

Ecosystems, continued

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

Abstract

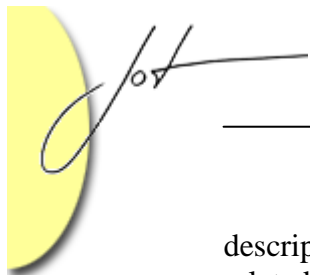
In the last issue of Strategic Software Engineering I began talking about ecosystems and software product lines. The network of dependencies among an organization and its stakeholders can become complex and difficult to understand much less manage. In this issue of Strategic Software Engineering I will continue this topic by looking at details of the SEI's Software Product Line practice areas and how they pertain to the product line's ecosystem.

1 INTRODUCTION

In the previous issue of Strategic Software Engineering I began talking about the ecosystem of a software organization [McGregor 09]. I'll try not to repeat myself so I suggest you read my column in the previous issue. In this issue I will use much of the terminology and concepts discussed there. In particular, the *ecosystem* of an organization is the complete set of entities with which it interacts to satisfy its goals.

I became interested in the concept of an organization's ecosystem when visiting a software product line organization. They were subcontracting the development of core assets to a company, call it subA, with whom they had done business for several years. That company subcontracted to another company, call it subB, for some specific expertise. For another set of core assets, the product line organization decided to contract directly with subB, who in turn decided to subcontract with subA, for parts for those assets. How could a manager in any of the companies keep track and manage strategic relationships in this environment? How can subA and subB interact in ways that are to their mutual benefit when they are both collaborators and competitors?

As I start work on this column I am attending the Software Product Line Conference (SPLC 2009). Jan Bosch of Intuit presented a paper on evolving a product line into an ecosystem [Bosch 09]. He is speaking about ecosystems in a slightly different but related manner to the ideas expressed here. Bosch sees the product line, which provides income for the developing company, becoming sufficiently attractive to other companies that additional business is generated for everyone. For example, selling add-on or plug-in products for a more established product is one such scenario. That fits within my previous



description of an ecosystem but I go further to include all transactions both directly related to the software development organization and those indirectly related as well.

“Business Ecosystems are defined as intentional communities of economic actors whose individual business activities share in some large measure the fate of the whole community” [Moore 05]. For me, the key word in this definition is “intentional.” It means I have some control over the outcome. I can make decisions that improve my situation.

In a software product line organization the ecosystem view is particularly useful because there is a larger cast of characters and relationships over the life of the product line than is typical for single product development. There is a correspondingly larger range of actions that can be used to add value. One of those ways is to add variations that are attractive to certain markets. More about that later.

At Philips, the very large size of the company makes it possible to think of the product line organization as an ecosystem within the corporation. The inner source approach used by Philips [Wessilius 08] and discussed in [McGregor 08] establishes an ecosystem that involves several business units in the corporation. The ecosystem extends outside the company to include the users of Philips equipment, a highly skilled audience, and more indirectly organizations that learn from the experience of Philips regarding the open source techniques. Philips has been evangelical in spreading information [Koch 08] about their software development techniques in the hope of stimulating improvements in tools and processes as well as more open source software from which they can benefit.

Every organization operates within larger environments that it influences and that influence it. Taking control of this environment to the extent possible, and anticipating the influences of the ecosystem, allows management to be pro-active. In this issue I will relate the ideas of ecosystems to the practice of operating a software product line. I will incorporate the transaction analysis technique [Baldwin 07] discussed in the previous issue into my discussion. First I will discuss the relationship between a software product line organization and the ecosystem.

