

# Bridges to computer architecture education

Peter Marwedel, Birgit Sirocic  
Dept. of Computer Science,  
University of Dortmund,  
44221 Dortmund, Germany  
{peter.marwedel,birgit.sirocic}@udo.edu

## Abstract

*Bridging the gap between the student's current knowledge and the technical details of computer systems is frequently required for the education of undergraduate students or students with a lack of previous technical knowledge. In this paper we will describe how to build bridges by the combination of introducing Flash-animations<sup>1</sup> and the educational units of the RaVi system<sup>2</sup>. In a first step the Flash-based animations make the students familiar with the underlying principles. After that the student could jump to the more technical context by using the educational unit of RaVi. We have developed two bridges, one for explaining the principles of cache coherency protocols and the other for showing the concept of processor pipelines.*

## 1 Introduction

Nowadays, many students are looking at computer science from an application perspective. Learning how to use the computer and to design new applications is very attractive. Learning why and how computers are functioning seems to be a lot less appealing for many students. Moreover, it is our experience that some kind of bridge has to be provided for students to learn how computers are working. It is not sufficient to start right away with technical terms and issues. There seem to be two reasons for this: missing prerequisites and lack of motivation. This problem is amplified in classes with

students with very heterogeneous backgrounds.

In the context of this situation, it is unacceptable to just give up and accept that students just focus on the application perspective. A serious lack of knowledge of fundamental principles would be the result and students would not really be able to understand the principles of the equipment they are using in their professional career. This could very easily lead to misconceptions and damages.

So, the issue is: how to motivate the students to learn about the internals of computers and how to provide the bridges that are required for the students to enter areas such as computer architecture?

One of the approaches for removing barriers that prevent students from entering certain areas is to write very motivating books. However, this approach does not really fully utilize the opportunities that exist due to modern media. Modern media can provide a very motivating path from the student's current knowledge to what the student should know about operating principles of computer systems.

In this paper, we will describe how modern media can be used to motivate students to study the operating principles of modern computer systems and how to use modern media to bridge the gap between the student's current knowledge and the technical details of computer systems.

This paper is structured as follows: in section 2, we describe related work. In section 3, we present our "bridge" for understanding the MESI protocol for multiprocessor caches. In the following section, we show how processor pipelines can be introduced to an audience that has limited previous knowledge about processors and a limited motivation. A short description of the didactical background you will find in section 5. Experiences and conclusions for our future work are explained in section 6. The final section provides a summary of the current paper.

---

<sup>1</sup>We acknowledge the contribution of Alexandra Nolte who created the artwork of Flash-based animations with a great level of dedication and artistic professionalism.

<sup>2</sup>We gratefully acknowledge the funding of the RaVi-project (which is a subproject of the SIMBA-project) by the German ministry of research and development (BMBF). The RaVi system is a set of interactive training components for the teaching of the computer architecture. Each component visualizes an aspect of the dynamic behaviour that is found in computer architectures.

## 2 Related work

The fact that bridges need to be provided for the students to look at the details of computer systems with some level of dedication has already been realized by a number of educators.

For example, this view is dominating for the book "Computer systems - A programmer's perspective" by Bryant and Hallaron [1]. In order to motivate students to look at caches, for example, the topic is introduced using the effect of caches on the performance to different versions of the same algorithm. In this context, it is remarkable that the author's University (Carnegie Mellon University) managed to significantly increase the percentage of female students (if compared to male students). Female students are said to focus more on applications of computers and less so on their internals. Unfortunately, the approach of Bryant et al. is totally restricted to the use of classical media, such as books.

One of the inherent limitations of books is their inability to visualize system dynamics. Moreover, they hardly provide any interactivity apart from adding marks to the pages. In order to remove these limitations, simulation of computer systems has been used. In order to be useful in a classroom, simulation-based educational units have to be designed for this purpose. Otherwise, complex menus and high graphical resolution requirements would make it too difficult to understand the units quickly. Simulators in this class include HADES [4], Ptolemy II [3] and the RaVi simulator [7]. The focus of RaVi is on the visualization of complex system behavior which cannot be explained with the animation capabilities of Powerpoint and similar tools. RaVi includes educational units for visualizing the dynamics of microprogrammed MIPS-machines, pipelined MIPS-machines, dynamic out-of-order instruction scheduling techniques and cache protocols. According to our experience, RaVi meets its goal. There is a corresponding large number of downloads of RaVi every month and according to our statistics, RaVi is used throughout the world. An inherent limitation of RaVi is that RaVi does not really provide bridges that bridge the gap between the student's knowledge and the details of computer architectures.

