We’d rather not XMILE: model interchange standards and software competition

Leonard Malczynski

P.O Box 5800
Department 6115 Geohydrology
Sandia National Laboratories
Albuquerque, NM 87185-0735
505-844-7219
lamalcz@sandia.gov

Abstract

There have been a number of articles describing the desire, by the system dynamics computer modeling community, to establish a common model exchange language (XL) for system dynamics models. Contrary to the seeming increase in the common good resulting from such an exchange language, as of today, none of the major vendors has stepped forward to implement this exchange language. We propose to demonstrate, within a strategy framework, the consequences of implementing the XL and the reasons why the apparent disinterest shown by the software community in the XL is intrinsic to markets of this type.

Introduction

As early as 1995 (Myrtveit) the system dynamics community proposed a model translation format and not a translation engine. Diker and Allen (2005) summarized the nature and reasons for an XL. They assume that the translation engines will arise independently. As of today, no independent group or system dynamics software modeling vendor has implemented an engine. Community discussion has often raised the issue that the construction of the translation engine will be difficult and of course moot if only one vendor implements it.

Diker and Allen point out that an XL will:

1. Reinforce peer review
2. Lower barriers to entry for valuable niche software
3. Foster increased collaboration between different research segments that have developed around specific software
4. Generate a standard for system dynamics models
5. Facilitate model conversion
6. Facilitate collaborative modeling

I am not concerned in debating the benefits listed above. Nevertheless, they must be addressed. The demand for an XL is driven by the desire to obtain the stock of models[1] from one or more platforms in a format compatible with the user’s current platform. Currently, as the stock of desired models in a non-compatible format increases, the desire for an XL may increase to the point that a customer may want to switch to another platform. First, there should be a clear distinction between model and application. An application is the sum of a model plus its interface. An interface is the conscious decision to expose portions of the model to user manipulation and examination. A common application of the Pareto principle to software states that typically a user interface will occupy 80% of the total programming effort. None of the XL proposals addresses the issue of interface conversion. Second, the number of desired models in a format incompatible without an XL to the potential customer’s current format is debatable. Many

[1] For clarity, models are considered system dynamics software models, not mental models or simply causal loop diagrams.

of the classic system dynamics models such as World3, Urban Dynamics, and the Molecules suite are available in at least two formats. In addition, the openness of the system dynamics community tends to make many models available. This happens via user groups, leading texts[2], model collections[3], or by a simple request to a list server member. Finally, as vendors compete they tend to introduce extra-methodological and ease-of-use features in to the software. Of course there are diminishing marginal returns to the addition of features to software, nevertheless consumers incorporate these new features into their models and applications. This makes model translation even more difficult.

Diker and Allan do say that, "Development of such an interchange language … requires a lot of time and effort. This problem can be couched in terms of standardization. Grindley (1990) opines that the standards game typically has a victor, the firm commands the largest installed base. The standards issue is not one of standards bodies and industry associations. "… there are no substitutes for market processes in deciding standards contests." (Grindley, pp 80). A brief examination of the system dynamics software market follows.

Coyle (1996) classified the known SD software of the time into two groups, those that are primarily text based (DYNAMO, COSMIC and DYSMAP) and those that are primarily graphics based (Powersim, iThink and Vensim). He goes on to give his opinion on the relative suitability of each. However, to my knowledge, there has never been a thorough public comparison of the three leading products. [4] The market for system dynamics software is now dominated by three firms providing graphics based tools, isee systems, Powersim, and Ventana, an oligopoly from an economist's point of view. Historically commercial versions were released as follows: Stella - 1985, Vensim - 1991 and SimTek – 1988 (Studio's original product name). [5] All three firms currently have several versions that offer varying degrees of capability for different prices. [6] Of the three, only Vensim has a translation utility from older iThink/Stella models to Vensim format. [7] All three can produce a text file that describes a model in a regular, well defined format. Most of the software is upward compatible, that is, internal conversion routines address the issue of converting older models from the same software platform to the current platform. [8] Finally, there is, of course, the methodological similarity of the software modeling languages. [9]

Demand relationships among products

The decision to purchase one of the three leading, or any, system dynamics software will depend upon a budget constraint and what Lancaster (1966) called the "attributes" of the goods. In this situation the attributes are the characteristics or capabilities that are a part of the software. The number of attributes can be quite large[10].