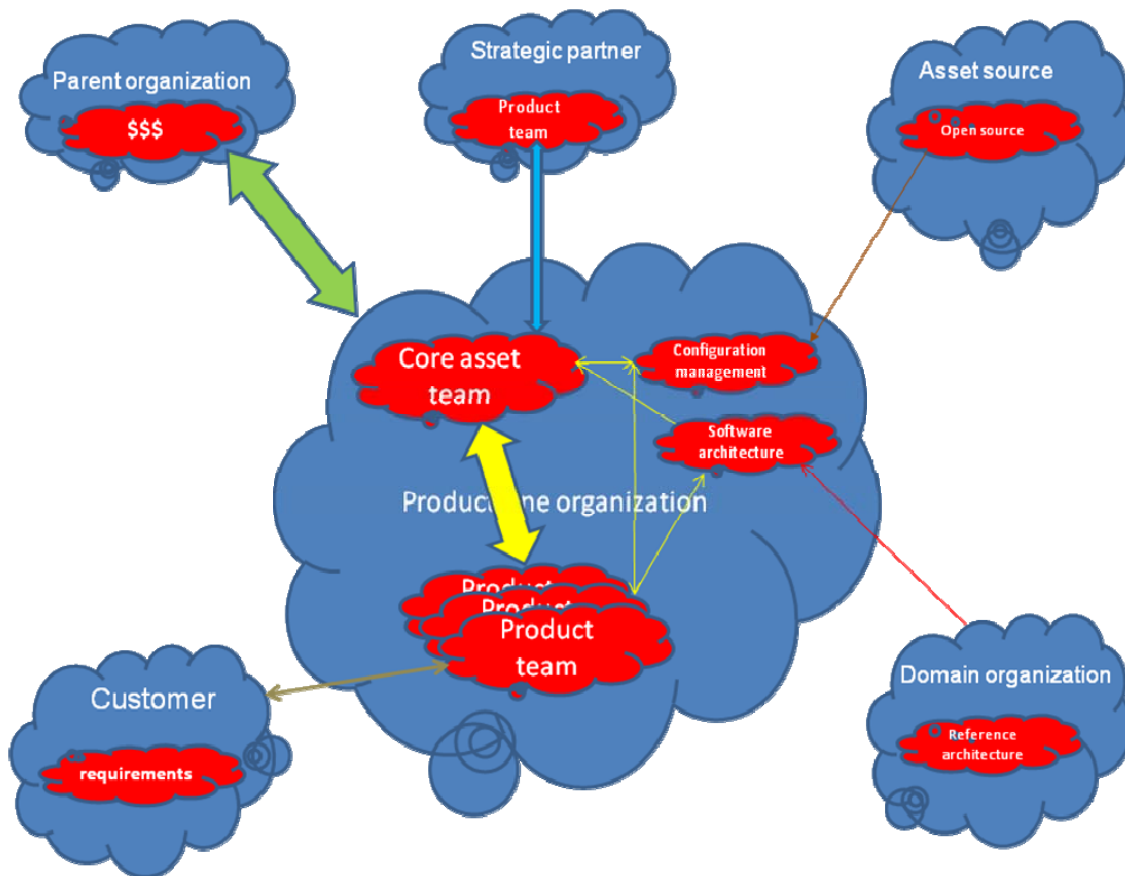


Figure 1 - Product line ecosystem

2 A SOFTWARE PRODUCT LINE ECOSYSTEM

Software product line organizations are part of an ecosystem. In fact the constituent pieces of the organization interact with external organizations in a variety of ways. There are many ways to arrange the transfers¹ among these keystone parts of the product line organization, which in turn influence which transactions occur with external entities.

Each product has its own specific audience within the market of the product line. Each audience has links to other products and people. For example, the product line organization uses tools, many of which have forums and bug reporting systems that provide access to other organizations and their experiences.

Figure 1 shows the transaction analysis technique that I discussed in the Part 1 article to represent a portion of the product line organization. The different colors shown in the figure denote different relationships. The yellow color indicates transfers while the other colors represent transactions. Note that even the acquisition of open source software is indicated as a transaction because a license agreement is entered into regardless of how

¹ Remember that transfers are very low cost activities while transactions incur significant costs.

restrictive it is. The ecosystem can be understood by looking at transactions, or transfers, for all of the entities:

Internal to the keystone (product line) organization – The two-way arrows between the core asset team and the product teams denote that core assets are delivered to the product teams and the product teams will provide feedback, and sometimes modified assets, to the core asset team. The feedback includes defect reports and feature requests. The difference in the color of the arrows between the internal and external product teams indicates that the core asset team will interact with the internal product teams using informal transfers while requiring some level of transaction with the external entities. For example, internal teams may be able to access source code for the code-based core assets while external teams may work with binary only.

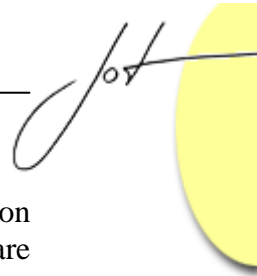
Between the keystone and its parent organization – The parent organization designates an organization to manage the product line. The parent may formally charter a new independent unit, it may form a unit that cuts across multiple business units, or it may form an organization within an existing business unit. The parent provides business goals for the product line, some statement of scope, and initial funding for product line adoption. In return, the product line organization provides a positive return on investment for the parent organization. Typically these are internal transfers unless some portion of the product line organization is in a different business unit which may require a more formal transaction.