The goal of this paper is to describe how the two approaches just mentioned can be combined.

## 3 A bridge for the MESI multiprocessor cache coherency protocol

The first bridge described in this paper addresses problems in explaining the principles of cache coherency protocols with RaVi educational units. The units pro-

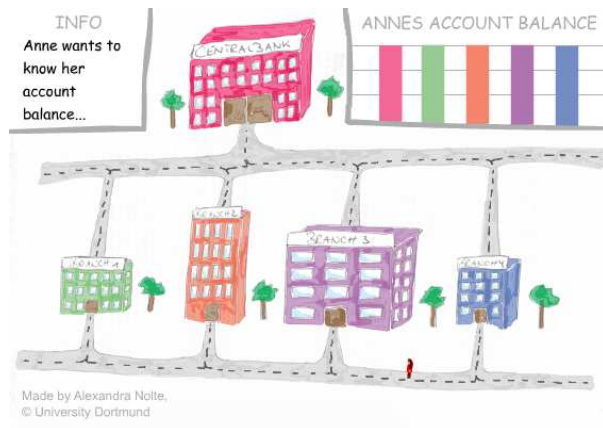
vide a detailed simulation of four caches and processors connected to one "main" memory. According to our experience, this is already a too detailed view for many classes of attendees, including colleagues from other areas within computer science. We found it useful to avoid this problem by relating the technical details to real life. For the multiprocessor cache, we choose a banking system as an example. In this example, there is a bank that has a number of branches (which will later correspond to processors). Customer requests at the branches include obtaining statements from the bank, withdrawals, and deposits. Initially, local branches are not allowed to have local copies of the account information. Accordingly, each of the customer service requests results in sending a messenger to the central bank in order to check and update account information. Obviously, this approach is very slow and it makes sense to allow local copies of account information. With such local copies (later called caches) customers using only one branch are serviced in a short amount of time. For customers moving to different branches, messages still need to be exchanged. Invalidate and well as update messages are required. For each of the accounts, four different states can be distinguished: modified, shared, exclusive and invalid. The meaning of these four states can typically be understood even by the non-specialists. Problems which exist for multiprocessor caches can be introduced very easily. For example, the problem of quickly invalidating local copies of account information in the local branches in order to avoid people getting too much money is very obvious. Also, the problem of identifying the branch that has the up-to-date account information while being in the exclusive state can also be demonstrated easily.

This example has been implemented in a Flash-based animation. The level of interactivity in these animations is sufficient for the "bridges". It would be less appropriate for the MESI educational units, as the behavior of the protocol is quite complex and difficult to implement in Flash. Fig. 1 shows a screenshot of the corresponding Flash animation.

In the actual presentation, the correspondence between the central bank and the main memory as well as the correspondence between the local branches and the local processor/cache systems are explained next. This is quite easy to understand even for the non-specialist.

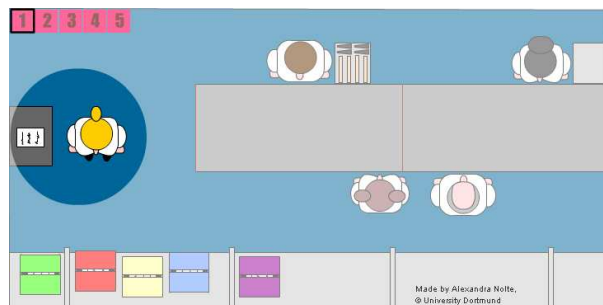
## 4 A bridge for processor pipelines

For processor pipelines, it is possible to use a similar case from real life. The corresponding Flash-animation is based on a production pipeline and uses pipe stages similar to those that are found in MIPS pipelines. The first stage is the dispatch stage. Instructions for fabri-



**Figure 1. Flash-animation for the cache coherency protocols (segment of a screen-shot)**

cating hammers are issued in this stage. Components required for this fabrication are fetched in the second stage. The third stage is the execution stage. This is where the hammers are actually made. The fourth stage is idle as long as only hammers are made. The fifth stage is the storage stage. Hammers are stored in this stage. It is important to show that these stages operate concurrently. Fig. 2 shows a screenshot of this Flash-animation.



**Figure 2. Flash-animation for processor pipelines (segment of a screenshot)**

In the actual presentation, the correspondence between the pipe stages and MIPS pipe stages are explained next. Again, it is quite easy to understand even for the non-specialist.