Coyle (1996) proposed, in general, a dynamic simulation software package needs to be assessed according to:

- its basis in fundamental system dynamics theory;

[2] John Sterman's, Business System Dynamics, for example.
[3] Creative Learning Exchange, MIT Roadmap.
[4] A System Dynamics list serve request for such a comparison has been unanswered since 2001.
[5] See Vensim's history at www.vensim.com and Powersim's history at www.powersim.com
[6] The software landscape is quite varied. Some of the firms suggest the purchase additional tools for the creation of flight simulators or microworlds.
[7] Powersim Studio has 'HELP' references that aid in the manual translation from DYNAMO and iThink/Stella models.
[8] None of these conversion tools is flawless. Software architectures, syntax, and features change over time.
[9] This similarity applies to the simplest of models, e.g. Molecules. Each of the vendors may be attempting to capture a specific portion of the market by providing unique, extra-methodological features. This is addressed later in the paper.
[10] Interface tools, causal tracing, ability to export and import data, ability to add extra-methodological features, function breadth, and the quality of the Interactive Development Environment (IDE) are examples.

- the ease with which it can be used;
- the support it gives to model building;
- the extent to which models can be documented and explained to a customer;
- the facilities it has for debugging a model;
- the ease of making experiments and producing output; and
- the scope of its facilities for policy design.

Figure 1 shows an example of utility maximization for a two attribute model.

In this example there are 3 products, **X**, **Y**, and **Z** with two attributes, *a1* and *a2*. Each of the products can be differentiated by the quantity of *a1* and *a2* they contain. With a given budget constraint, a consumer may purchase at **X***, **Y***, or **Z*** or along any of the points on the lines **X*-Y*** and **Y*-Z***. The curves **U_o** and **U'_o** show the utility curves for two different consumers who have different preferences for *a1* and *a2*. Their respective purchase combinations E and E' show different combinations of **Z** and **Y** for E and **Y** and **X** for E'.

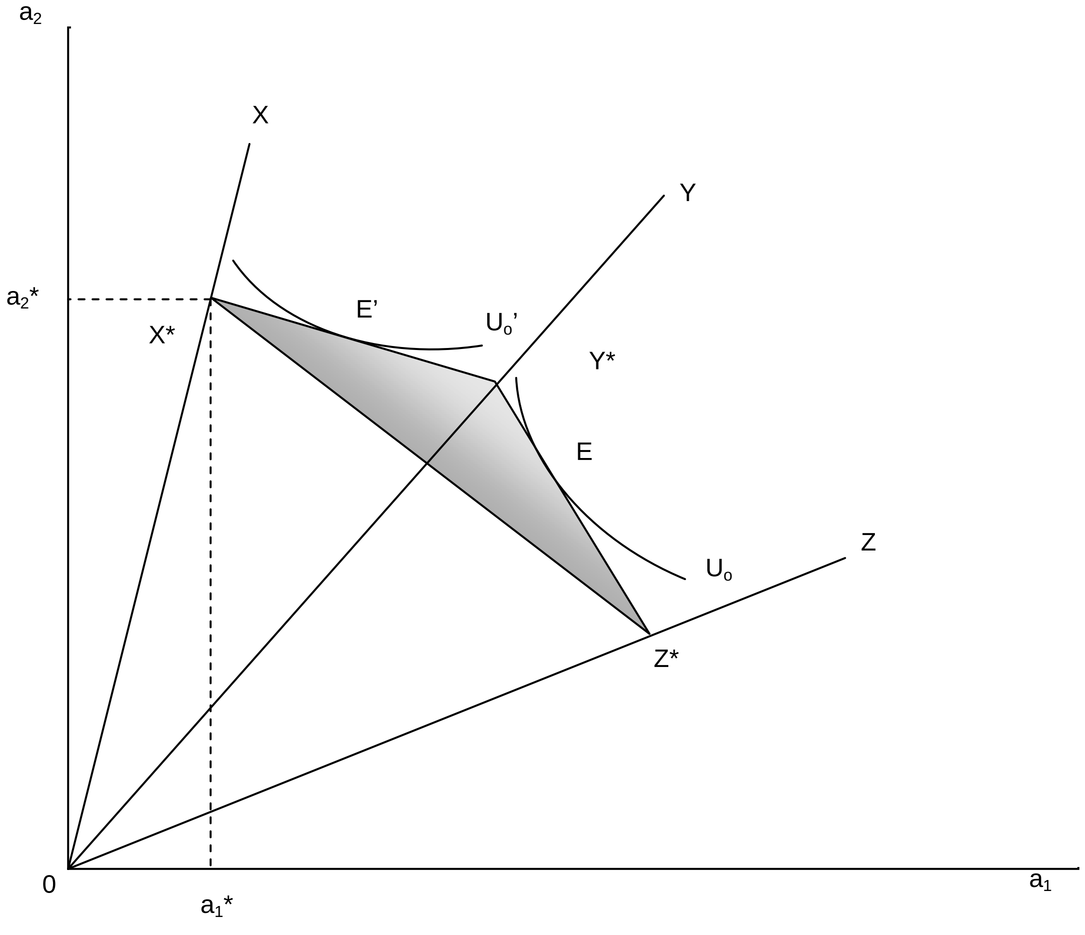

Figure 1. Utility maximization in the attribute model. (Adopted from Nicholson, 1995)