Between the keystone and a strategic business partner – The software product line organization may establish alliances with other organizations to produce products in the product line. The transactions between them may involve a sale of rights to use the core asset base (money flows to SPL organization) or a contract to build a specified product (money flows to the partner). In both cases free transfers of bug reports and feature requests flow back to the SPL organization.

Between the keystone and a customer – The software product line organization establishes an External Customer Interface that provides the avenue for transactions and transfers with customers. Interfaces may include salespeople who interact directly with customers, automated forums and FAQs, and user groups. These are usually transfers of information unless there is a charge for tech support.

Between the keystone and a source for assets – The Externally Available Software practice area in the SEI's Framework for Product Line Practice addresses how to bring software that has been developed elsewhere into the product line. Each piece of such software establishes a dependency between the product line organization and the organization that built the software. Even if the software is free there are costs for adding into the core asset base. The product line organization needs policies about how to handle these dependencies. For example, when the external software is updated, how does the product line organization adopt the upgrade?

Between the keystone and a domain organization – The product line organization needs to Understand Relevant Domains and often this is accomplished by transfers and transactions with a domain organization. For example, if the SPL is about software



development tools the appropriate domain organization might be the Eclipse Foundation [Eclipse 09] and/or the Object Management Group [OMG 09]. The typical exchanges are flows of domain models to the product line organization and sometimes feedback from the SPL organization. Whether these are transactions or transfers depends on the organization. Many of the domain efforts are working groups whose membership is determined by contributions of money and the time of professionals, but free riders often can obtain the completed domain model for free.

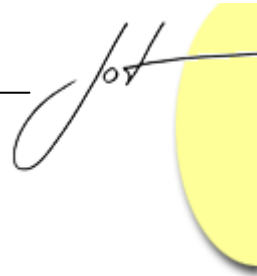
Many of the other practice areas defined in the SEI's Framework for Product Line Practice [Clements 01] affect the ecosystem of the product line. Table 1 provides a brief description of how each practice area interacts with other elements of the ecosystem. Each of these can result in transfers and transactions between entities. For example, Figure 1 shows that new versions of externally available software are handled in the configuration management system; the standard configuration policies regarding new versions of any asset are followed. This may mean that the new version is included in the next release of the entire core asset base or it may be immediately available as part of a continuously available asset base to which everyone has access.

Within the product line itself the ecosystem nature of the organization can be seen. In an ecosystem, changes to one entity can ripple through many other entities. The most common example is the ripple across products. The product line manager has the opportunity to make trade-offs among the products: spend more on building an asset for product B making it possible to make product C more quickly.

Table 1 - Practice area contributions

Practice Areas	Interaction with ecosystem
Software Engineering	
Architecture Definition	A standard or dominant software architecture in a domain can be the foundation for an ecosystem. Sun's J2EE architecture drove markets in books, other software, and consulting. Organizations begin to interact because they want to use the architecture or want to influence its evolution. Remember CORBA? The Eclipse architecture has made possible many commercial products, which contribute to the Eclipse code base.
Architecture Evaluation	The evaluation of the product line architecture requires a broad examination that covers the concerns of all relevant members of the ecosystem.
Component Development	The product line architecture may make it feasible for others to write and provide components. This may be low-cost plug-ins that add capabilities or it may be an iterative scheme to rush a product to market with limited functionality and gradually increase functionality over time.
Using Externally Available Software (formerly COTS Utilization)	The Externally Available Software is acquired and integrated into the core asset base. This establishes a dependency within the ecosystem between the product line organization and the provider of the software.
Mining Existing Assets	If some entity in the ecosystem has access to previous products, the SPL organization may be able to create core assets from those products. The broader the ecosystem the larger supply of code from which to choose.
Requirements	Members of the ecosystem are a source of requirements for the

