JOURNAL OF OBJECT TECHNOLOGY

Online at www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2010**

Vol. 9, No. 3, ;May-June 2010

It’s a Small World - Globalization

John D. McGregor, Clemson University and Luminary Software LLC, U.S.A.

Abstract

Globalization continues to be a hot button issue despite having been talked about for years. It brings many opportunities while raising many risks. Globalization has gone from being a strategy to a necessity, but one that has strategic impact. In this issue of Strategic Software Engineering I want to explore some aspects of globalization including the globalization of products and the globalization of development that continue to have strategic implications.

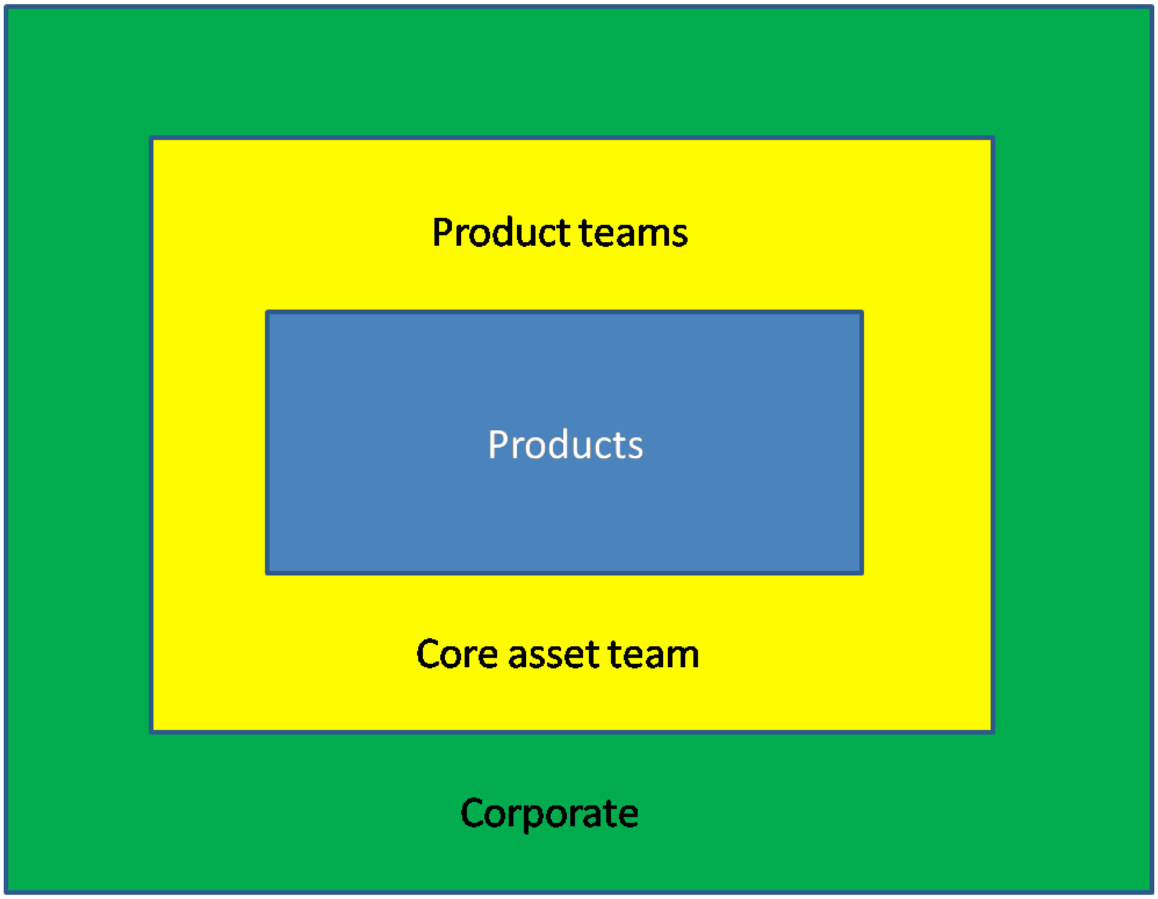
1 INTRODUCTION

Anyone who has visited any of the Disney parks probably has the tune to “It’s a Small World” running through their heads about now – over and over again. This attraction was a light-hearted way of exposing people to diversity – peoples from many lands. In the world of software development our boundaries have been broad for a long time. We have had diverse teams for many years. Development organizations have been globalized, albeit in a single location, for many years. By that I mean we have had people from many countries working together to build a software product, but they did it together in one country, even one building, whether in the US or Germany or any other country. In more recent times mergers and acquisitions, expansions, and other initiatives have led to an organization having development facilities in multiple countries. And as software has become larger and more complex, several of those facilities have collaborated to build a single product. Software product lines have taken this still further where an organization may have core assets and products being built in numerous locations around the world.