Present day consumers of system dynamics software are faced with a slightly different problem. The attribute levels are discontinuous in system dynamics modeling software. Each vendor typically offers different versions of the same product. Each of these products is aimed at a niche market that is defined by the purchaser’s budget constraint. In that way many segments of the market may be captured by one vendor. Most modelers are budget constrained by a threshold that limits the purchase to 1 of the 3 leading products. In that case, one single attribute [11] typically takes precedence in the purchase decision. In some cases, the system dynamics modeler makes the choice of purchasing different levels of the same attribute by splitting their budget between a version from one vendor and another version from a second vendor, resembling a typical Lancaster choice. For example, one product may be limited in its interface capabilities but strong in causal loop diagramming; the second product has the opposite strengths, the modeler may purchase one of each of the lesser cost versions.

In the strategy dynamics analysis of this problem we will limit the attributes to two, functions available and an ability to read/write the XL. Looking back at Figure 1 we would let *a1* represent functions available and *a2* represent XL capability. In the industry as a whole there are several other issues to consider. First, there are smaller firms, new entrants, trying to obtain market share. Recently one vendor has added to their functional attraction by offering extra-methodological features, namely agent based capabilities. Second, there is no consideration of substitutes in this market. Either the software is methodologically consistent or not. Finally, there is no consideration of external forces that influence this market place. Some pundits have stated that 2006 is the ‘tipping point’ year for system dynamics. In that case software vendors should be able to easily deliver product but must consider the inflow of large numbers of neophytes on their support resources.

Strangely, XL capability is not currently an attribute of any of the three leading products. It should be obvious that at least two vendors must implement XL to be of any value to consumers. Note, that in the absence of an XL one can substitute optimization, interface capabilities, or any of Coyle’s categories for *a2* in the utility maximization treatment above.

Software economics and switching costs

Typically, switching costs have been divided into three categories:

- **Informational:** The training costs associated with learning the new product
- **Transactional:** Costs associated with completing the transaction of switching from one vendor to another.
- **Contractual:** Firm’s actions that create switching costs.

Burnham et. al. (2003) define switching costs as “*the onetime costs that customers associate with the process of switching from one provider to another*” *they* note that “the switching costs that reduce customers’ desire to leave an incumbent provider” have been little studied.

A Typology of Consumer Perceptions of Switching Costs

Procedural Switching Costs	Financial Switching Costs	Relational Switching Costs

[11] Note that the inclusion of an XL in a product does not guarantee any additional purchases since it takes two vendors (or a third party) to implement the XL for it to have any success in lowering switching costs.

- Economic Risk Costs
- Evaluation Costs
- Learning Costs
- Setup Costs

- Benefit Loss Costs
- Monetary Loss Costs

- Personal Relationship Loss Costs
- Brand Relationship Loss Costs

Figure 2 from Burnham 2003

In the XL framework, the industry-wide existence of XL would in the long run reduce the financial switching costs of software to near zero. Consumers that do not switch vendors, i.e. those that only take advantage of the XL, would not incur procedural or relational switching costs. Procedural switching costs would remain for those consumers switching vendors to take advantage of a particular vendor's offering that increases their modeling utility and lets them make use of their pre-existing stock of models. [12] Of course, they may incur relational costs.

National Economic Research Associates (2003) propose explanations for encouraging and discouraging compatibility of products. Among the incentives for compatibility are demand expansion effects, as compatible product demand increases the entire market increases, while incentives for incompatibility, include smaller firms typically favoring standards so they can gain market share and intellectual property protection, especially important as the XL may have to match changes in vendor's offerings..

Floribert (1993) addressed many of these issues in a system dynamics framework. The following causal loop diagram explains a generic duopoly with switching costs.