Engineering	product line and for validation of requirements provided by others.
Software System Integration	Members of the ecosystem provide a variety of types of integration. In some cases it is typical code integration, but in others it is integration of the products of the product line into their business processes.
Testing	One scheme is to involve members of the ecosystem in using and providing feedback regarding defects and features. Users groups have performed this function for many years and now users of open source products also provide feedback.
Understanding Relevant Domains	The domains involved in a product line must be coordinated. By that I mean the needs of one domain may directly contradict another domain. The requirements and architecture must arbitrate these contradictions using the appropriate constraints.
Technical Management	
Configuration Management	The ecosystem involves many assets coming from many different sources. Some are code but models, test cases, data, and other information in a variety of formats are essential to the success of the product line. The ecosystem will encompass multiple repositories with differing security levels. A worker may need to retrieve information from several sources to work on requirements validation, for example.
Measurement and Tracking (formerly Data Collection, Metrics, and Tracking)	The product line organization should maintain a program tracking the cost/benefit of each transaction in the ecosystem. This information should be used to continually evaluate relationships in the ecosystem.
Make/Buy/Mine/Commission Analysis	The product line organization searches all sources in the ecosystem to find all possibilities for acquiring needed assets.
Process Discipline (formerly Process Definition)	Processes are needed at critical points in the ecosystem. They need not be formally defined or even written down if the team is sufficiently small and co-located. Those critical points correspond to the “transactions” identified in the ecosystem mapping exercise. The processes define what triggers the transaction, who is responsible for the transaction, what is included in the transaction, and how the results of the processes are evaluated.
Scoping	The scoping practice interacts with ecosystem creation. Narrowing the scope may eliminate some of the potential participants in the ecosystem. That is, the fewer products in the scope, the more limited will be the possibilities for interaction with outside groups.
Technical Planning	The technical planning in the ecosystem determines how the various elements of the ecosystem will communicate and share data. Which servers will be open to outside access?
Technical Risk Management	The technical risks associated with the ecosystem include the inability to keep up in a rapidly changing environment. New members of the ecosystem may cause it to change direction or to lose competitive advantage. These risks are mitigated by active involvement with strategic partners and organizations such as open source foundation boards of directors.
Tool Support	The flow of information can be facilitated by using a transformation approach in which one type of information is transformed into another type. Standard mappings can be defined and then reused every time a source document is updated.
Organizational Management	
Building a Business Case	The business case for a software product line will usually be constructed before an ecosystem forms, but the business case can anticipate the benefits from external relationships.
Customer Interface Management	The ecosystem subsumes the customer interface and provides a natural interface between the company and its customers. The user groups, forums, and other feedback mechanisms provide information that is captured contemporaneously rather than surveying people after they have forgotten specific issues.



Developing an Acquisition Strategy	Acquiring assets is handled in many ways. There may be several sources within the ecosystem for any given asset. There needs to be a technique for determining the best source.
Funding	Several members of the ecosystem may share funding the cost of the product line organization. Multiple business units may join together to create the product line. Sharing the cost can require major negotiations. This can be harder than working with external organizations where formal contracts are norm.
Launching and Institutionalizing	The startup of the product line organization can involve interactions with members of the ecosystem.
Market Analysis	Collaborating with customers through user groups and forums gives the product line organization a high-quality source of information about the levels of customer interest and the requirements they would most like to add to the product. Collaborating with competitors to develop a basic infrastructure reveals some of the competitor's future plans.
Operations	The operations of the product line organization are organized around the two roles of asset developer and product developer. Members of the ecosystem are customers or they fill one of these two roles. There has to be operational procedures for how each transaction, like those shown in Figure 1, will be handled.
Organizational Planning	Clearly the care and feeding of the ecosystem is part of the strategic planning of the company. This requires tough decisions. For example, the decision to participate in an open source project requires giving up some amount of control over the final output. Supporting an elaborate user support network but ignoring feature requests supported by that community is worse than not listening in the first place.
Organizational Risk Management	Many of the most significant risks associated with software development are related to interactions with customers and suppliers. Making the connections with suppliers more explicit will in and of itself reduce risk but it will also enable the collection of more accurate information about deliveries and demand. What's the rhythm of the delivering organizations? How can they be coordinated?
Structuring the Organization	The transaction analysis technique can be useful while the product line organization is being created, but it can also be applied to analyze the external relationships with the rest of the ecosystem. This analysis should be done before establishing the relationship. For example, it has been suggested that Boeing recent problems in delivering the Boeing 787 aircraft are the result of too much out sourcing [Cohan 09].
Technology Forecasting	The ecosystem will be an invaluable aid in forecasting the directions of the many streams of development present in most product line organizations. Exploiting the relationships can provide valuable information about suppliers' plans. If the ecosystem is explicitly managed these ties are on-going and the flow of information continuous.
Training	Some of the relationships in the ecosystem will be sufficiently complex that personnel should be trained to manage them. For example, if the products require certification by a regulatory agency, that agency is part of the ecosystem. Personnel are trained on the regulations and techniques for satisfying them. Those same personnel may participate in forums that give the agency feedback that will shape future versions of the regulations.

