Teaching Internationalization — Internationally

Jesse M. Heines
Dept. of Computer Science
Univ. of Massachusetts Lowell
Lowell, MA 01854 USA
+1 978 934 3634
heines@cs.uml.edu

Krzysztof Jassem
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
61-614 Poznań, Poland
+48 606 933 392
jassem@amu.edu.pl

ABSTRACT
This paper describes a foray into teaching internationalization by attempting to do a collaborative project between students in the United States and Poland. The project required Polish students to work with software developed by American students and to provide feedback to the Americans on how easy it was to understand and modify their code. Students communicated via email and online chats as well as in a live session facilitated by Google Hangout. The goals were to get students in both countries to appreciate the clarity needed to communicate and work with international colleagues and to have them experience the myriad issues involved in such collaborations. We report the details of the project we assigned, the processes we went through to set up the collaboration, and our successes and failures as we worked toward our goals.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education — *computer science education, curriculum.*

General Terms
Design, Human Factors, Standardization.

Keywords
Internationalization, JSON, GUI programming.

1. WHY TEACH INTERNATIONALIZATION?
Teaching programming techniques is easy. Our students are smart, they've come up through a well-structured curriculum, and most of them, at least, are willing to a little work. It is easy to convince them that a quicksort is faster than a bubble sort on any but the smallest data sets, and even easier — in the age of Google — that it's important to study database principles and search algorithms, to name just two examples. By the time they get to our upper-level classes, our students also have experience in using myriad tools to help solve problems themselves.

Teaching supporting topics, however, is far more difficult. For example, it is hard to teach software engineering to people who have never written a program longer than a few hundred lines of code. It is particularly difficult to convince students of the importance of documentation when they've never worked with someone else's code or code that they themselves wrote just a few months ago, much less 5-10 years ago. And it is virtually impossible to get them to see the value of internationalization when the only people they interact with are members of their own peer group.

Yet in today's global economy, internationalization is critical [2]. Students must learn that if their work is to be used in other countries, where there may be significant untapped markets, that work must

- follow international standards,
- be engineered to be adaptable, and
- be structured to allow easy customization (especially of all text messages).

A simple example involves how dates are written. In most of the world, dates are written in day→month→year format, such as "1 July 2013." In the United States, however, dates are typically written in month→day→year format, such as July 1, 2013. This may not seem like a big deal, but it becomes critically important when dates are abbreviated. For example, does 7/1/13 mean "1 July 2013" or "7 January 2013"? ISO8601 dates resolve the ambiguity by standardizing on YYYY-MM-DD format, but most programmers want their user interfaces (UIs) to be more user-friendly than to display dates as 2013-07-01.

Numerous other UI-related internationalization examples exist, from alternate spellings ("colour" vs. "color") to units of measure. (English vs. metric systems). But these considerations are not limited to the UI [1]. Backend code must also adhere to the principles embodied in the three bullets listed above, as programmers in other countries may have to adapt that code to make the program suitable for their clientele. So how can we motivate students to adhere to these principles?

2. SEEING ONE'S WORK FROM ANOTHER'S PERSPECTIVE
While authors readily cite the importance of teaching internationalization [3, 4], it's hard to convince students in a vacuum. No matter how hard one tries, *talking* about internationalization will not suffice. Students have to experience first-hand what happens when people unlike themselves try to use their programs.

This has long been done through usability testing, which is an invaluable teaching tool. When Student A tries to use Student B's program and encounters difficulties, B will listen far more intently to A's concerns and take A's suggestions for improvement far more seriously than if those same concerns and suggestions are expressed by the professor.

When it comes to internationalization, students must likewise communicate directly with people across the international divide to truly see their work from another perspective. Only then will they fully appreciate and really understand the importance of internationalization and begin to perceive ways to achieve it.

ITiCSE'13, July 1-3, 2013, Canterbury, England, U.K.

3. FINDING AN APPROPRIATE TOPIC FOR STUDENT COLLABORATION

We began our collaboration by discussing our respective classes and exploring ways in which we might have our students collaborate on some sort of joint project. We looked at each other's syllabi and assignments, searching for an area that "clicked" and that we could adapt to an international project.