I really faced the globalization of products head-on several years ago when we helped a church denomination build and deploy an accounting system that had to have configurations from some 170 plus different languages – including an appropriate keyboard - and a very large number of different currencies. Individual churches were able to define policies about how their systems worked and the system had to enforce the policies. This “localization” of a product is routine these days and while it seems to be the antithesis of globalization it really is a result of globalization. I am going to push the meaning a bit here and say that globalization also means that the product has to work in different environments around the world. The basic product was used globally but configured to work locally.

John D. McGregor: “It’s a Small World – Globalization”, in *Journal of Object Technology*, vol. 9, no. 3, May-June 2010, pp. 7-17 http://www.jot.fm/issues/issue_2010_05/column1/

Globalization is often spoken of as a strategy. It is too late for that. A strategy is something you select for advantage. Globalization is a necessity. It is an evolutionary path the world is following. Information technologies have not made this evolution happen but they do faciliate it. The fact that there are still many significant differences among the peoples of the world, institutions of the world, and countries makes it a bumpy path. The question of interest now is what strategies can we apply in a globalized world?



Harper identifies four dimensions of globalization: social, political, economic, and cultural [Harper 93] (although [Potter 02] deleted the social dimension). Our industry is impacted by all four:

- The political dimension is illustrated by the Google/China issue and many other similar ones and there have been different but similar problems in Iran. It is doubtful that any country will be able to resist the technology wave of the future. Concern: what features should products have and who should determine when they are enabled?
- The social dimension is illustrated by the potential of social networking systems to propagate the idea of the global village and the threat of under-age people

being brought into contact with negative forces. Concern: Can we have variation points that are based on users rather than functional requirements?

- The economic dimension is illustrated by the increasing regional integration such as the European Union. Concern: How should licensing and deployment models vary around the world?
- The cultural dimension is illustrated by the increased communication of a variety of ideas and the availability of Western movies, books, and magazines around the world. Concern: are there elements in different cultures that can be used to make designs more flexible?

Globalization affects a product development organization at the product level by requiring mechanisms to be placed in a product so that it can be configured to run anywhere on earth. Globalization affects the engineering teams by introducing interactions among people of different cultures with different approaches who are all striving for common goals. Globalization affects organizational management by requiring strategies that are global in scope but sufficiently flexible to accommodate local variations.

There are several cross-cutting aspects of globalization that interact with the dimensions of globalization. I will speak to two of these.

The geographic aspect, recently referred to as distributed development, has perhaps been a fact of life for project managers longer than other factors. Most professionals still seem to work better in a distributed environment if there are occassional “face-to-face” interactions even though it is more expensive in terms of travel costs and imposes a heavier price on the employees’ personal lives. While consulting with a telecommunications company I once traveled with 5 other people to a two hour meeting with 4 people at another of the company’s sites. Ironically, due to a shortage of conferences rooms, the meeting was held in the video conferencing facility.

The locovore movement in food is driven by the assumption that eating those things that are grown close to us is better for us than food grown further away. There is something of the same reasoning in software development about requirements elicitation. The feeling is that the development project needs local access to users of the system being built. Many US corporations became multi-national when they discovered that they could sell more in foreign markets if they were not perceived as outsiders. An explicit strategy.

The temporal aspect has both short term and long term considerations. Daily there is the problem of scheduling teleconferences among the members of a distributed project. Scheduling system maintenance and backup becomes more difficult. The software product line strategy introduces an additional temporal wrinkle. The organization often has multiple product development teams working in parallel. Sharing an evolving set of assets and program construction techniques is more complex when multiple groups wish to use the same assets but are in different stages of product development. Rather than thinking about a portfolio of projects as independent entities, a rhythm must be established for the product line. This becomes the driving signal.

Consider the following software product line scenario that I will use in the discussion of globalization. A multi-national corporation is developing a product line that it will sell in virtually every country on earth. It is a set of personal productivity tools that will be tailored to the preferences of indvidual locales. The products must use the native language of typical citizens in these locales. The US-based corporation is using a team of its best developers in the US to build the core assets. That team of 10 people is composed of people from 4 countries: US, European Union, Russia, and Thailand. These core assets will be used by product teams in about 20 development labs in various locations outside the US. It is virtually impossible to schedule a real-time telecon due to time zone differences and several product development teams will not be chartered for as much as a year so information is communicated using a wiki, FAQs, and pre-recorded webinar presentations. Each lab is responsible for assembling a set of products that will be appropriate for audiences in their region.

