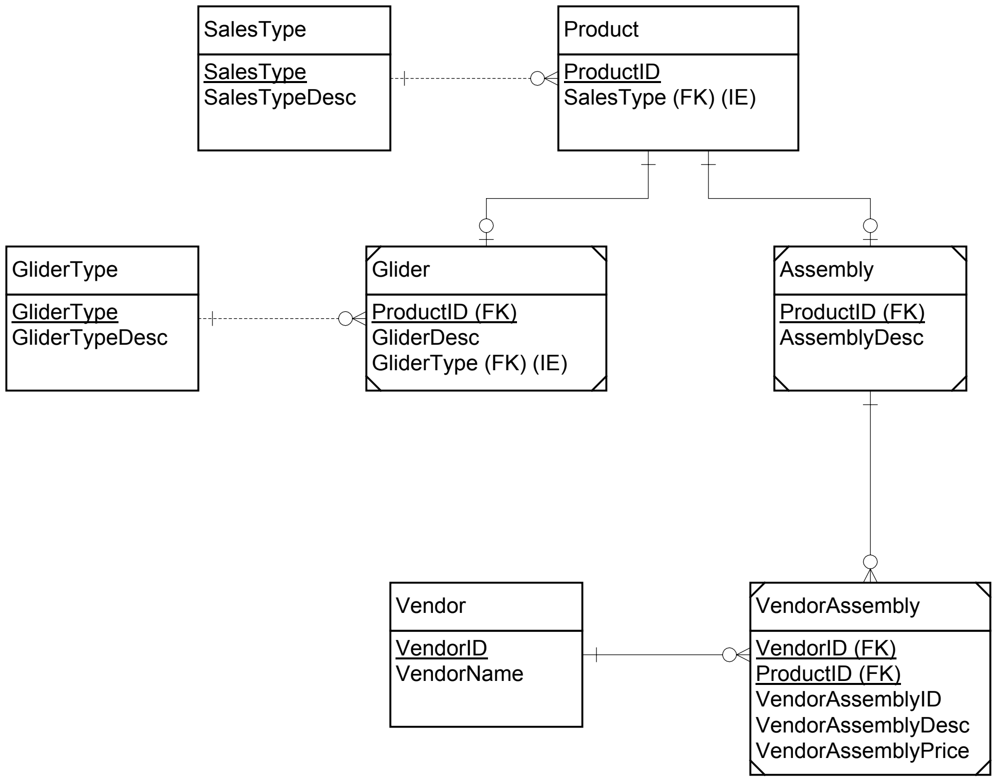
b. **List the attributes in the box for each entity (5 points)**

c. **Indicate the primary key for each entity either by underlining it or by placing the phrase (PK) next to its name. (2 points)**

d. **Draw all relationships between the entities in the model. Indicate foreign keys by placing the phrase (FK) next to attributes that are foreign keys. (5 points)**

e. **Indicate the cardinality of the relationship: many-many, many-one or one-one. Use crows-foot notation; if you use another notation, define it. (3 points)**

You only need to turn in one drawing that includes all the elements listed in steps a-e above.



3. **Database. (21 points)**

   a. **List the name and vendor ID of all vendors that supply at least one assembly to MITG. List each vendor only once. (7 points)**

Standard SQL:

SELECT DISTINCT Vendor.VendorName, Vendor.VendorID FROM Vendor, VendorAssembly WHERE Vendor.VendorID= VendorAssembly.VendorID;

MS Access:

SELECT DISTINCT [Vendor].[VendorName], [Vendor].[VendorID]
FROM Vendor INNER JOIN VendorAssembly ON
[Vendor].[VendorID]=[VendorAssembly].[VendorID];

(If you take a strict interpretation of the data model, which says that each vendor supplies one or more assemblies, you can just SELECT from the Vendor table.)

   b. **List the product ID and glider description of all high performance hang gliders with a wholesale sales type. Assume the code for high performance is "H" and for wholesale is "W" (7 points)**

Standard SQL:

```
SELECT Glider.ProductID, Glider.GliderDesc FROM Product, Glider WHERE
Product.ProductID= Glider.ProductID AND Product.SalesType= "W" AND
Glider.GliderType= "H";
```

MS Access:

```
SELECT Glider.ProductID, Glider.GliderDesc
FROM Product INNER JOIN Glider ON Product.ProductID = Glider.ProductID
WHERE (((Product.SalesType)="W") AND ((Glider.GliderType)="H"));
```

c. **List the name and vendor ID of all vendors whose average assembly price is above $50. (7 points)**

Standard SQL:

```
SELECT Vendor.VendorName, Vendor.VendorID
FROM Vendor, VendorAssembly WHERE Vendor.VendorID =
VendorAssembly.VendorID
GROUP BY Vendor.VendorName, Vendor.VendorID
HAVING (AVG(VendorAssembly.VendorAssemblyPrice)>50);
```

MS Access:

```
SELECT Vendor.VendorName, Vendor.VendorID
FROM Vendor INNER JOIN VendorAssembly ON Vendor.VendorID =
VendorAssembly.VendorID
GROUP BY Vendor.VendorName, Vendor.VendorID
HAVING (((Avg(VendorAssembly.VendorAssemblyPrice))>50));
```

d. **Delete all vendors that make assemblies with retail sales type ("R"). Assume that cascade delete is enabled. (7 points)**

Standard SQL:

```
DELETE * FROM Vendor WHERE VendorID IN
(SELECT Vendor.VendorID FROM Vendor, VendorAssembly, Product WHERE
Vendor.VendorID= VendorAssembly.VendorID AND VendorAssembly.ProductID=
Product.ProductID AND Product.SalesType= "R");
```

MS Access:

```
DELETE * FROM Vendor WHERE VendorID IN
(SELECT  Vendor.VendorID
```

```
FROM (Vendor INNER JOIN VendorAssembly ON Vendor.VendorID =
VendorAssembly.VendorID) INNER JOIN Product ON VendorAssembly.ProductID =
Product.ProductID
WHERE (((Product.SalesType)="R")));
```

4. **World Wide Web. (20 points)**

For each of the following questions, give a short answer (2-4 sentences or phrases):

a. Compare and contrast HTML and XML tags. Define each; describe their differences. (4 points)

HTML tags are keywords delimited by <> that direct a browser to display the contents of an HTML page. HTML tags guide browsers in how to display objects, such as text, graphics and applets on the device screen. HTML tags do not define the meaning of any of the items in the browser window.