In the fall 2012 semester Jesse was teaching a course in graphical user interface (GUI) programming, while Krzysztof was teaching one in software engineering. Jesse's course stressed data-driven programming techniques that allowed user interfaces (UIs) to be easily adapted and modified for internationalization. Toward this end, students wrote programs that stored all text in external JSON files rather than being hard-coded into embedded strings [5]. They designed and implemented web page layouts that were liquid and flexible, that is, with areas that could shrink and expand to accommodate whatever they needed to display, and they populated these areas using jQuery to process the JSON data.

The coding part of this approach was no problem. Illustrating its power, however, was more difficult. The class included students who spoke languages other than English, but almost all were reluctant to use those language skills in class.

In Krzysztof's course, students were also doing web-based projects, but they had not worked with JSON. Krzysztof was interested in adding this element to his course, but the geographic distance and six-hour time difference between our two locations seemed to prohibit a concurrent approach in which our students could actually work together.

We therefore devised a serial approach in which Jesse's students developed web pages with internationalization in mind and Krzysztof's students modified those pages to see how well the designs and code held up when, for example, Polish text was substituted for the English text and the page's various graphic logos and/or icons were replaced with ones suitable for Polish audiences. Other assignments, more specific to international collaboration, might have been devised, but the ones described below fit into our existing courses and were sufficient to get collaboration started.

3.1 The Assignment for American Students

The essence of the assignment made in Jesse's GUI Programming course was as follows. (To see the full assignment, please visit teaching.cs.uml.edu/~heines/91.461/91.461-2012-13f/461-assn/UsingGraphics-v02.jsp.)

What This Assignment Is About

Create a web page (or a short series of web pages) that introduces a computer science course that you have taken at UMass Lowell. The target audience for your page is to be other students who have not yet taken the course. The overall concept is to create a page that will be linked from our department's course listing page to provide information about each course being offered.

What You Are To Do

1. Work with your partner to design your overall page.
   - Use the UMass Lowell logo and other graphics to give it an official look-and-feel.
2. Describe what the course is about.
   - Look at the course syllabus and use text from that.
3. Introduce the textbook(s).
   - Capture images of the textbook covers and provide links to book's web page on the publishers' website(s).
   - Find pictures of the authors if you can, or perhaps the publishers' logo(s).
   - Add these to your page to enhance its look and to experiment with image placement.
4. Talk about how the class is run.
   - Stick to factual information, but if you express an opinion, please be sure to clearly label it as such and perhaps use some graphical technique to distinguish it.
5. Give a feel for the types and difficulty of assignments, and the amount of time required to complete them.
6. Provide examples of the things one learns.
   - This should make it crystal clear to other students just what the course is about.

Links to students' submissions for this assignment can be found at teaching.cs.uml.edu/~heines/91.461/91.461-2012-13f/461-lecs/lecture18.jsp.

3.2 The Assignment for Polish Students

Once the American students completed their assignments, we posted them at the URL shown above so that the Polish students could begin their work. The essence of their assignment was as follows.

What You Are To Do

1. Select a course in your AMU department to work with.
2. Choose one of the American students' solutions and replace the English text in the JSON file with your own Polish text.
3. Replace graphics on the American students' pages with graphics appropriate to our institution.
4. Check the result for formatting and other issues and report those issues to your American colleagues.
5. Comment on the design of the American students' solutions that you chose to work with and how you might change it and why. If you wouldn't change anything about it, explain why you chose this design to work with over others.
6. Comment on how easy (or difficult) it was to work with the American students' JSON templates and code.
   - Were these software components easy to figure out or would you have benefited from more documentation?
   - Did you find the code well-structured or did you struggle to adapt it for your purposes?

A typical pair of projects is shown in Figures 1a and 1b, the English and Polish versions. Note the proper automatic expansion of the English "Topics Covered" section to accommodate the longer Polish text, but also the anomaly caused by the absolutely positioned graphic such that it obscures part of the longer Polish text. The JSON that populated the page content is shown in Figures 2a and 2b.

4. SETTING UP THE FACE-TO-FACE INTERNATIONAL EXPERIENCE

Despite the time difference between our two locations, we did not want to abandon the goal of having our students actually work together. We therefore executed a multi-phased plan.