Globalization is exciting and scary. Every new market that is opened also brings new competitors.A number of well-credentialed groups have written reports on globalization and I do not intend to try to recreate their work. I will stick with what I know which is what managers and engineers can do to make projects successful. In this issue of Strategic Software Engineering, I will examine how globalization influences our choice of strategy and our design and method decisions.

2 GLOBALIZATION EXPANDS OUR CONTEXT

Globalization results in boundaries being crossed and eventually virtually eliminated. The context for software-intensive products broadens as we become more globalized. For many years the audience for most software development was English-speaking people. As software became more of a commodity that context expanded to include anyone in the domain of the product and language became a variation point. Now software is part of the everyday lives of almost anyone who has access (even occasionally) to electricity.

I am a guest editor of a special issue of IEEE Software that will be out about the same time as this column. I am based in the USA and my co-editors are from Germany, Japan, and Texas, so 3 different countries (4 if you ask the Texan ☺). We have had a good working relationship due to knowing each other before hand and communicating asynchronously (more about this later). We share the common culture of software engineering but not the same timezone so we use email. Same religion? I don't know because it is irrelevant in this context. What is relevant is their influence in the software product line community to attract submissions from the very best authors. Our "globe" includes anyone using the software product line approach. Globalization itself is context dependent.

Each person is the center of their own universe. How big that universe is depends on the aspirations, knowledge, and opportunities the individual has. I routinely have graduate students at Clemson (in the US) who are from India and who intend to work for a company whose headquarters is in yet another country. Their capacity for working with a diverse group is far greater than the student who has never left India or South Carolina. Each person has their own unique context that guides how they act and react to situations. As a consultant as well as a professor I have a different context than many of my fellow professors. I see things in a different perspective.

We must be more strategic to succeed in an era of globalization. Jobs are leaving the "developed" countries and going to the "developing" countries. The good news is that raises the standard of living for those in the developing countries but the standard of living in the developed countries becomes harder to maintain. Many companies never make their strategy explicit. They simply have it in the heads of a few managers. The problem with this is mantaining consistency. The era of globalization has given rise to many new strategies. By being more explicit about strategy we can engineer the best strategy. We have had success with, for example, engineering the production strategy in a software product line organization [Chastek 09].

Globalization results in linkages among various institutions and peoples around the globe. Stock markets, medicine, other knowledge-based areas are linked. Events that a few years ago would have caused some local distress are now known around the world and quickly affect actions. Terms such as offshoring become meaningless when a corporation is a citizen of the world and no place is outside their context. "Distributed development" is the new term and it recognizes that any situation in which persons are not face-to-face has certain characteristics; however, offshoring was difficult to manage, not because of a lack of face-to-face meetings, but because of differences in culture. As the local cultures blend into a software engineering culture these problems subside.

3 VALUE INNOVATION

This column was originally going to be about innovation but I decided globalization was of more immediate importance and had implications for innovation as well. As institutions are globalized and our horizons extend further, because boundaries have been removed, it is more encumbent upon each worker to add value by innovating. I can't guarantee that if you are an innovator you will not be laid off, but I am fairly certain that the innovator will stay in the company longer than the person who does their job the way it has always been done and never thinks critically about what they are doing.

Value innovation is a term used to describe a type of strategy in which the strategist focuses on providing value to customers rather than focusing on beating the competition [TBA]. The idea is to make the competition irrelevant or at least force them to change their business model. The open source strategy was a value innovation. It changed the way software is thought about. In the broader world of business the meaning here is clear but what about our technical world, for those of us who do not determine business strategy? Here are three techniques that support value innovation in a globalized environment:

Service-oriented architectures (SOA) are nothing more than the latest relaxation of static binding. SOA is attractive from a strategic perspective if we are convinced that something better than our current implementation will come along. We want an architecture that will allow changes to the system in the simplest way possible [TBA]. SOA allows small businesses, even individuals, to provide a unique service much more quickly and cheaply, than building a complete application. One business model anticpates services that will be relevant for a short period of time to be replaced with a new service, derived quickly and cheaply from the previous one.

Software product line organizations build an inventory of assets that represent options for the future. They address the long-range temporal dimension by using attached processes to communicate to future product builders [McGregor 10]. The organization provides value innovation by managing variation – differences in software, in people, in institutions. Variation points help maintain global consistency by modularizing differences while allowing tailoring for local situations.