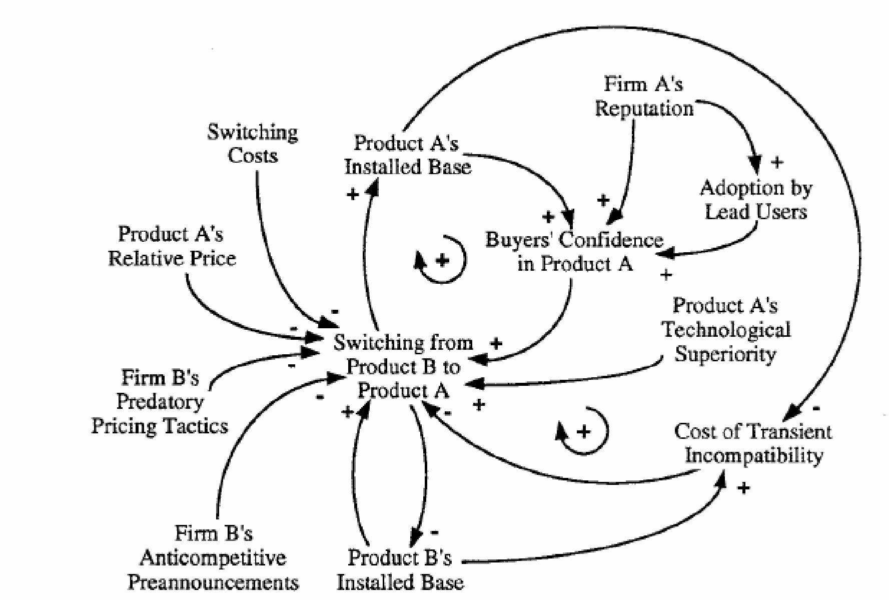
Firm A's Reputation
Switching Costs
Product A's Installed Base
Adoption by Lead Users
Product A's Relative Price
Buyers' Confidence in Product A
Product A's Technological Superiority
Switching from Product B to Product A
Firm B's Predatory Pricing Tactics
Cost of Transient Incompatibility
Firm B's Anticompetitive Preannouncements
Product B's Installed Base
+
-

Figure 3 from Floribert 1993

Floribert focuses the CLD around the switching rate from Product B to Product A. Note that the switching rate is a member of two reinforcing loops. The first loop involves Products A's installed base. As that base increases Costs of Transient Incompatibility decrease and then

[12] It is important to recall that model + interface = application.

decreases the volume of switches. The second reinforcing loop involves Buyer's Confidence in Product A that influences consumers of Product B to switch to Product A.

Competitive strategy and the exchange language

Using Warren's typology (2002, pp. 157-158) there are 3 forms of rivalry,

Type 1 – the struggle to develop potential resources
Type 2 – the tug of war to switch resources away from competitors and prevent the reverse; and
Type 3 – the struggle to win a share of attention from customers and other resources that may be shared with competitors

The strategic issue

The primary issue here is whether a system dynamics software vendor should allocate any of its development resources to building an XL. Building an XL would require that two distinct functions be added to the software, an export engine to produce the common format and an import engine to produce the common format. The system dynamics community has long assumed that the format of the XL would be determined by the community and not by a particular vendor. The XL case deals primarily with Rivalry Types 1 and 2. Although many customers may be purchasers of at least two of the leading products (primarily due to the lack of an XL) I will not deal with Type 3 rivalry here. In addition I also assume that there is little, if any, competition for development resources among the three firms. This is primarily due to the historical accidents of geography (Powersim is located in Norway) and computer language (different development from Pascal to .COM and platforms Macintosh and Microsoft Windows).

Software, unique among many products and services, typically exhibits a production function of decreasing marginal costs. The first copy of the software is typically very expensive and each additional copy is marginally much less, often just the cost of duplication. Software has a dark side in that customer support can often make or break the firm. Customers come in the proverbial '*many shapes and sizes*' and require varying degrees of support that drain resources from the development effort. One of the strategies to lessen the support burden is to encourage, or at least not hinder, the rise of user groups. [13] User groups can handle neophytes and expert users through list servers and model repositories. Support is not included explicitly in my model.

Assuming that the majority of customers are system dynamics practitioners, the winning of potential customers has much to do with developing their pre-cursor, ever likely customers. Ever likely customers are introduced to vendor's offerings and system dynamics through both training and education.

Rivalry

Capabilities: related to tangible resources?

This examination of rivalry in the system dynamics software market suffers from a lack of primary data. Although I know the launch dates, number of functions, and user group activity of the firms basic resource stocks and their respective development rates are unknown.

[13] As of 16 March 2006, the Vensim Forum has 404 members (includes the System Dynamics Discussion Forum members) with the earliest posting being 3-15-2003. The Powersim yahoo group (independent of Powersim) has 312 members with the earliest posting being 1-6-2004. isee systems has no specified user group.

Causal loop analysis

Resource-system

Figure 3 gives a causal loop explanation of a two-firm rivalry focused on the issues associated with XL. In order to simplify the analysis several variables remain exogenous and not causally determined namely price and marketing expense.

