



The ObjectWatch Newsletter

A Quarterly Newsletter for Software Architects
Issue # 54 – January 2007
Now in Our Eleventh Year

IT Architectures and Weddings

by: Roger Sessions

Contents

Feature Article	1
IT Architectures and Weddings	
Quotation of the Month	4
Subscription Information	5
ObjectWatch Services	5
Legal Notice	5

Let me tell you the four most frightening words known to mankind. And for those of you who think I am being sexist, I assure you, I am not. I mean this literally, “man”-kind. Womankind probably have their own most frightening words. I wouldn’t know. But for mankind, the words are these: “Dad, I’m getting married”.



Of course, in addition to fear, these words engender many other emotions: joy, love, hope, and happiness, to name but a few. The only other words that carry the same weight are the words I heard twenty-six years ago: “It’s a girl!”

So in a mere six months, my daughter, Emily, will complete the evolution from infant to little girl to young woman to life partner. For those readers who are still early in this process, I beseech you: enjoy every moment – the transformation moves much faster than you can ever imagine.

My daughter is getting married. *My daughter is getting married!* And only six months to plan the perfect wedding.

But there is good news and bad news. The good news is that I already have every detail figured out,

Sponsored by



International Association
of Software Architects

www.iasarchitects.org

everything from the doppio espresso machiatos to the lemon meringue wedding pie to the Texas bean chili and beer at the reception dinner.

That's the good news. The bad news is that a dozen other people also have every detail figured out. And their details and mine don't agree.

This puts Emily in an awkward situation. How does she deal with these dozen or so other people? How does she tell them that her father's plans will rule and that everybody else's will be ignored?

One possibility would be to organize a big meeting. We could then discuss the details together until everybody agreed with my ideas. Unfortunately, July would probably come and go and we would still be discussing the shape of the cake .. er, pie.

Emily came up with a different approach, one that is brilliant. And what is particularly interesting to me is that her approach to organizing her wedding is amazingly similar to my methodology for architecting complex IT systems. Like father, like daughter.

But before I get sidetracked on architectural methodologies let me share with you the email I received from Emily so you can see her solution first hand on the right.

Notice four things that you do *not* see in her letter:

- doppio espresso machiatos
- lemon meringue wedding pie
- Texas bean chili
- beer at the reception dinner

In Emily's system, I can make any choice I want, but my choices are limited to two key areas: the champagne for the toast and the plans for the morning-after brunch. I suspect that the other dozen people have also been given similar areas of responsibility.

Of course, I'm totally thrilled. I get to have fun. I

Hi Dad!

I just wanted to give you an update on the wedding stuff and ask a couple of questions. Everything is coming together great - we found a caterer and the food looks great (we're going to have grilled shrimp and chicken and stuff--very party-y! They also have yummy looking appetizers, including a Greek food platter that we ordered and thought you would get a kick out of) and we looked at the site and found a nice, inexpensive hotel nearby and started on invitations and everything. So exciting! So the date is still July 28th. We're going to have a rehearsal and rehearsal dinner (Chinese food! fun!) the 27th and there's going to be plenty of activity going on all week if you want to come up even earlier and hang out.

A question--I'd love to have you involved in some of the planning of the wedding, and there are a few aspects that need an expert hand that I wanted to ask you if you'd like to get involved in:

1. I really want to have a brunch the day after the reception, either at the hotel or nearby, for one last chance to see out of town guests before everyone parts ways. Is this something you'd enjoy organizing? It can definitely be something simple--coffee and doughnuts from Dunkin Donuts would work totally fine, or you might enjoy driving into New York and picking up bagels, or there are lots of pretty diners and restaurants in the town if you want to go that route.
2. Bing's brother is taking care of the wine but we'd like to have a few cases of champagne for a toast. If you don't feel like doing a breakfast, picking out some simple sparkling wine would be a perfect wedding present for us.

Of course you don't have to do either of these, and you probably wouldn't want to do both, but I thought you might have fun with one of them! It would be such a special gift to us if you want to!

Lots of love,
Emily

get to watch my daughter getting married *and* I get to host the brunch. I already know exactly what we will serve. We will start with doppio espresso machiatos, move on to lemon meringue pie, and end with Texas bean chili and beer. Won't Emily be surprised!