Domain specific “languages, models, …” would seem to be contradictory with the intention of globalization but they provide a means of expressing value innovation by delving into a specific idea [TBA]. They provide a means of specifying unchanging ideas at one level while letting ideas at other levels to vary. That is, a domain specific vocabulary can be used to express unchanging ideas while the code generated from that vocabulary can be changed significantly.

4 NOW BACK TO GLOBALIZATION

So what does value innovation have to do with globalization? Competition and position. The global market place is much more crowded than the local market place. Almost always our reaction is to look to our competitive advantages and use them. In fact now we need to create competive advantages. This is difficult to do in a globalized environment because our “production capital” are our people, but our people are perhaps the most mobile of all our resources.

How do you retain the best people? The answer is walk a tightrope. Help your people grow and do the things they wish but allow them to do it in your organization.

What is your strategic advantage? Why would your boss give you the choice assignment rather than someone else? Breadth of vision. Your innovative ideas need to be global.

Today's software systems are becoming globalized in the sense that they stretch around the globe. The telephone network has been global for a very long time. Ultra-large scale systems (ULS) are becoming more commonplace. It is one thing to design a product so that it can be used all around the world, it is another for an inter-dependent set of systems to evolve into a world-wide system of systems used at all points around the world at the same instant. The study of ULS published by the SEI and led by Linda Northrop provides some thought provoking ideas for globalization at the product level [].

The ULS study shows that global systems will be decentralized, heterogeneous, and inherently inconsistent. This basically takes distributed development to logical extremes.

It illustrates that coordination beyond an interface level will be impossible and even interfaces will be coordinated through levels of increasingly complete definitions.

5 GLOBAL SOFTWARE DEVELOPMENT

Software product line techniques allow us to achieve economies of scale while still supporting mass customization to satisfy niche markets. How can we be successful at software development in a world where everything is "local" and "global" at the same time? By taking the long term view, being strategic about how we design.

Attributes

Several product quality attributes are important in the context of globalization of development. The standard attributes of information hiding, encapsulation, and modularity are even more important in this context.

Modularity – By dividing the behavior into units, each of which has a single focus, the overall complexity of the system is reduced and the complexity of each piece is manageable.

Information hiding – Language constructs that limit references to attributes from outside a specific boundary maintain the clean separation between units.

Encapsulation – Having distinct packaging for behaviors makes it possible to remove and add packages cleanly.

When we consider globalization of the products another attribute becomes important:

Interoperability – To support products that have to be competitive around the world, they have to be designed to interoperate with a wide range of environments.

Techniques

There are several architectural techniques that support gloabalization.

Platforms – The term platform has been overloaded for some time. I mean a software platform, often it includes a hardware abstraction layer, that provides some specialized services such as advanced graphics capabilities or computational capabilities. The design explicitly supports moving products from one environment to another.

Interfaces – An interface gives a definition against which multiple implementations can be written. In a globalized marketplace as many interfaces as possible need to be elevated to domain standards. This ensures that an inventory of implementations can be built up.

Constraints – The assumptions of an architecture are all too often undocumented. Adding constraints to the architecture definition puts that bit of knowledge in the hands of the architect. In a distributed development environment all knowledge about a particular architectural element needs to be packaged together.

Business Models

An IBM sponsored study showed that companies that are financially overperforming put twice as much effort into business model innovation as those companies that are financially underperforming []. A major result of the study showed that collaborative innovation was a major tool used by a majority of the companies. Open source development has long been globally distributed. The model of self-organizing groups is useful in this context.

The IBM study developed a model for exploring busines smodel innovation. The three components are:

- Industry model – What products does your industry make? What compliments those?
- Revenue model – How does the organization realize income from their activities? Sales of products or sales of services? What are the widely used pricing models?
- Enterprise model – What are the main businesses of the enterprise? Success has been achieved both by integrating traditionally separate activities and by focusing on a "core competence" and eliminating traditionally related activities.

In Table 1, I use the three models and the framework of issues developed in the IBM study to examine a globalized software product line organization producing generic products.