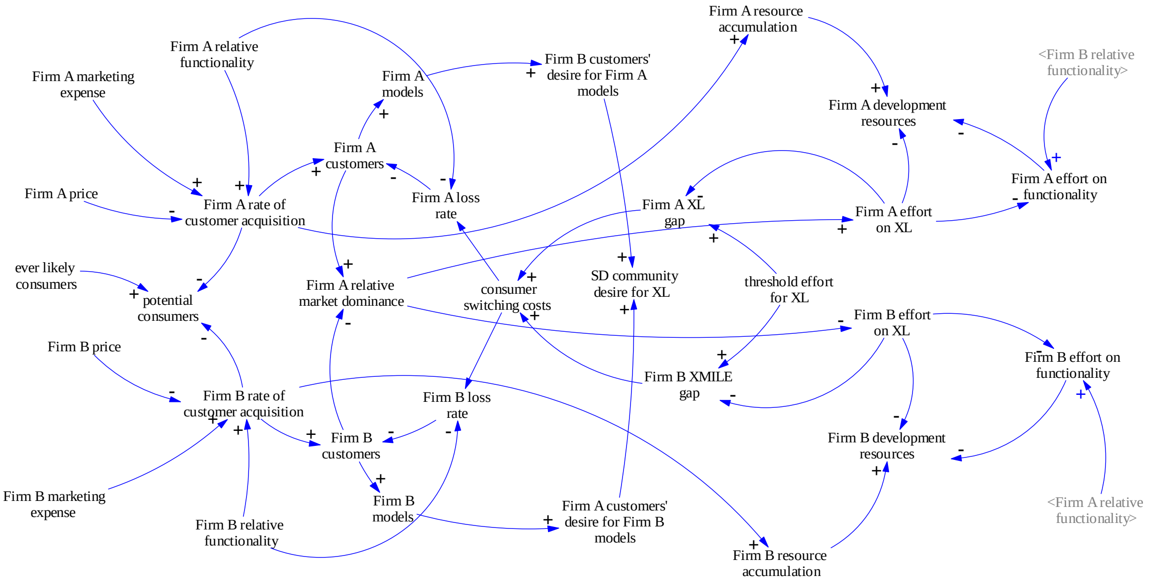
Firm A resource accumulation
Firm A relative functionality
<Firm B relative functionality>
Firm A marketing expense
Firm A models
Firm B customers' desire for Firm A models
Firm A development resources
Firm A customers
Firm A effort on functionality
Firm A price
Firm A rate of customer acquisition
Firm A loss rate
Firm A XL gap
Firm A effort on XL
ever likely consumers
potential consumers
Firm A relative market dominance
consumer switching costs
SD community desire for XL
threshold effort for XL
Firm B effort on XL
Firm B price
Firm B effort on functionality
Firm B XMILE gap
Firm B rate of customer acquisition
Firm B loss rate
Firm B customers
Firm B development resources
Firm B marketing expense
Firm B models
Firm A customers' desire for Firm B models
<Firm A relative functionality>
Firm B relative functionality
Firm B resource accumulation

Figure 3 Two-firm competition and XL

Interpretation

The causal loop diagram includes several dynamic hypotheses concerning firm behavior. Let's consider the hypothesis that firm success depends upon customer purchase of software. In order to encourage these sales firms spend their resources on development efforts and marketing. Allocating any resources to the XL can only decrease the number of resources allocated to improving product functionality. A decrease in functionality then leads to decreased sales. Likewise if the total marketing resources are diminished by effort put into the XL.

The desire by consumers for an XL is dependent upon the stock of models in another software format. As the number of models unavailable increases, the desire for an XL increases. Note that the variable, '*SD community desire for XL*' has no causal relationship to any other model variable. This is true because a firm will invest in XL only if that investment is advantageous in gaining or retaining customers. The XL development decision is not dependent upon the community desire for an XL. In fact, maintaining high switching costs is advantageous to each firm, marginally more so to the firm that is the weakest competitor in functionality.

Driving the '*threshold effort for XL*' to zero is akin to having an outside, neutral party develop the XL. Here I hypothesize that the community over time will migrate to the vendor that provides the most functionality per dollar. In the simple example in Figure 1, only two attributes influence a customer's purchase decision. One XL is adopted by all vendors, marketplace dynamics will tend to create an overall winner. Myrtveit (1995) alludes to the technical, not business, similarity of Microsoft's Rich Text Format (RTF) and an XL. Note that Microsoft invented RTF in 1987 and now controls the word processor application market.

Finally, as firms compete they follow the leader. A simple function count extracted from the function lists of the three major software vendors shows isee systems' iThink with 80, Powersim's Studio with 183, and Ventana's Vensim with 152. [14] iseesystems' website is forthcoming with future development plans that include the addition of several functions that exist in Studio and Vensim.