4.1 Introductory Lecture

The structure of Krzysztof's course was somewhat different from Jesse's, in that Krzysztof's involved both a lecture and a lab. Krzysztof asked Jesse to lecture to his class on the implementation

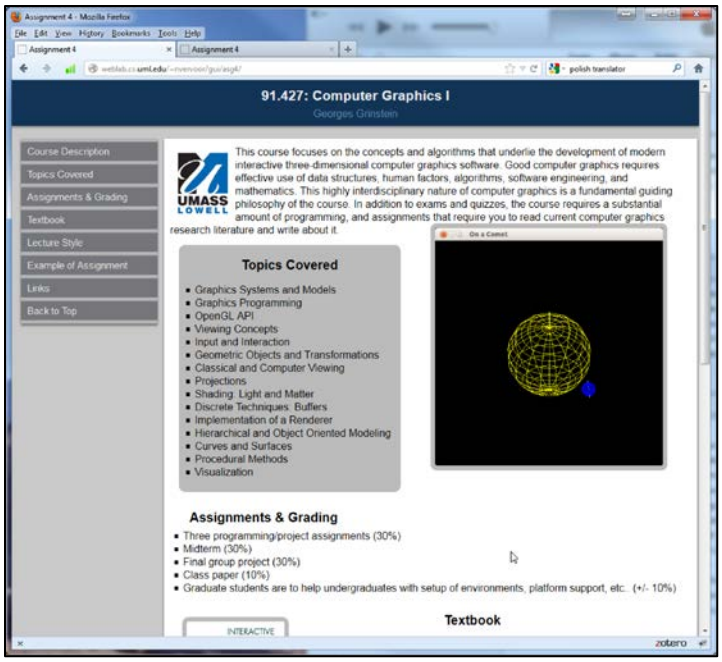

Figure 1a. English page generated from JSON data

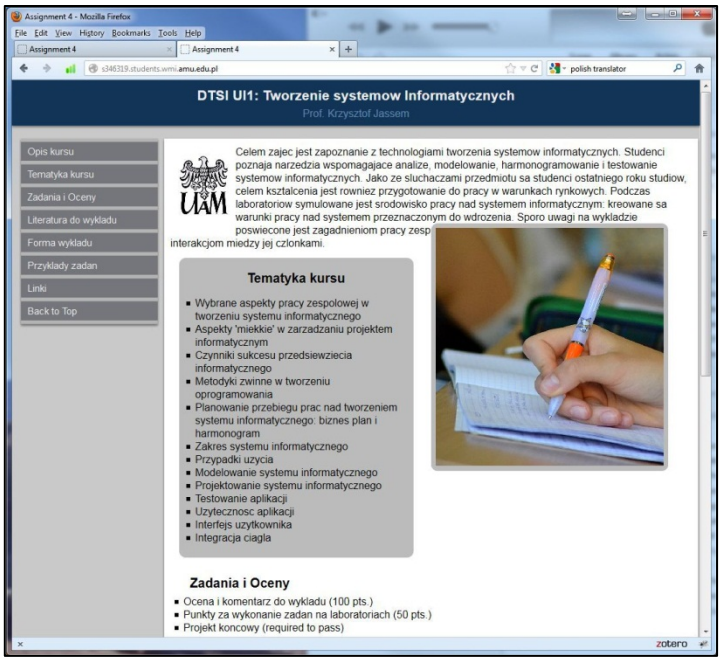

Figure 1b. Polish page generated from JSON data.

```
var root = {
  "title": "Computer Graphics I",
  "instructor": "Georges Grinstein",
  "instructorURL": "http://www.uml.edu/centers/ivpr/Faculty/
Georges_G._Grinstein.html",
  "coursenum": "91.427",
  "prereqs": "91.201 Computing III",
  "credits": "3",
  "description": "This course focuses on the concepts and
algorithms that underlie modern, interactive three-dimensional
computer graphics software. ... ",
  "textbook": {
    "title": "Interactive Computer Graphics: A Top-Down Approach
with Shader-Based OpenGL",
    "edition": "6th",
    "author": "Edward Angel, Dave Shreiner",
    "isbn": "9780132545235"
  },
  "topics": [
    "Graphics Systems and Models",
    "Graphics Programming",
    "OpenGL API",
    "Viewing Concepts",
  ],
  ...
} ;
```