3 VARIABILITY AND THE ECOSYSTEM

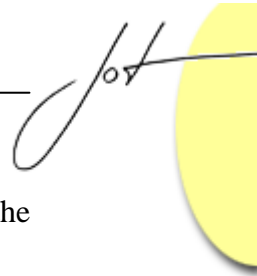
Variability is an essential attribute of a software product line. The variation model makes the differences among products explicit. The variation model is part of the product line roadmap. It provides essential information to the definition of the products that constitute the product line and to the assets used to build the products. The roadmap is a key element in ecosystem success. It provides members of the ecosystem with information that they can use to make decisions about their directions and their interactions with the product line organization.

Modeling is one technique for sharing and coordinating across the ecosystem. A standardized modeling language, such as UML, is particularly useful in communicating across various audiences, but not as necessary anymore. The use of common meta-models and the availability of transformation languages makes the interchange of models written in different languages, provided they share a common meta-model, among members of the ecosystem feasible.

A software product line organization with its ecosystem may have sufficient mass to make it useful to define a UML profile or an even more independent domain specific language [Bell 07][Im 08]. This language can facilitate sharing information within the ecosystem aiding in sharing understanding and even useful for automation of some actions. Projects within Eclipse, for example, support the generation of software engineering tools for domain specific languages.

For example, AUTOSAR, a reference architecture for automotive systems, has been the target of several modeling activities by a number of consortia [AUTOSAR 09]. The Artop.org group is a members-only group that has developed domain-specific tools based on the architecture. This is open to all AUTOSAR member organizations. IBM provides a domain specific modeling environment in its Systems Developer tool [IBM 08]. [Tolavanan 07] discusses Domain Specific Modeling (DSM) in the automotive industry. This is an ecosystem that has been hurt by the economic downturn but these efforts in the ecosystem will reduce costs helping the design portion of the industry to come back as quickly as possible.

Variations are propagated across many of the transactions in the product line's ecosystem. Some of the variations will be resolved as a side effect of the transaction. That is a customer will only want a specific product or will only contract to produce a specific asset needed by the product line. Variations will be bound as part of these transactions. When the assets are passed on to other members of the ecosystem those variations are now seen as common. This traces back to my earlier statement [McGregor 09] that resolving variations is something like partial evaluation [Consel 98]. Partial evaluation resolves some but not all the variability in an asset, this corresponds to having variation points within the same asset but with different binding times. The partial evaluator in this case is some type of transformation engine, such as ATL [ATL 09], that performs a



transformation and produces the partially bound asset needed by the receiver of the transformed asset.

4 MAKING THE ECOSYSTEM VALUABLE

Analyzing and systematically exploiting the potential of the ecosystem provides the opportunity to derive value for the product line organization.

Structure The structure of the ecosystem influences its health and well-being. Many companies that out sourced to the far reaches of the world found that managing the transactions with companies that far away was more costly than the labor savings. The transaction modeling shown in the figures provides a basis for analyzing interactions and determining the implications of certain relationships in the ecosystem prior to implementing those relationships. Figure 2 shows a different structure than shown in Figure 1. By assigning costs to each of the interactions it is possible to make informed decisions about such basic product line questions as whether to have a separate core asset team or not.

A comparison of the structures shown in Figure 1 and Figure 2 is not as straight forward as a manager might think because of interactions of many factors. For example, there are more transfers in Figure 2 than in Figure 1 since having an integrated team requires the core assets produced by each team must be shared with all the other teams. This appears to be more costly (and hence less valuable) than the model in Figure 1. However, the transfers within the integrated teams in Figure 2 may add sufficient value to out weigh the reduction of transfers in Figure 1. The manager conducting this analysis would have to develop a more detailed model to resolve this issue.

