Fundamentals of Web Programming [a]

Class Management

Teodor Rus

`rus@cs.uiowa.edu`

The University of Iowa, Department of Computer Science

Contents

- The goal of the class;
- Topics to be covered in this class;
- Rationale for the chosen topics;
- Student expectations;
- Instructor expectations;
- Student assessment.

Goal of this Class

The goal of this class is to expose students to the increasing trend of computer use in the process of service oriented problem solving, which consists of:

- Formulate your problem;
- Express it into a language available to the computer (note, not necessary to your own computer);
- Send the problem to be solved to the network of computers where a computer to solve it can be found;
- Collect the result and use it as appropriately!

Questions

- What knowledge do we need in oder to use this approach?
- Where is Web programming in here?
- Why is this approach different?
- When did it start and where does it go?

Why is this approach different?

Problem solving appear as a service not as an own activity!

- With conventional CBPSP computer is the problem solving tool and problem solving boils down to Computer Programming.
- With service oriented problem solving the computer is a node in a computer network and problem solving activity consists of searching for a service that can solve it.
- Problem solving services are businesses run on computer networks! Problem solving is now performed by targeting such businesses to the problem domain!

Example: buying an airplane ticket!

The Web

- In 1989, Berners Lee developed a proposal to build a "Hypertext project" called "WorldWideWeb" ("W3") as a "web of nodes" with "hypertext documents" to store data.
- The nodes would be computers and the data would be viewed in "hypertext pages" (web-pages) by various "browsers" using an "access protocol" connecting computers on the “web of nodes".
- The project was targeted to the two computer networks, "Internet" and "DECnet", in existence at that time.

Web Evolved!

- The computer network supporting the Web became a collection of computer networks that can access each other and can communicate using various communication protocols.
- Web pages are still viewed as hypertext but the markup is no longer standardized, it can be user defined;
- The data represented on the web has two main attributes: it has a presentation form, the syntax, and a contents, the semantics;
- Computer interaction is no longer restricted to exchange info only; computers can perform computation tasks on behalf of each other.

Thus, computers on the Web are clients and servers in a service oriented business which seems to grow in an endless competition.

Programming The Web

So, what does Web Programming mean?

It means the design of the software that allows computer user to see the Web as a “computer society" where:

- Computers can identify each other "by name" and "by value";
- Computers can communicate with each other using standardized communication protocols (communication languages);
- Computers can interact with each other according to computation needs required by computer based problem solving process.
- Computer user does not program her computer, rather she interacts with her computer using a high-level interaction protocol (language) appropriate to her problem solving process.

Fundamental knowledge

The fundamental knowledge needed for Web Programming:

- Computer Networks: how are they designed and managed ? Open System Interconnect (OSI) Protocols;
- Computer Processes and Computer Process Interactions. Client/Server Paradigm of Computer Programming;
- Sockets: the mechanisms for computer network communication;
- High level computer protocols (communication languages), HyperText Transport Protocol, HTTP;
- Client programming languages: XHTML, XML, JavaScript, and Document Object Model (DOM);
- Server programming language: XHTML, XML, PHP, and Document Object Model (DOM).

Where does it come from?

Computer has been conceived as a problem solving tool!

So, as any human-designed tools computer evolved from:

1. Special tool designed and used by special people! (Original computers, ENIAC) to:
2. Special tool provided by special people to other people (IBM mainframes) to:
3. Special tool dedicated to people (PC, Desktop, Laptop, etc) to:
4. Anonymous tool used by anybody who need it to solve his/her problems (a node in a computer network).

Where does it go?

- Computer as anonymous tool (node in a network) is like any other human tool.
- The essential difference is that computer is a brain dedicated tool.
- Consequently, computer evolution and use are unlimited:

  grow with human ability to think and evolve.

CS Hot Problems

Computer science and computer programming will never disappear!

- CS will be there as long as computers will be there;
- Computers will be there as long as people will be there!!!

However, while fundamental problems with computer programming remain:

- new problems raised by computer evolution as a problem solving tool keep occurring;
- focus of Computer Science keep changing, accordingly!!!

Current Web Hot Problems

- Asynchronous Client/Server Communication on the Web (Ajax);
- Database use and management on the web (Tomcats);
- Service Oriented Architectures (SOA):
  - Small Object Access Protocol (SOAP),
  - Web Service Description Languages (WSDL),
  - Universal Description, Discovery, and Integration (UDDI).
- Domain dedicated problem solving environments (XMLRPC)
- Software dedicated to non computer-expert computer users!