Emily is using, perhaps unconsciously, a powerful technique for organizing highly complex systems (Weddings definitely qualify for this classification.)

She is using a technique that I call *simplified partitions*. Simplified partitions are part of a more comprehensive methodology that I call *simple iterative partitions* (SIP). I'll leave the discussion of the *iterative* piece for another newsletter, and focus here on the *simplified partitions* piece.

With simplified partitions, a complex system is divided into a series of partitions. The word *partition* has a very specific meaning. It comes from the mathematics of set theory. A universe is said to be *partitioned* when all of the elements of the universe can be placed in one and only one subset of that universe - in other words, when the subsets of the universe completely divide the universe and do not intersect with each other. When these conditions are met, the subsets are called *partitions*.

Figure 1 contrasts two universes, A and B, both of which are divided into subsets. The subsets in Universe A are partitions, because every point lives in one, and only one, subset. The subsets in Universe B are not partitions, because there are some points that do not live in any subset and others that live in more than one.

Emily has used the dozen interested parties as natural partitions. She has partitioned the universe of wedding choices into subsets, with each assigned to one of the parties. My partition owns the champagne toast and the morning after brunch.

We can see a typical characteristic of partitioned systems. Few get one hundred percent of what they want, but the end result is delivered much faster and efficiently. By partitioning the various components of the wedding, the same number of people are involved, but each group's area of focus is smaller and more manageable.

In order for partitions to work, they must follow what I call the Five Laws of Partitions. The laws are as follows:

- **First Law: Partitions must be true mathematical partitions as defined in set theory.** Remember, this means two things. First, every element of the universe must live somewhere. For Emily, this means that every element of the wedding is controlled by somebody. Second, no element of the universe lives in more than one place. For Emily, this means that two of us will not show up with champagne for the wedding toast.
- **Second Law: Partition definitions must be appropriate.** This means that partition definitions must be appropriately defined for the problem at hand. For example, Emily could have said that my partition controls the first ten minutes of the wedding, somebody else's partition controls the next ten minutes, and so on. Mathematically, this is still a valid partitioning. But it isn't one that makes any sense.
- **Third Law: Partition numbers must be appropriate.** This means that the number of partitions must be well-defined to the problem at hand. If Emily had chosen only five partitions, then

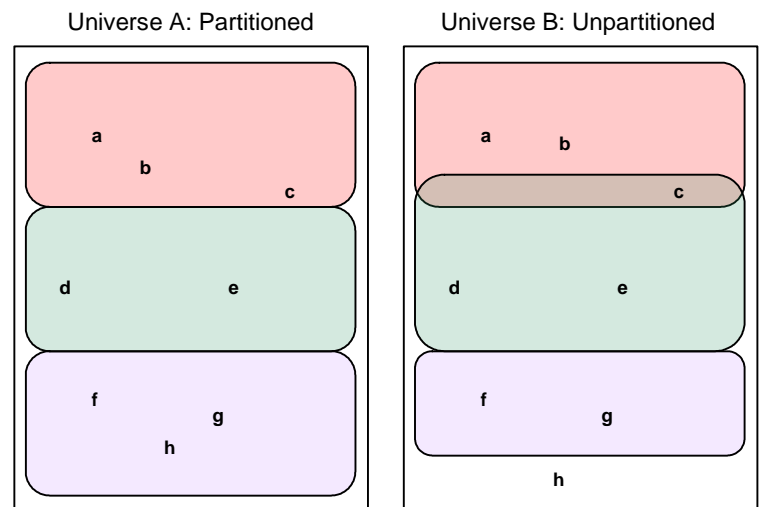


Figure 1. Partitioned and Unpartitioned Universes

seven people would have felt left out. If she had chosen twenty partitions, she would need to find another eight people to help plan the wedding (as if she doesn't have enough already!)

- **Fourth Law: Partition sizes must be roughly equal.** To minimize overall complexity, the number of elements should be roughly equal in each partition. If Emily had filled my partition with every wedding detail except her dress size and then given that element to her new Mother-in-Law, family harmony (a good indicator of complexity) would have likely been severely tested.
- **Fifth Law: Partition interactions must be minimal and well defined.** Decisions made by one partition should not bleed through to other partitions. If my decision on champagne changed the color of the wedding cake, then the benefits of the wedding partitions would rapidly erode.