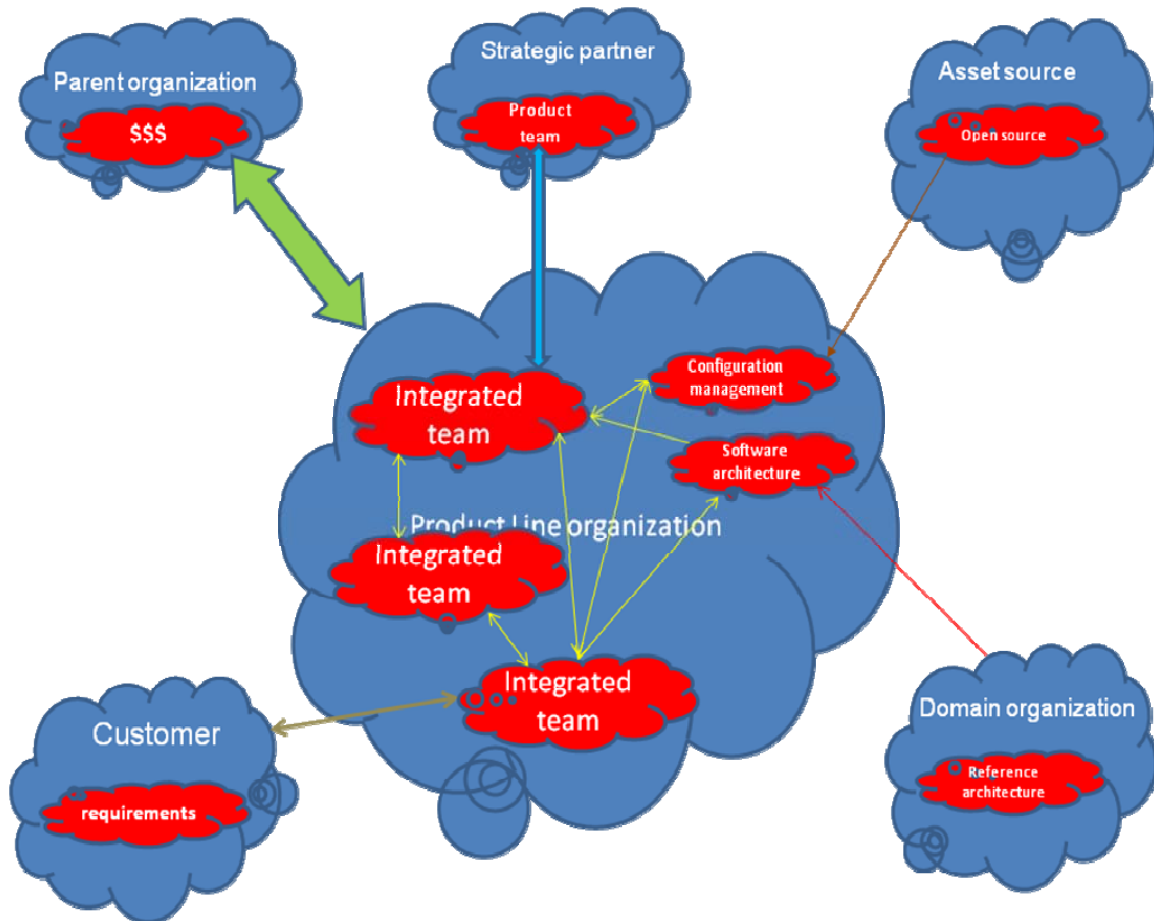
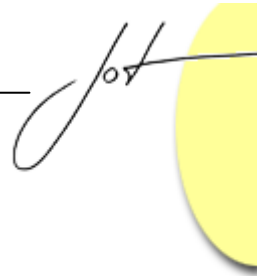


Figure 2 - Alternative Organization

Scope – To an extent a business ecosystem is organic and expands and contracts based on the forces of the market and technology evolution, but a healthy ecosystem requires some planning which requires resources. Management needs to ensure that the ecosystem has the appropriate blend of resources. Planning, such as production planning, provides the opportunity to analyze and balance the ecosystem through Structuring the Organization and managing external partnerships. Transactions require formalisms to define and control the exchange. Even the artifacts exchanged by transfers must be under management of the configuration management system and must be updated as new versions become available.

The ecosystem is a source of

- Ideas and information
- Assets
- Resources



Ideas and Information

Chesbrough's description of Open Innovation gives guidance on how to capture and utilize ideas and information.

“Open innovation is the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively. [This paradigm] assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology.” [Chesbrough 03]

A company's ecosystem includes the sources of these inflows. Of course these days there are plenty of flows, the task is to sort through them to find the useful ones. They range from blogs that are hours old to archival journals that take 2 years to publish information. In a product line organization the multiple development teams deal with multiple customer bases each with its flow of information. Much of that information needs to flow on to the core asset team and the owners of specific assets so the product line manager introduces transfers that carry the information where it needs to be.

Having members of your organization participate in professional activities outside of the standard flow of software development strengthens your ecosystem. Right now there is a new working group forming to investigate defining a variation modeling language. I found out about this effort via an email from one of several lists for which I am registered. By exploiting my own ecosystem I was able to contact the group and arrange for a presentation on this effort at SPLC that was being held the next week. This transfer of information just might result in wider participation in the effort.