XML tags are also keyword delimited by <> that define the meaning of the contents of an XML document. XML tags define the content of the document, such as addresses, names, quantities, etc. They do not define how to display these items; XSL and cascading style sheets are used for display.

b. **Who defines HTML tags? How can a new HTML tag be added to the HTML language? (4 points)**

HTML tags are defined by a standard such as HTML 4.0, administered by the WWW Consortium. The tags are fixed; users cannot define new HTML tags. Tags can only be added in new versions of HTML, by agreement of the WWW Consortium.

c. **Who defines XML tags? How can a new XML tag be added? (4 points)**

XML tags are defined by Document Type Definitions (DTDs), which can be created by users, industry groups or standards bodies. XML tags are flexible and can be user-defined. A user may create her own DTDs with her own XML tags. To have these DTDs accepted by other organizations requires acceptance by some informal or formal group.

d. Describe the security provided by JavaScript running on a user's browser:

i. What PC resources can JavaScript access on a user's computer? (3 points)

We didn't cover this in much detail, so only a general discussion is looked for. Early versions of JavaScript allowed substantial access to the user's file system, directory and network. Later versions have tighter security, but also allow signed scripts, similar to signed ActiveX controls, to have fairly full access to files, directories and networks. There is a default restriction on JavaScript network access that allows access only on the same server/host that served the original JavaScript. The point is that JavaScript potentially has substantial access, either through configurations that must be set, or via holes, a number of which have existed.

ii. Describe any potential security problems that JavaScript's access to the user computer may create. (2 points)

JavaScript can obtain directory listings of files on the users' PC, and it can upload the contents of files to a remote site. JavaScript can start new programs (processes) on your PC that take large amounts of CPU time, which is a denial of service attack. JavaScript can open many windows quickly, another form of denial of service attack. JavaScript can connect to remote sites and send files or emails under the client's name.

iii. **Contrast the security level of using XML and XSL (and any related elements such as cascading style sheets) with the security level of using JavaScript on the client computer. (3 points)**

Because XML and XSL (and related tools) are part of the browser and obtained from Microsoft or Netscape, the security of this software is much more assured than from JavaScript downloaded from potentially unknown servers on the Internet. Also, XML, XSL and XLL are more restricted in their capabilities and access to the client machine, which also improves security.

5. **Telecommunications networks. (20 points)**

Seven layer model. Describe the 7 layer ISO model as it is used in the Internet:

a. **Name each of the layers and, in 1 sentence or 1 phrase, define its role. Name and describe the principal (or defining or characteristic) unit of data that is transmitted at that level. You do not need to identify layers that are rarely or never used in the Internet. (5 points)**

1: Physical: Transmits physical signals, coded as <u>bits</u> (0-1) across a network link
2: Data: Transmits <u>frames</u>, or groups of bits across single network links
3: Network: Transmits <u>packets</u>, or quantities of data from origin to destination. Not necessarily reliable: packets may be lost, out of sequence or damaged, etc.
4. Transport: Ensures reliable, sequenced delivery of <u>packets</u> from origin to destination
7: Application: Network communications, such as HTTP GET, POST or responses, used at the browser/server layer to implement Web services

The principal unit of data in each layer is:

1. Physical: bits
2. Data: frames
3. Network: packets
4. Transport: packets (augmented to ensure reliable delivery) or connections
7. Application: messages or transactions, such as GET, POST, responses, etc.

b. Describe, in 2-4 sentences, how a multi-layered model (like the ISO 7 layer model) allows a variety of network technologies to be integrated to provide apparently seamless communications across the Internet. (5 points)

The 7 layer model allows software at each layer to be written as if it can communicate with another system at that layer. Thus, HTTP can communicate to another machine's HTTP as if physical communications are available at layer 7. In fact, physical communications occur only at layer 1, and each intervening layer reads and writes additional data to be able to pack and unpack application (layer 7) data transparently.

This also allows technologies to be substituted or multiple technologies to be used, especially at the lower levels. Thus, many different layer 1 and 2 protocols are used underneath TCP/IP at layers 3 and 4, and HTTP at layer 7. This allows the evolution of network technology while leaving software applications unaffected.

c. Network technology for local loop access between central office and end user: List the data network technologies or protocols that can be supported over copper wires from a central office to an end user (residence or business). List at least 3 options. (5 points)

Dial up modem, digital subscriber line (DSL), ISDN, T-1 over copper, frame relay over copper

d. Local loop, continued: What is the typical bandwidth supported by each of the 3 options? You may provide a general range. Describe briefly how one of the technologies with higher bandwidth achieves its performance. (5 points)

Dialup modem: 56kbps or less. Limited by filters on copper line for voice use that allow only 4kHz bandwidth on line.

DSL: Up to 7Mbps. Filters for high frequencies are removed, so a series of 4kHZ frequency bands are used beyond the 0-4kHz voice band to transmit data. Frequencies of 1MHz and beyond can be used on copper loops of 18000 feet or less.

ISDN, T-1, frame relay: All provide up to 1.5Mbps on copper using similar technology.