To approach these we need to identify and handle fundamentals of web programming!

Topics To Be Covered

1. Computer Networks: OSI Protocols (%5);
2. Computer Network Process Management (%5);
3. Sockets: Client/Server Process Interaction Using Sockets (%10);
4. HyperText Transfer Protocol, HTTP (%10);
5. Client Programming: XHTML, XML, JavaScript, and DOM (%25);
6. Server Programming: XHTML, XML, PHP, and Ajax (%25)
7. Applications (%20)

See class web-site *(Topics to be covered during Fall11)* for a detailed schedule of the topics to be discussed in this class.

Teaching/Learning Support

There is no textbook that would cover all the material I plan on covering in this course. Therefore, I will relay on my slides which will be made available to the students attending the class.

However:

The textbook I have chosen for this class R.W.Sebesta, *Programming the World Wide Web*, Addison-Wesley 2010, is probably unique with respect to the range of material it covers.

I advise you to get a copy and start reading it.

Note: The projects you need to choose in order to get a high grade in this class can be selected from the exercise published at the end of each chapter of this book.

Student Expectation

What can a student expect from this class?

Depending upon student involvement in class teaching/learning the answer would be:

- Get a global picture of the Web and its usage in problem solving process, $WWW = \langle ComputerNetwork \oplus InfoFlow \rangle$;
- Understand what a computer network is and how computers interact on it;
- Learn to develop client programs using XHTML, XML, JavaScript, and DOM;
- Learn to develop server programs using XHTML, XML, PHP, and Ajax.

More Student Expectations

Go beyond all what is usually done, to:

Use the Web for the Development of the

Web Based Problem Solving Methodology

which is also known as the

Service Oriented Architecture, (SOA).

Job Hunting Power

Plunge into the myriad of Service Oriented Problem Solving aspects that provide basis for:

Job Hunting on Computer Based Job Market.

Instructor Expectation

This class is at its third edition! Based on the experience of its previous editions:

- I expect students to like it and get involved;
- I improved the mechanism used to assess student knowledge acquired in this class, and I do expect even better results;
- Based on student suggestions I propose the following approach for student grading:

Student Assessment

1. Class attendance (no class missing) 20% (a D);
2. (1) above and a self-selected project showing ability to handle sockets programming 40% (a good C) (due 1 October or before)
3. (1), (2) above and two client-based applications chosen from the textbook chapters 1 through 8, 65% (a B) (due 25 Oct. or before)
4. (1), (2), (3) above and two server-based applications self-chosen from the textbook chapters 9 through 15, 90% (an A-) (due 15 November or before)
5. (1), (2), (3), (4) above (90 points) plus a self selected application project 100% (due 6 December or before).

Attendees

This class is very heterogeneous: attendees expand on a large spectrum of knowledge and interests (from undergraduate to PhD).

Question: could my assessment suggestion be appropriate for this class?

My Answer: no, because it violates class principle:

accessible, fun, and useful to every attendee!

Amending the assessment

Students can replace the assessment requirements by special projects they can perform during this class, such as:

1. Present one or more lectures on topics of interest but not scheduled for this offering of the class.
2. Chose to demonstrate Web programming by performing special project’s. Example implementing and demonstrating Ajax.
3. Chose to demonstrate research on Web programming by approaching and solving various hot problems.

   Examples: Airline Flight Control, Anti-Money Laundering, Event-Driven Productivity Infrastructure. (See instructor for reference).
4. Using Service Oriented Architectures to perform Computational Emancipation of Application Domains (CEAD)!

Assessment Wavers

Students can get individual class assessment wavers using the following procedure:

1. Prepare a written document where the student proposes a project (among those enumerated above or another project suggested by the student).
2. Provide a complete description of the work to be performed specifying the amount of grading credits claimed for this work;
3. Provide a schedule of the work to be done proposing the validation method at each step of the schedule;
4. Submit this document to the instructor.
   Due date: 15 September 2009

Instructor will study this document and will approve/disapprove it as appropriate.

Class Policy

All due dates are strict!

If you miss a due date you can go for the next;

but one missed due date is one lost due date!

Other Policies

See syllabus for links to College of Liberal Arts and Sciences (CLAS) policies pertinent to this class.

Good Luck!