## 5 Didactical Background

The underlying didactical method which we are realising by the Flash-based animations, is a kind of anchored

learning [2]. The method of the anchored learning is based on the idea, that the learner can find a link from his already existing understanding of the real-world to the new information and concepts by providing them a so called anchor. The learners who find the link, can construct their own new knowledge base and the understanding of the new main principles will be easier.

Our Flash-animations are providing a kind of an anchor by highlighting the analogy of the every-day example and the technical example. That means, the students who know the main principle of a production pipeline can find the link to the main principle of the processor pipelines. For those students who do not know how a production pipeline works, the anchor does not archive its aim.

A problem for implementing the anchored learning is that an adequate anchor has to be provided. That means, the analogy of both examples have to match every aspect of the examples. There is a high risk for misunderstandings or missing a key point, if one or more aspects do not match.

## 6 Experiences and future work

Both bridges described above have been used in presentations. The pipeline was used in a presentation to sophomore students having no previous knowledge of computer architecture. Feedback was limited, but it is interesting to note that without this "bridge" we would not even have dared to discuss such technical details in this audience. A second presentation was done at Seoul National University. The audience was quite mixed, including colleagues, graduate and undergraduate students. A first result was the observation that the Flash animations helped to maintained a good level of interest in the audience, despite the heterogeneity of it. A second observation was that switching between Flash-, Powerpoint- and RaVi-animations was challenging for the presenter.

This problem was avoided in a follow-up presentation at KAIST in Taejeon, Korea. RaVi- as well as Flash-animations were linked to Powerpoint slides using the macro facility of Powerpoint. This improved the situation for the presenter significantly, as he could better focus on his speech and did not have to care about switching between applications. An interesting result is the proposal to have a Korean version of the MESI-bridge. Unfortunately, there is only very poor support for generating localized versions of Flash-animations.

We believe that the approach described in this paper has been very successful so far. Next, we will be working on a similar approach for embedded system design. The

topics to be covered are presented in our recently published book [5]. Slides corresponding to that book are also available for download [6]. We observed that the unavailability of tools for the visualization of the dynamic behavior of embedded systems is a serious bottleneck. We have therefore decided to extend the approach described above to embedded systems. Candidate topics include models of computation, real-time scheduling and communication protocols.

There are still some limitations of the approach described in this paper. As can be seen from the above examples, currently available technologies for dynamic visualization are far from optimal. The separation between simulation-based visualization and Flash is certainly not ideal. Powerpoint-macros are frequently disabled due to security issues. Unfortunately, none of the two tools can actually replace the other.

## 7 Summary

In this paper, we have described an approach for bridging the gap between the knowledge and motivation of undergraduate students and the details that must be discussed when the internal operations of computer are explained. It is based on using an intuitive entry into the world of computer architectures. Flash animations turned out to be sufficient for this purpose. These animations were appreciated by the audience, but it must be kept in mind that they are just bridges. The area reached via those bridges must also be presented, preferably using animation and interactivity. Currently, a combination of Powerpoint and simulation-based presentation is used. The combination of presentation techniques provided the required bridges to computer architecture to a very heterogeneous audience, including freshman and software-oriented people. However, it would be nice to have a portable and serviceable tighter integration between the different presentation techniques.

## References

- [1] R. Bryant, E. Randal, and O'Hallaron. *Computer Systems A Programmer's Perspective*. Prentice-Hall, 2003.
- [2] Cognition and Technology Group at Vanderbilt (CTGV). Anchored instruction and its relationship to situated cognition. *Educational Researcher*, 19 (6), pages 2–10, 1990.
- [3] U. B. EECS. The Ptolemy II homepage. <http://ptolemy.eecs.berkeley.edu>, 2004.
- [4] N. Hendrich. A Java-based framework for simulation and teaching. *Proceedings of the 3rd European Workshop on Microelectronics Education*, pages 285–288, 2000.
- [5] P. Marwedel. *Embedded System Design*. Kluwer Academic Publishers, 2003.
- [6] P. Marwedel. The homepage for the book *Embedded System Design*. <http://ls12-www.cs.uni-dortmund.de/marwedel/kluwer-es-book/>, 2004.
- [7] P. Marwedel and B. Sirocic. Multimedia components for the visualization of dynamic behavior in computer architectures. *23rd Design Automation Conference*, pages 378–384, 2003.