The Inside Track, March 2001
UML – Understanding Meta Language

If you've been around Great Plains for a while, you may have heard of the 3 C's: caring, courage and commitment. We've extended that to the 4 C's by the addition community. These are referred to as core values. Ideas and principals that we (Team members, both internal and external) are encouraged to reflect in our business practices, and, dare I say, perhaps worthwhile practicing in our personal lives too.

Since developers are people too (wink, wink) these values apply to us equally well. As developers, we have ideas and principals that are at once universal and yet unique to our task. We build things, things that no one can touch, yet that other people depend on and use every day. These are things of value and importance that we produce from ideas in our heads. I'd like to suggest that there are 4 additional C's that are of particular importance to developers: coordination, communication, conceptualization and combination.

Coordination is the value that represents the need of multiple developers to work together to accomplish a goal. We build large, complex things and without cooperation (another, alternative C!) we cannot build these things, we cannot build them by ourselves.

Communication. Since we must work together, the most important and perhaps the most difficult to elicit value is communication. What and how we tell each other about these things we build makes it possible to build something that is greater than the parts.

The value of conceptualization or abstraction is perhaps a developer's most unique value. We build nothing out of nothing. Or put another way, we build bits out of brains. Neither the electronic bits nor the synaptic exchanges in our brains are "real"; no one will pay you one red cent for either one. But, if you are a person who can conceive of a "real" problem and realize its solution in a software system, both you and your solution have become quite valuable. Both are conceptualizations because neither the solution nor the software are fully "real". But by bending those bits to comply with what is in your brain, you can produce a "simulation" of the solution that is so compelling, people will pay big bucks for you and your solution. It may happen one day, that what we imagine is reality, ala "Tron" or "The Matrix", but until that day developers will conceptualize a simulated solution in software.

Lastly we have the value of combination: combination of software elements to produce software systems and combinations of circumstances that those systems must deal with. But also, combination of pieces to produce a consistent (still another alternative C) whole that is beautiful. Call it symmetry or calling it efficiency, when you get it right you know it. When everything comes together "just right", I know it's a great source of satisfaction for me. If you've been developing software for some time, I'm sure you've had that feeling too. I imagine it's like building a fine piece of furniture. While it may not be evident to anyone else, you know that all the pieces fit together tightly, that the product serves it purpose without an excess of material or undue ornamentation and that the versatility and strength of your work will serve its new owner well for a long time. Such is the feeling a developer gets from a job well done.

Who ML?

UML stands for Unified Modeling Language. We use languages to communicate with one another for many reasons coordination being one, especially in the workplace. A word (or symbol) in a language is a concept. If I refer to my "hand" when talking to someone, the word is a conceptualization of my hand; I didn't give the person my hand. And certain words are more pleasing than others, sometimes they literally "sing"!

UML is a modeling language. Specifically, UML is for modeling the object-oriented analysis and design of a software system. A model will be quite useful in fulfilling the four C's of software development. Models are a conceptualization of the solution that allows analysts or designers to communicate those ideas to others who must coordinate and finally combine their activities to produce the software system. Whew, all four in one sentence.

UML is unified because it brings together the work of 3 individuals known as the three amigos and ideas from many others. The 3 prominent individuals are Ivar Jackobson, James Rumbaugh and Grady Booch. I'll spare you more C's from here on out.

Another Language?

Why is another language needed to communicate the results of the analysis or design process? What's wrong with spoken languages like English? Well, the answer is nothing is wrong with English, but specialized languages are used everyday as a concise means of exchanging ideas. "Are pork bellies up or down?" Who the heck cares of pigs are laying on their back or on their stomachs? This specialized language may be heard in the context of rural America where one farmer is asking another if he knows the market value of the portion of a pig used to make bacon! Need I go on?

UML is specifically designed to address the needs of OO analysis and design. It is also a standard (see http://www.omg.org/uml/). Because it is a standard, when I speak (or draw) UML, then other people can understand it. Because UML is a specialized language, communication can be more concise, precise and self-evident. Notice I said more not completely. Any language can be misused.

And like any language, there can be local dialects. UML supports further specialization of the language. UML might be the language of software development, but there is also the language of software development at Great Plains or even within your organization.

Who me?

So why am I spending precious bits and white space writing about UML? Because as software development becomes more complex (haven't I beat that one to death in previous columns?), the 4 C's become more important and I hope I have or will convince you that UML will support those 4 C's very well. And not surprisingly, UML will be an important part of the new platform at Great Plains. Yes, the new platform based on .NET!

I'm sure you have an opinion about how well Great Plains has supported its Solution Developers. I hope you will agree that Great Plains has always tried to excel in that area. But I know you'll agree that Great Plains can always do better.

UML and Rational Rose, a "word processor" for UML, will allow Great Plains to live up to the 4 C's of software development to an even greater extend in the future. With Rose (see http://www.rational.com/rose), you will be able to more quickly understand the products Great Plains builds. You will be able to more quickly extend and integrate to those solutions. Rose, along with additional tools provided by Great Plains will allow you to turn ideas into code, to adapt to changes and deliver greater functionality with a higher level of quality than every before.

The You in UML!

I don't have any code this month, bummer. I really like code. I don't even have any UML diagrams as examples. Double bummer. Why? I'll more than likely take a stab at UML specifics in a future

column. But it's like I said about XML, its relatively simple at its core, though not nearly as simple as XML. More importantly, its what you do with UML that makes it interesting and useful.

UML is a set of diagrams that represent views into a model of a software system, hence a modeling language. Just showing you some diagrams without describing the process of conceptualization and communication, without having a model that underlies those views is pretty hollow and meaningless. It will happen. In the mean time, I have a couple of suggestions for UML books if you are interested.

- UML Distilled by Fowler, Scott and Booch (ISBN: 020165783X), 1999, Addison-Wesley
- Instant UML by Muller (ISBN: 1861000871), 1997, Wrox Press

Four C's for Me

If you have any comments, complaints or suggestions for content to this column, feel free to e-mail me via solutiondeveloper@greatplains.com. Good communication requires the involvement of both parties.

Till next month,
Karl Gunderson
Technical Evangelist