

Dancing, Software Styles, and Good Software

by Paul Stachour

I've been writing software for more than 30 years, but only recently have I discovered an interesting analogy: dancing. To give a computer directions as written expression to your thoughts is to move to the rhythm of the language.

Why it has taken me so long to recognize the obvious, I don't know. I've been dancing on and off since high school.

But that was freestyle. For the last 5 years I have been taking lessons -- in ballroom, swing, Lindy, Argentine Tango, the whole deal -- and now, instead of just making things up, I've learned to step and turn with a partner according to certain prescribed patterns.

Partner dancing involves learning to operate within a set of rules. It's a lot like writing software. The ultimate challenge is not just learning the conventions but figuring out how to work within them in a way that conveys your directions correctly and succinctly.

Dancing and writing software share these characteristics:

- **Discipline.** Both require concentration and sustained effort. Competence is achieved only over time. The more capable you become, the more you realize how little you know and how much there is to learn.
- **Practice.** To learn to do an outside turn or to create a container with controlled contents is just the first step. You must practice each move until you can do it so automatically that you are free to concentrate on your partner or the interactions with other types, objects, routines, tasks, and external systems.
- **Satisfaction.** Learning how to rise and fall in a waltz, or how to synchronize a set of tasks using a set of concurrent objects through a short, time-critical code sequence in a large application, doesn't just happen. It requires time and effort. To finally get it right is especially gratifying.
- **Basics vs. technique.** Beginning dancers tend to concentrate on footwork and neglect technique. Beginning programmers think once they've written a simple program, done a clean compile, and run a few tests, there's nothing more to learn.
- **Footwork.** Beginning dancers tend to look at their feet rather than at their partners or over their partners' shoulders. Beginning programmers tend to concentrate on what's going on inside their individual program snippet rather than what other software components need to manage their activity correctly.
- **Posture and frame.** The way dancers hold their bodies and the care with which software writers create a context for their message determine how clearly intent is communicated, accepted and responded to.
- **Flexibility.** You can't dance beautifully if you never stretch; you can't write software fluently if you never read large-scale software with no or few bugs, instead spend your time only with incorrect, simple software in only one language.
- **Rhythm.** Most dancers can hear the beat. Unfortunately only a very few software writers learn how to handle time, since the ability to receive and use time information is absent or semantically wrong in most programming languages.
- **Relationship.** The goal in dancing is to make your partner look beautiful by asking them to perform only what they can do. The goal in writing software is to have other components function correctly, by not sending them requests that cause them to fault or return wrong answers. In dancing, the fault is ascribed to the lead, even when the follower makes the mistake. In software, the fault is ascribed to the caller, even when the called program is wrong.
- **Attentiveness.** Being aware of how you move your body as you listen to the music makes you more attentive to how you present yourself to others. Paying attention to programming language semantics leads to more precision in your code, and correctness of that code under more circumstances.
- **Floorcraft.** Where and how you dance depends on understanding the position and movement of the other dancers of the floor. How your code processes input and provides output depends on understanding the dependencies between your tasks, other tasks, the operating system, and the hardware.
- **Manners.** You can't be a good dancer or software writer without thinking about how you are coming across to your partner or to another software component. Good dancers and good software components have good manners.

Published in "Tango Moments", December 2003, Tango Society of Minnesota, www.mntango.org

by Paul D. Stachour, December, 2003
9532 First Avenue S., Bloomington, MN 55420
PStachour@acm.org, 952-884-5977

Motivation and dancing comparison text,
based on an article by Stephen Wilbers
see <http://www.wilbers.com>