Each of us exists within an ecosystem that includes our work ecosystem and our personal ecosystem. Once again an important word is “intentional”. I can influence my personal ecosystem even if I can't totally control it. The people who attend Software Product Line Conference (SPLC) are part of my ecosystem. My partners in Luminary Software are also part of my ecosystem. The companies with which we consult are mostly long-term associations and we have evolved together. I write this column and interact with the readers. I spoke at the Practical Product Lines conference in Amsterdam in October and that invitation came as a result of writing the JOT column. The ecosystem continues to grow and evolve.

The LinkedIn group “Software Product Lines” is an example of a value chain from a professional networking website. Need information to back up a pitch to the boss? Send out a query on one of your networks and chances are you will at least get pointed to a source if not given the data you need directly. Users groups and social networking sites also provide value to an ecosystem. They can be sources of answers to direct queries or discussions on the site by others may expose you to new ideas. User groups such as www.Intuit.com provide access to user forums and a marketplace for related products and services.

Assets

Obtaining assets is an activity that exercises the supply chains in the ecosystem. When I need code libraries, document templates, review checklists, and other usable assets my default action is to first search various networks. By looking at numerous assets from a variety of sources I can more thoroughly explore the commonality/variability of the products. This allows me to obtain an asset that is more robust and generally applicable.

The Make/Buy/Mine practice area in the SEI's Framework for Product Line Practice provides a gateway into each of the three approaches to obtaining assets.

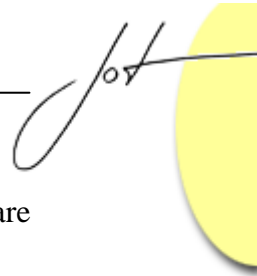
- Making an asset should be the last resort in a product line organization. Some element in the ecosystem should have an asset that meets the need or at least provides a starting point.
- There are several ways to buy an asset. Companies that supply assets become a part of the ecosystem as they provide updates over the life of their product. Professional associations can be a valuable member of the ecosystem and a valuable source of assets. A company subscription to the IEEE Standards library opens the door to numerous generic forms, checklists, etc. Conference proceedings and workshop reports give access to new algorithms and processes.
- By Mining Existing Assets I can populate some portions of the core asset base using work my company has already paid for.

An alternative source of assets these days are open source projects and foundations. Much of the networking infrastructure, the software development tools and other functionality can be obtained from one project or another. Mike Milinkovich describes Eclipse as a source of the necessary infrastructure for products [Milinkovich 08]. Furthermore the market has already proven that these sources are producing high-quality assets. But don't just be a free rider, the greater benefit comes from contributing to the organization even if it is through feedback and feature requests.

Resources

Ianstiti and Levien raise the questions: "how can you promote the health and stability of your own ecosystem, determine your place in it, and develop a strategy to match your role? [Ianstiti 04]" One of their answers to the first question is to use the resources of all elements of the ecosystem. Perhaps the most important resource of all is the people. In a software product line organization there are distinctive roles with very different perspectives and responsibilities. The power of the cumulative knowledge of the people in the ecosystem can be seen by looking at the amount of help available on various product and organization forums.

The ecosystem can play an important role in identifying the resources needed for particular tasks. The networks formed within the ecosystem provide avenues along which are many sources. When security core assets are difficult to complete on time due to lack



of qualified people, a strategic partner may have someone available or even a software asset that will work.

As a member of the ecosystem you should be interested in your position in the ecosystem and work to achieve a position where you can be the most value to the organization and have optimal exposure to new knowledge [Milinkovich 08]. Some people enjoy creating elegant designs while others are interested in solving customer problems. When you look at the 29 practices in Table 1 it is easy to see there are a variety of skills needed in this ecosystem.

Health of the ecosystem

Ianstiti and Levien identifies productivity and robustness as the measures of the health of an ecosystem [Ianstiti 04].

- Productivity – One of the consistent results in software product line research is large increases in productivity. The core assets form the basis for that health. The problem I see most often with core assets is the misconception that only software can be core assets. Many projects spend less than 10% of their total effort on code. If that is the only category of core assets the product line organization will not see the productivity gains that it expects.
- Robustness – How likely is a software product line to survive long term in the face of new technologies, new competitors, and other forces? The Technology Forecasting and Market Analysis practice areas help eliminate surprises in the way of new technologies for building assets and products to be built from assets. Once again the core assets are the basis for good health. The variations built into the assets position the organization to address many types of changes with no architecture changes. All of these practices need to be applied periodically to keep the assets current.

