

Martha Kosa, Tennessee Technological University Department of Computer Science

I have taught a wide range of undergraduate computer science courses at my institution for more than ten years, and was the first person to teach Java at my institution. I appreciate the elegance of well-written object-oriented programs which follow the sound principles of polymorphism, inheritance, and encapsulation. I believe that design patterns can help software developers achieve elegance in addition to maintainability in their software systems. Many advanced topics have trickled down to the undergraduate curriculum, and design patterns now have their turn.

I consider myself a novice in the area of design patterns. Some talks at SIGCSE and ITiCSE have sparked my interest in design patterns, and I bought a copy of "Head First Design Patterns" by Eric and Elisabeth Freeman a few months ago. Earlier in the summer, before I had other commitments, I worked through a few of the chapters (dealing individually with the Observer, Decorator, Factory, Singleton, and Command patterns). I can see connections between just these few patterns and a lot of Sun's new Java APIs. I think that the Decorator pattern is a great way to help in the extreme encapsulation that Richard Rasala has advocated with Java Power Tools, which really impressed me several years ago at SIGCSE and ITiCSE.

As a recipient of one of this year's OOPSLA Educators' Scholarships, a requirement was to state my objectives for attending OOPSLA and my plan for achieving those objectives. One of my objectives was as follows:

I want to help my students learn OOP concepts better. This may require that I learn more about design patterns and find additional examples that incorporate multiple OO concepts.

I wrote the following paragraph as a part of my plan for achieving my objectives.

I also think that attending the Killer Examples Workshop will help me achieve my OOPSLA-related objectives. I was captivated by the Java Power Tools (JPT) GUI toolkit several years ago at the ACM ITiCSE and SIGCSE conferences. Richard Rasala of Northeastern University, the "father" of JPT, was so enthusiastic about the JPT group's work when presenting at the conferences and at one-on-one discussions. I remember that his goal was to leverage the OO principles of polymorphism, encapsulation, and inheritance (with design patterns) to provide a simple, powerful toolkit on top of Sun's standard Swing AWT. This toolkit could then be used to provide compelling user interfaces for applications. He demonstrated his Kaleidoscope application at a previous Killer Examples workshop.

In summary, I believe that the Killer Examples workshop will serve as a catalyst for my further exploration of design patterns so that I can eventually bring them to our computer science students.