Figure 2a. English JSON data excerpt.

```
var root = {
  "title": "Tworzenie systemow Informatycznych",
  "instructor": "Prof. Krzysztof Jassem",
  "instructorURL": "http://psi.amu.edu.pl/pl/index.php?title=
Krzysztof_Jassem",
  "coursenum": "DTSI UI1",
  "prereqs": "None",
  "credits": "4",
  "description": "Celem zajec jest zapoznanie z technologiami
tworzenia systemow informatycznych. Studenci poznaja narzedzia
wspomagajace analize, modelowanie, harmonogramowanie i
testowanie systemow informatycznych. ... ",
  "textbook": {
    "title": "Analiza i projektowanie systemow informatycznych",
    "edition": "1th",
    "author": "Jacek Plodzien, Ewa Steposz",
    "isbn": "not known"
  },
  "topics": [
    "Wybrane aspekty pracy zespolowej w tworzeniu systemu
informatycznego",
    "Aspekty 'miekkie' w zarzadzaniu projektem informatycznym",
    "Czynniki sukcesu przedsiewziecia informatycznego", ...
  ],
  ...
} ;
```

Figure 2b. Polish JSON data excerpt.

of data-driven UIs and the use of JSON to orient his students to the international collaboration. Jesse agreed to do so, but with the lecture portion of Krzysztof's class scheduled for 10:00 AM Polish time, Jesse had to deliver his lecture at 4:00 AM! The title of this lecture was "Human Factors in Web Programming."

4.2 Student Interaction

After completing his lecture, Jesse headed to his university to meet with eight of his students who had volunteered to come in for breakfast at 7:30 AM and be ready to "hang out" with the Polish students at 8:00 AM Massachusetts time, corresponding to the 2:00 PM meeting time for the Polish students' lab. Many of the 32 students enrolled in Jesse's class work in addition to going to school fulltime, so he did not feel that he could *require* all students to come to school at 7:30 AM for a special class. The eight who volunteered were, as one would expect, some of the best students in the class. Therefore, Jesse felt that having 25% of his students present was a reasonable representation.

Once again we used Google Hangout to connect our two classes. Students communicated both "live" (Figure 3) and via chats (Figure 4), and they exchanged files via email.

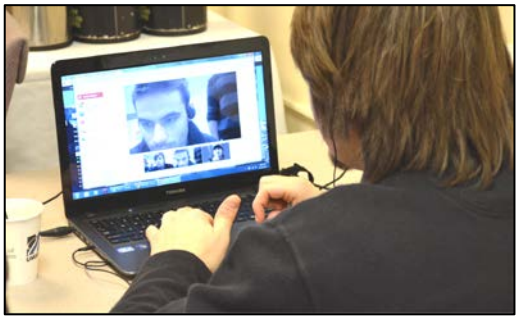

Figure 3. Looking over an American student's shoulder as he communicates with Polish students via Google Hangout.

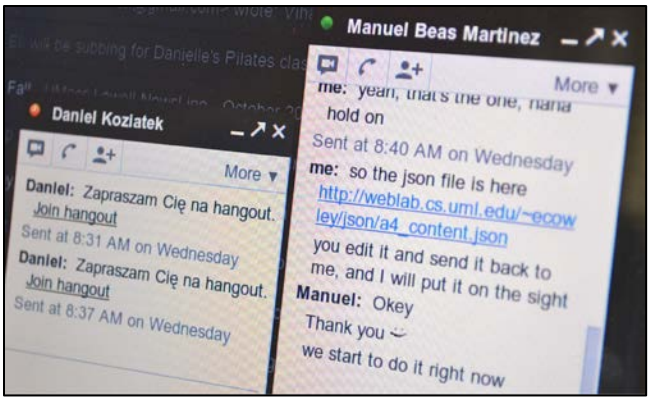

Figure 4. A Google Hangout chat between one American and two Polish students.

4.3 Capturing Student Reactions

WMIpedia, a system developed by students in the Faculty of Mathematics and Computer Science at AMU, played an important role in the exchange of ideas. (WMI is an abbreviation for the Polish name of this Faculty.) The system allows students to expand and publish the knowledge they gain from lectures, as well as to express their opinions.