Support for my hypothesis comes from the field of economics. Support for the competition killing effect of high switching costs may be found in Besson and Farrell (1994), Beggs and Klemperer (1992), Burnham et. al., Farrell and Shapiro (1988), Floribert (1993), Gandal (1995), Hess et. al. (2002), Katz and Shapiro (1996), Klemperer (1987, 89, 95) and the NERA (2003). Using non-system dynamics techniques, both Beggs (1992) and Shapiro empirically support my hypothesis. Beggs analyzes a multi-period duopoly and finds that prices are higher than without switching costs. That result is the status quo in the SD software market. Driving switching costs to zero would drive consumers to the lowest cost per function vendor above a threshold function level. Once switching costs are zero, vendors would have to compete by adding functionality and/or lowering prices. Of course the XL would have to simultaneously keep up with new functional capabilities. In that case products would have to be less differentiated since new functions from Firm A would have to be adopted by Firm B while both firms expend effort on XL that could be used to develop new functions. Beggs also shows that new entrants benefit from high switching costs although they still have the Type I rivalry issues to deal with.

Model results

Having only outsider access to information about resources and capabilities [15] have constructed a two-firm model in Powersim Studio 2005 (Figure 6) in order to simulate rivalry issues that focus on the XL. The two firms are 'representative' of the leading system dynamics software vendors.

Key parameters of this model include:

1. Purchase decision is based solely on software functionality.
2. Firms A and B start offering their product in 1985 and 1992 respectively. Each software launch starts with 50 units of functionality.
3. Diversion of software engineering effort away from functionality to the XL begins in 2007.
4. No price or marketing effects are included.
5. There are two sets of model resources, those basic models that are readily transferable between software platforms and those that contain each vendor's enhanced functionality.
6. Once the XL is completed the firm with the most enhanced models and functionality draws 90% of the other firm's new customer acquisition.

Experimentation with the model shows that if functionality (i.e. functions per software package) is the main determinant of demand then the firm that most rapidly develops that resource will be the victor. Rapid development not only makes the software more effective but those functions used by customers produce a pool of enhanced models. That pool of models is attractive to users of competing software. Once the XL is available, switching costs essentially revert to the

[14] Of course the functions across platforms may be exclusive, e.g. several vendors implement the same function differently and some have functions unique to themselves.

[15] Start dates and function counts were available. I declined to use user group membership as a proxy for the number of customers held by each firm.

purchase price of the new tool. In addition that consumer's current models will port to the new tool thanks to the XL.

One way to test the underlying relationships is to increase the number of software engineering resources available to the firm with the lower function base. Assuming that engineers are equally productive it would take a quadrupling of Firm A's software engineering resources after 2006 and a large increase (25 models to 100 models) as the threshold number of enhanced models of Firm B desired by Firm A in order to improve Firm A's customer base. Essentially, the current market position is difficult to change given the lead by Firm B in functionality.

Equally ineffective is the setting of XL development costs to zero. In that case neither firm devotes effort to XL, the XL is constructed by a trusted outside party. Since Firm B had the lead in functionality by a wide margin it could build its own translation engine, not an XL, but the full capacity to import models from other vendors from their exportable format. Diker and Allen attempted such a task with success for one vendor's software export format.

Figures 4, 6 and 7 show output from the base case run of the model.