But *partitioning* is only one aspect of simplified partitions. The other aspect is *simplification*. Partition-simplification is the process of opportunistically removing choices from partitions and, when possible, eliminating partitions altogether. Emily has examples of both forms of partition simplification.

She used choice reduction in, for example, the wedding toast. I get to choose any champagne I want. But I am limited to champagne. This greatly reduces the complexity of this particular choice from what it would have been if I could also choose from, say, wine, soda, juice, iced tea, beer, and mixed drinks (or doppio machiatos) in addition to champagne.

She used partition elimination as well. For example, all of the choices related to room accommodations (such as wall color, bathroom arrangement, floor plan) have been outsourced to the hotel. Nobody need worry about any of them.

Quotation of the Month

I do not support the call by 23 academics to the House of Commons Health Select Committee to commission a review of the NPfIT's [National Program for Information Technology] technical architecture. I want the Programme's management and suppliers to concentrate on implementation, and not be diverted by attending to another review.

- Lord Warner, in explaining why his \$24 billion dollar British Health Services IT project, now widely believed to be in chaos, should not have its technical architecture reviewed by a panel of highly respected software architects. BJHC.co.uk, November, 2006.
www.bjhc.co.uk/news/1/2006/n611009.htm

Both business processes and IT systems can be architected using the SIP methodology. Since both are therefore subject to the same underlying mathematical principles (those governing set theory, partitions, and, as I will discuss at a later date, probability theory), the two end up naturally aligned with each other.

On the business process side, we see autonomous business capabilities (ABCs) emerge from the SIP analysis. An ABC is a description of some business process and how it relates to other business processes, but does not include details on how that process is implemented.

On the IT side, we see software fortresses emerge. Software fortresses are autonomous software systems that can then be implemented as either service-oriented architectures (SOAs), software-as-a-service (SaaS), or any of a number of other approaches.

Autonomous business capabilities turn out to be naturally aligned with both SOAs and SaaS systems. This is no accident. It is a natural consequence of using a unified methodology on both sides of the equation (the business side and the technology side.)

So even from this introductory discussion of SIP, you can see several tantalizing promises:

- Faster delivery of results at lower costs
- A closer alignment between business needs and technology systems
- Reduction of complexity
- An architectural methodology that is grounded in mathematical theory

There is really only one serious drawback to the SIP methodology. No doppio espresso macchiatos at the wedding. You just can't have everything you want. Which, come to think of it, is one of the guiding principles of SIP. Hopefully there is a nearby Starbucks at which I can get my morning fix.

I'll SIP it slowly. Don't know when I'll see another.

- Roger Sessions
Chappell Hill, Texas

ROGER SESSIONS is a well known speaker and consultant in the field of high-end enterprise software architectures, SOAs, and Web Services. He is the originator of the Software Fortress Model for enterprise architectures and the author of the book *Software Fortresses; Modeling Enterprise Architectures*. Roger is the author of six books, dozens of articles, and many white papers. Roger writes and publishes the Architect Technology Advisory, a widely read and highly regarded newsletter on high-end enterprise software technologies. He is on the Board of Directors of the International Association of Software Architects and is recognized by Microsoft as an MVP in Architecture. More than 50,000 people have attended his workshops on scalable software systems design throughout the world. He can be reached at roger@objectwatch.com.

Subscription Information

Feel free to copy, forward, or repost this newsletter, as long as you don't make any changes to the newsletter (see the legal notices below). Thanks.

Visit www.objectwatch.com and sign up for your own free subscription to The ObjectWatch Newsletter.

Legal Notices

The ObjectWatch Newsletter does not rent out its subscription list. This newsletter is Copyright (c) 2007 by ObjectWatch, Inc., Austin, Texas. **All rights are reserved, except that it may be freely redistributed provided that it is redistributed in its entirety, and that absolutely no changes are made in any way, including the removal of these legal notices.** ObjectWatch and Software Fortresses are registered trademarks® of ObjectWatch, Inc., Austin, Texas. The green logo and The Architect Technology Advisory is a trademark™ of ObjectWatch, Inc. All other trademarks are owned by their respective companies.