In a typical scenario, an instructor first posts 1-10 tasks related to a lesson on the system and selects a student in the course as the moderator for this lecture. Other students post their solutions (or reactions or opinions) to the tasks on WMIpedia *anonymously*. The then moderator comments on and grades those posts without knowing their authors. The instructor grades the job done by the moderator and may make the best posts public.

WMIpedia is intended to stimulate students' creativity, help them assimilate knowledge by comparing it to their own experiences, and expand their knowledge through other sources. It also helps moderators learn how to evaluate others' work and to recognize plagiarism. For instructors, the system helps them know if the material was comprehended, which parts were most interesting to students, and which need additional attention in the future.

For this lesson, students were asked to comment (in English) on the introductory lecture and on the follow-up class in which they interacted with the American students. The specific tasks were:

1. Grade the lecture on a scale of 1-10.
2. Comment on both the lecture and the class.
3. Present some examples of applications that do or do not take human factors into account.
4. Prepare your own brief course on JSON and parsing JSON in JavaScript. An example will serve better than theory.

Representative student comments on the lecture and class (Task 2) are presented in the next section.

5. RESULTS

5.1 From the Professors' Perspective

From our perspective we felt that things went well. However, as one would expect, the experience was not without glitches. We had tested the Google Hangout connection from Jesse's home to Krzysztof's lecture hall a week before the lecture and it worked fine. We had no trouble connecting for the lecture a week later, but the connection from Jesse's classroom to Krzysztof's lab later in the morning was a different story. First, students had trouble connecting via Google Hangout due to two problems.

1. Hangout initiators had to know other's email addresses to be able to "invite" them into a hangout. Jesse had posted the addresses of his students, but since the Americans initiated the hangouts, this was useless. The Polish students solved the problem themselves by emailing the Americans, who then initiated new hangouts and invited the Poles in.
2. The wireless bandwidth for the eight American students in a single classroom was fine, but it was insufficient on the Polish side to support the number of students there who were trying to establish video connections, even though they were in three separate labs. The connections kept getting dropped, resulting in the frustrating need to reestablish them before the collaborative work could proceed. Again, the students solved this problem themselves by reverting to text chats.

The second problem involved file formats. The web server we used was at UMass Lowell. This meant that the Polish students had to send their revised HTML and JSON files to their American counterparts for posting. The issue is that Polish requires the Unicode character set, but for some reason the text was being converted to ANSI when it was emailed. For the third time, the students solved this problem among themselves by ensuring that the Polish students saved their files in UTF-8 format and then sent them is such a way that the original file format was retained.

From our perspective, these are exactly the types of issues that we had hoped the students would encounter, giving them a truly international experience. We were delighted that the students managed to work out solutions to the problems themselves. More than one of the American students commented that they never would have had such experiences without connecting to their Polish counterparts "live."

5.2 From the Students' Perspective

It is clear from transcripts of the students' chats that those in both countries enjoyed and learned from the experience. Below are two excerpts, where P and A identify posts by Polish and American students, respectively (grammar edited slightly by the authors).

Chat #1

P:	*I'm so happy that we can talk.*
A:	Yes, me too. It looks like everything worked great!
P:	*Yes, but there are not Polish characters. You should change for UTF-8.*
A:	The page is set up for that — I think it's a problem with the web server.
P:	*No, we will send you again.*
A:	Could you try saving the file with Unicode encoding?
P:	*We've made mistake. Now it will be better.*
A:	Great. All of the characters display correctly now — I think they saved it as you said, using UTF-8.
P:	*It's okay, awesome. We have to go now — next lecture is going right now. So, we will talk to you later.*
A:	Ok, it was good talking to you!

Chat #2

P:	*We have just sent you a JSON file from* [name suppressed] *email. Could you check if it works?*
A:	Oh, I found it. It was in my Spam folder.
P:	*We find your code amazing. Unfortunately your codepage doesn't work with multilingual documents.*
A:	Yes, I need to fix that.
P:	*We can send you UTF-8 version, since our text editor is set to save in ANSI by default which isn't too good.*

A: Either works. I think my webpage is just not telling the web browser the content type.

P: *We sent you upgraded version of our first JSON file (Polish characters should now work correctly) and other JSON file with information on the project we mentioned. Could you check on our English and upgrade it accordingly if there are mistakes?*

A: Yes, I can.