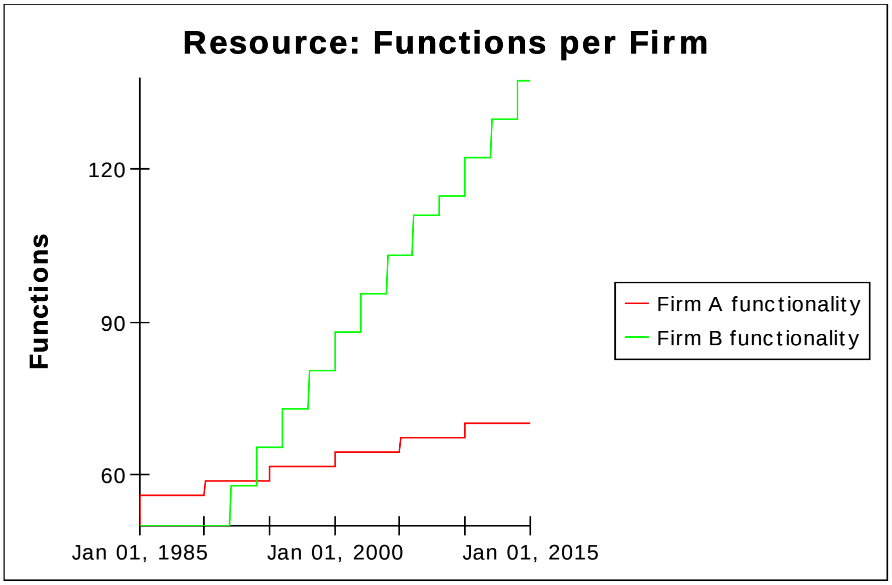

Figure 4 Software functionality as a resource

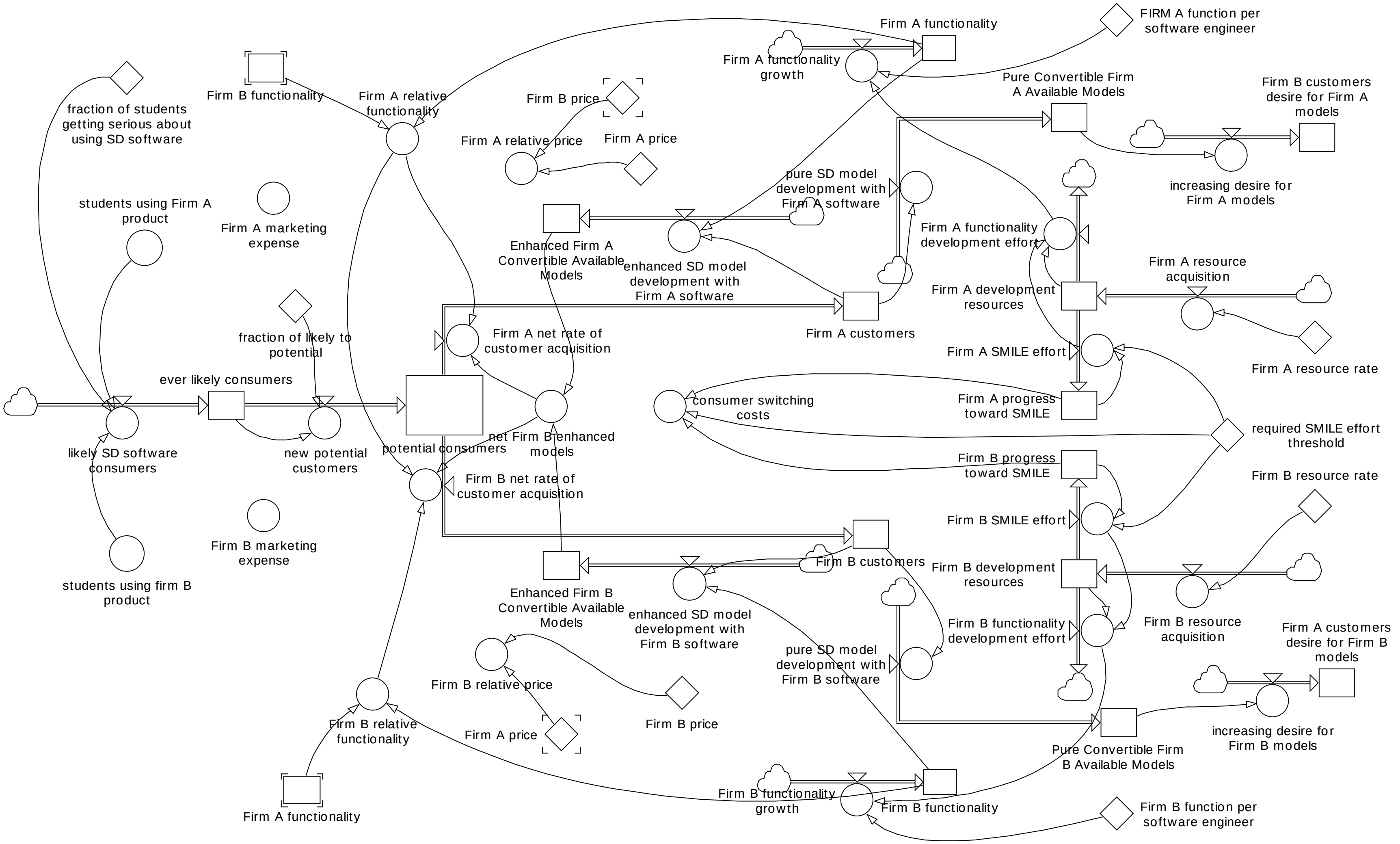
FIRM A function per software engineer
Firm A functionality
Firm A functionality growth
Firm B functionality
Firm A relative functionality
fraction of students getting serious about using SD software
Firm B price
Pure Convertible Firm A Available Models
Firm B customers desire for Firm A models
Firm A relative price
Firm A price
increasing desire for Firm A models
pure SD model development with Firm A software
students using Firm A product
Firm A marketing expense
Firm A functionality development effort
Enhanced Firm A Convertible Available Models
enhanced SD model development with Firm A software
Firm A resource acquisition
Firm A development resources
Firm A customers
fraction of likely to potential
Firm A net rate of customer acquisition
Firm A SMILE effort
Firm A resource rate
ever likely consumers
Firm A progress toward SMILE
consumer switching costs
required SMILE effort threshold
potential consumers
likely SD software consumers
new potential customers
net Firm B enhanced models
Firm B progress toward SMILE
Firm B net rate of customer acquisition
Firm B resource rate
Firm B SMILE effort
Firm B marketing expense
Firm B customers
Firm B development resources
students using firm B product
Enhanced Firm B Convertible Available Models
enhanced SD model development with Firm B software
Firm B functionality development effort
Firm B resource acquisition
Firm A customers desire for Firm B models
pure SD model development with Firm B software
Firm B relative price
increasing desire for Firm B models
Firm B price
Firm B relative functionality
Firm A price
Pure Convertible Firm B Available Models
Firm B functionality growth
Firm B functionality
Firm A functionality
Firm B function per software engineer