Table 1 Tool for defining strategy

	Context	Current position	Capabilities
Industry	Mass customization is a business model enabled by SPL; we have domain models of the products	Companies that adopt SPL range from large market dominators to small companies.	Many of our techniques are scenario-based and we use domain models
Revenue	Nokia saw that selling the core assets as well as the products would maximize revenue.	After the first few products, the costs are amortized over the products and revenue can be used to mature the core asset base.	SPLs have a variety of revenue models. Philips has used a variety of models to pay for reusable assets.
Enterprise	By separating core asset development and product building SPL has an enterprise with two perspectives: parts and complete products.	SPLs have been successful with collaborative development. HP has numerous strategic partners around the world.	Quick response; cheaper production; higher quality

Globalization sets a rhythm for software business activity. Agile methods are a direct result of this perceived need to move ever faster in software development. Jan Bosch of Intuit speaks of ecosystems in a different way than I do [Bosch 09][McGregor 09a]. In his view the ecosystem feeds off of the product line of a company and in return provides nourishment (customers) for the product line. This approach pushes product development out into the community as one way to meet the demand for faster and yet faster deliveries. He provides an ecosystem that sustains smaller companies developing small products that work together or at least they work with the focal product. My view of an ecosystem more closely resembles a biological ecosystem in which the entities are interdependent and sometimes mutally supportive while at other times they are competing.

Speculation

A question has arisen as I write this, could we define a style of development in which teams are deployed geographically so that someone was working a specific product every minute of the day? As one developer left off another comes "on duty." Certainly we have organizations with sufficient geographic diversity that they have people doing software development every minute of the day and some projects are dispersed sufficiently that that someone is working on some aspect of the product all the time but what about one part of the product worked on different people at different times of the day? So that one has the lock and then passes it on. Are there economies to be realized?

About 15 years ago I had surgery to remove a tumor on my auditory nerve. It took 13 hours. It was done by a pair of surgeons. One a neurosurgeon and the other an acoustic surgeon. I was impressed as they explained their surgical plan how they timed things so that each of them got a break every so often and each got lunch while the other carried on with the surgery, but only doing tasks rlated to their speciality. Is it possible to take advantage of geographic distribution to make more software development more efficient just as they have temporal distributed their tasks?

6 SUMMARY

Globalization is not a business strategy, it is reality. It is happening whether we like it or not. So our business strategy must react to globalization. We do this technically by making our designs less ambiguous, more robust, and as flexible as possible. We do it for our products by using variation mechanisms that support configuration for local conditions. We do it for the organization by seeking competitive advantage through software product line strategies and other initiatives that make our products as attractive as possible but take advantage of economies of scale and scope.

7 ACKNOWLEDGEMENT

Thanks to Greg Barnette of Clemson University for valuable comments that allowed me to greatly improve this column.

REFERENCES

[Arora 09] Ashish Arora, Matej Drev, and Chris Forman. Economic and Business Dimensions: **The Extent of Globalization of Software Innovation** http://mgt.gatech.edu/directory/faculty/forman/pubs/acm.pdf, 2009.

[Bosch 09] Jan Bosch. From Software Product Lines to Software Ecosystems, in Proceedings of the 13th International Software Product Line Conference 2009, McGregor and Muthig, eds., 2009.

[Chastek 09] Gary J. Chastek, Patrick Donohoe, and John D. McGregor. Formulation of a Production Strategy for a Software Product Line, SEI, CMU/SEI-2009-TN-025, 2009.

[Harper 93] C. H. Harper, Exploring Social Change, Prentice Hall, 1993, pp. 244-249; and L. Sklair, Sociology of the Global System, The Johns Hopkins University Press, 1993.

[McGregor 09a] John D. Mc Gregor: "Ecosystem", in *Journal of Object Technology,* vol. 8, no. 6, September-October 2009, pp. 7-16 http://www.jot.fm/issues/issue_2009_09/column1/ .

[McGregor 9b] John D. Mc Gregor: "Ecosystem, continued", in *Journal of Object Technology,* vol. 8, no. 7, November-December 2009, pp. 7-23 http://www.jot.fm/issues/issue_2009_11/column1/

[McGregor 10] John D. Mc Gregor. "Attached Processes", in *Journal of Object Technology*, vol. 9, no. 2, March – April 2010, pp. 7-16 http://www.jot.fm/issues/issue_2010_01/column1/

[Microsoft 09] Welcome to the Windows Ecosystem Readiness Program, http://www.microsoft.com/whdc/Win7/default.mspx, 2009.

[Potter 02] Potter, C. 'Global Convergence, Divergence and Development', in Desai V and Potter, R. (eds) (2002), The Companion to Development Studies, London: Arnold, 2002.

About the author

Dr. John D. McGregor is an associate professor of computer science at Clemson University, a visiting scientist at the Software Engineering Institute, and a partner in Luminary Software, a software engineering consulting firm. His research interests include software product lines and component-base software engineering. His latest book is *A Practical Guide to Testing Object-Oriented Software* (Addison-Wesley 2001). Contact him at johnmc@lumsoft.com.