P: *Since we get our grade basing on the validity of the links to your site, please don't delete it too fast.*

A: OK, I won't.

P: *Thanks m8!*

A: Looks much better now. I am going to add a few features to my JavaScript to make your midterm a better user experience and hopefully give you a better grade

P: *Thank you!*

A: No problem.

P: *Thanks for cooperation, we are going to dinner! Bye.*

A: You're welcome. Bye.

Informal comments from the American students after the experience underscored their enjoyment and its educational value. One simply said that the experience was "awesome," and all of the American students said that they were very impressed with the Polish students' knowledge of English.

From a technical perspective, the following Polish student's responses to the questions in their assignment were typical. Like the chats, they show both learning and enjoyment.

A Polish Student's Responses to the Assignment Questions

4) Check the result for formatting errors and report the errors to the American colleagues.
We had some troubles while formatting [the] JSON file. Luckily, we managed to get it working. The problem stemmed from the lack of a comma. :)

5) Comment on the design of the American students' solutions that you chose to work with and how you might change it and why. If you wouldn't change anything about it, explain why you chose this design over other designs.
Our choice was dictated by the structure of the JSON file. [Name suppressed]'s JSON file is well written and easy to work with. We had no problem understanding it. As for graphical design, we would change floating side menu — it overlaps other elements on a 15-inch screen.

6) Comment on how easy it was to work with the American students' JSON template and code:
– Were these software components easy to figure out or would you have benefited from more documentation?
It was quite easy to figure out what each component [does] and how to use it. JavaScript code is well commented and JSON file is easy to understand.

– Did you find the code well-structured or did you struggle to adapt it for your purposes?
As mentioned above, the code is very well-structured and we had almost no problem to adapt it for our task. We had some difficulties because of missing commas or brackets, but it wasn't [the] fault of the code.

9) Ask American colleagues to verify the correctness of your English.
[Name suppressed] said our English is quite good. Unluckily we didn't connect [video] directly with [her] during the class because of the connection problems.

5.3 WMIpedia Comments

The Polish students' anonymous posts on WMIpedia revealed a fascinating scope and variety of opinions. The bulleted items below are students' actual comments, edited slightly for grammar.

5.3.1 On the Lecture

The overall opinion on the lecture was positive.

- In the beginning, [it was] hard to understand what was hidden behind the words "human factor." But when we got into it, I must say it was [a] really interesting point of view on web programming.
- Amazing accent and no technical problem. I am [a] Mr. Heines fan!

The students appreciated the high standard of the presentation. They stressed the Professor's attention to being understood by non-native students.

- Prof. Heines's presentation was very straightforward and presented in accessible manner.
- First of all, he speaks very clearly so the language barrier disappeared — it was very important to me.

What the students liked most was the new experience.

- Online transmission of lecture was something new here.
- I found this lecture great! … Every time we see something new, it's interesting, stunning and amazing. Well, it was this time as well.

Students liked the idea of cultural interchange. They realized that there are different styles of lecturing in different countries: lectures are much more interactive in America.

- This amazing American style of teaching made it interesting.
- I really appreciate [him] waking up so early just to say some words to Polish students (who are very hard to work with and it's almost impossible to communicate with them).

5.3.2 On the Class

The overall impression of the class was considerably worse than that of the lecture, but like the Polish student's responses to the assignment questions quoted above, there were some positive reactions.

- Class was a total disaster. We were supposed to [connect] with other students from USA. None of this worked. Exercises made no sense. Not clear enough. Templates made by these USA students were not complete at all.
- Class exercise dominated by communication problems. It should have been a little bit better prepared. Some of us were testing a Google hangout service about hour before class and we had detect communication problem with this service (huge delay, maybe caused by WMI network), many of us ended up on chats only, some students were kicked out of WMI local network during process. I personally believe there should be designed a "backup" solution in case of such tech problems occur before class starts.
- Class with assignments was a little bit unsuccessful — but still, a good idea. Cooperating with other students and their code this way was nice exercise.

- As for class, I also liked it very much. Our task was simple, yet it required from us to do some work and research. Unluckily our group didn't have a chance to talk with American students. All in all — our last class was great!