5 SUMMARY

A software product line organization has internal and external dependencies and interactions. The external dependencies link your organization to other organizations that each have their own goals and strategies. Strategic decisions made by the product line organization may not have the desired effect if the other members of the ecosystem make counter-acting decisions.

Informal communities and formal consortia are increasing in number as managers seek to strengthen their position by having allies. The ecosystem of a software product line organization can help it to grow and be successful or it can constrain that success. The care and feeding of your ecosystem is not something to be left to chance. Be strategic, manage your ecosystem intentionally so that strategies and decisions drive you in the correct direction and they are complementary not contradictory. Avoid wasting resources on partners that have little or no value to add to your organization. A

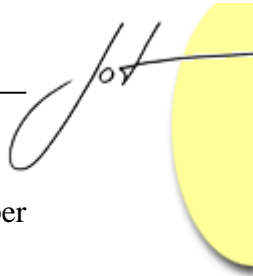
strategically managed and nurtured ecosystem can add tremendous value to your organization but it will not happen by accident.

ACKNOWLEDGEMENTS

My thanks to Soujanya Vullam, Clemson University, for her comments which greatly improved this column.

REFERENCES

- [ATL 09] Atlas Transformation Language, www.eclipse.org/m2m/atl
- [AUTOSAR 09] Automotive Open System Architecture, AUTOSAR, www.autosar.org, 2009.
- [Baldwin 07] Carliss Y. Baldwin. Modularity, Transactions, and the Boundaries of Firms: A Synthesis. Harvard Research Report 08-013, 2007.
- [Bell 07] Peter Bell. A practical high volume software product line, Conference on Object Oriented Programming Systems Languages and Applications, 2007.
- [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.
- [Chesbrough 03] Henry Chesbrough, *Open Innovation: The New Imperative For Creating and Profiting from Technology*, Boston: Harvard Business School Press, 2003.
- [Clements 01] Paul Clements and Linda Northrop: *Software Product Lines: Practices and Patterns*, Addison-Wesley, 2001.
- [Cohan 09] Peter Cohan. Will Boeing delay the 787 Dreamliner another two years? <http://www.dailyfinance.com/2009/08/19/will-boeing-have-to-delay-the-787-dreamliner-another-two-years/>, 2009.
- [Consel 98] [C. Consel](#), [L. Hornof](#), [R. Marlet](#), [G. Muller](#), [S. Thibault](#), [E.-N. Volanschi](#), [J. Lawall](#), and [J. Noyé](#). Partial evaluation for software engineering, ACM Computing Surveys (CSUR) Volume 30 , Issue 3 (September 1998).
- [Eclipse 09] Eclipse Foundation, www.eclipse.org.
- [Iansiti 04] [Marco Iansiti](#) and [Roy Levien](#). Strategy as Ecology, Harvard Business Review, March 2004.



-
- [IBM 08] IBM, AUTOSAR System Modeling in IBM Rational Systems Developer http://www.ibm.com/developerworks/rational/library/08/0812_paul-rooks/
- [Im 08] [Kyungsoo Im](#), [Tacksoo Im](#), and [John D. McGregor](#). Automating test case definition using a domain specific language, Proceedings of the 46th Annual Southeast Regional Conference, 2008.
- [Koch 08] Luc Koch. Keynote address at SPLC 2008.
- [McGregor 08] John D. McGregor "Agile Software Product Lines, Deconstructed", in Journal of Object Technology, vol. 7, no. 8, November - December 2008, pp. 7-19. http://www.jot.fm/issues/issue_2008_11/column1/
- [Milinkovich 08] Mike Milinkovich. A Practioners Guide to Ecosystem Development, http://www.eclipse.org/community/training/webinars/081015_Ecosystems_Webinar.pdf, 2008.
- [Moore 05] James F. Moore, Antitrust Bulletin, Fall 2005
- [OMG 09] Object Management Group. www.omg.org, 2009.
- [Porter 98] Michael E. Porter. Competitive Strategy, Free Press, 1998.
- [Tolvanen 07] J.P. Tolvanen and Cord Giese. Modeling for Software Development in the Automotive Industry, Eclipse Magazine, July 2007.
- [Wessilius 08] Jacco Wessilius. The Bazaar inside the Cathedral: Business Models for Internal Markets, (2008) 60 – 66.

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.