The students realized the merit of international cooperation.

- More important part was just to try working together from the distance.
- Cooperation with [name suppressed] during the class was perfect.

The students found the exercises simple, too simple:

- The exercise was just pointless (replace English text with Polish text).

5.3.3 Conclusions from WMIpedia Comments

1. It is very important and appreciated by students that the lecturer speaks clear English directed to non-native students.
2. All technical details of the international lecture should be prepared beforehand.
3. A lecture given by a foreign professor makes students aware of cultural differences.
4. Conducting an international class is a risky exercise. It should be tested beforehand with volunteers. Connection issues are the most important to test, as they frustrate students very quickly.
5. The advancement level between both groups should be matched as closely as possible. Otherwise, at least one side will find the assignments either too easy or too difficult.
6. Students expect "hard results" of cooperation, that is, a visible improvement to what can be achieved "inner"-nationally.

6. OVERALL CONCLUSIONS

As educators trying to give students a true experience in internationalization, here's what we learned.

1. Do not be overly ambitious, especially in the beginning. There are lots of kinks to work out. It's fine to "start small," and it's more important to get something going than to get something amazing going. Students can learn very valuable lessons from seemingly basic experiences.
2. Start early. Regardless of the sophistication of your technology and the quality of your support, live international connections are fraught with obstacles. As noted in the previous section, connections must be fully tested before expecting students to use them.
3. Plan an activity that will have a life beyond the immediacy of the live connection experience. In our case, some students continued to communicate with each other after the Google Hangout hookup to complete their projects together. This enriched the experience for all involved.
4. Try to schedule the live connections session so that students can continue working together immediately after the formal class has concluded. 60 or even 90 minutes goes by far too quickly, especially when time is lost to connection problems.
5. Among the three free collaboration programs that we considered using — Google Hangout, Skype, and TeamViewer — we found that Google Hangout provides the best online experience for lecture presentation because it allows sharing a single window rather than an entire screen and simultaneous video connections between multiple users.
6. Offer free food to students and a reasonable number will volunteer to participate at almost any time of day! ☺

All in all, we felt that our first foray into teaching internationalization was successful and that students both enjoyed and learned from the experience we set up. In future collaborations, we would like to devise methods to gather quantitative data on the students' experiences to help us understand how their behavior might have changed due to these experiences.

7. ACKNOWLEDGMENTS

The seed for our collaboration was an exchange program between our universities under which Jesse was given the opportunity to spend two weeks at Adam Mickiewicz University jointly sponsored by UMass Lowell Provost Ahmed Abdelal and AMU Vice Rector Jacek Witkoś. This gave Jesse the opportunity to meet professors and students at AMU, observe classes there, and make presentations to students and faculty on his research and his teaching. Jesse and Krzysztof met toward the end of Jesse's stay, but that was enough to make a connection and open communication channels on potential future collaboration before Jesse's visit ended.

8. REFERENCES

[1] Kersten, G.E., Kersten, M.A., & Rakowski, W.M. (2002). *Software and Culture: Beyond the Internationalization of the Interface.* Jrnl. of Global Information Management **10**(4):86-101.

[2] Vu, J.H. (2010). *Software Internationalization: A Framework Validated Against Industry Requirements for Computer Science and Software Engineering Programs*. M.S. Thesis in Computer Science, March 2010. digitalcommons.calpoly.edu /cgi/viewcontent.cgi?article=1261&context=theses *accessed* Dec. 29, 2012.

[3] Welzer, T., Riaño, D., Brumen, B., & Družovec, M. (2004). *Internationalization Content in Intelligent Systems — How to Teach it?* In *Knowledge-Based Intelligent Information and Engineering Systems*, pp 1039-1044.

[4] Welzer, T., Golob, I., Druzovec, M., & Kamisalic, A. (2005). *Internationalization as a Part of the Database Development*. Proc. of the 3rd IEEE Int'l. Conf. on Computational Cybernetics, pp. 145-147. Mauritius.

[5] Xiaoyin, W., Lu, Z., Tao, X., Hong, M., & Jiasu, S. (2009). *Locating Need-to-Translate Constant Strings for Software Internationalization*. Proc. of the 31st Int'l. Conf. on Software Engineering, pp. 353-363. Vancouver, Canada.