Figure 5 Rivalry Model

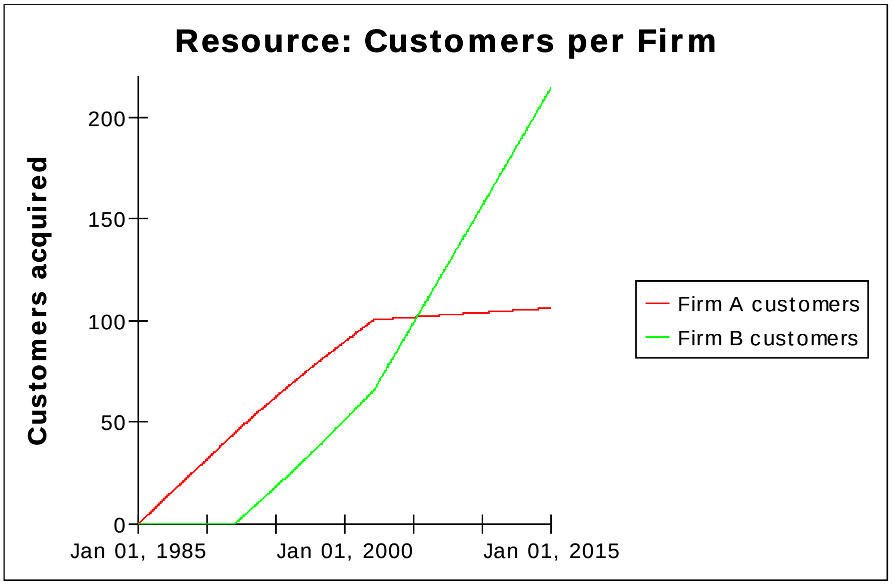

Figure 6 Customers as a resource

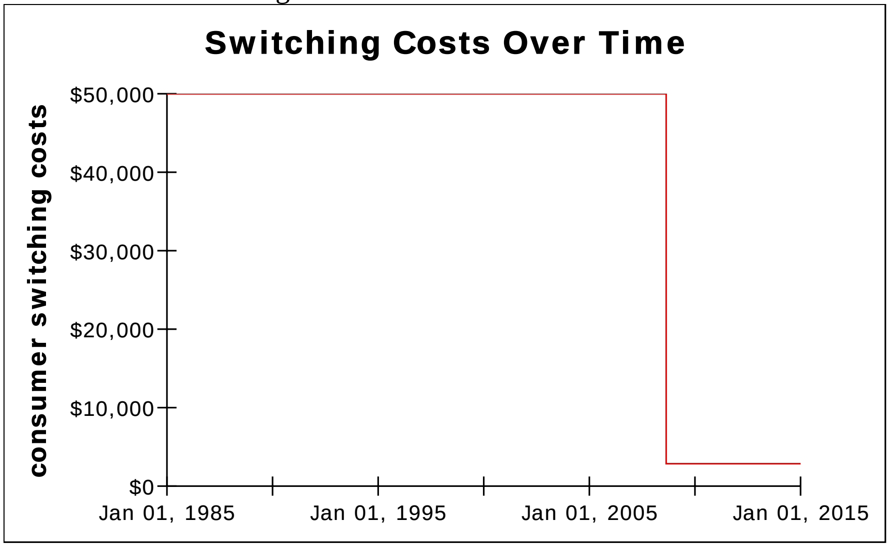

Figure 7 Switching costs to consumers of Firm A software

Conclusion

The strategy posed to system dynamics software firms is whether or not to allocate resources to the development of an exchange language. An exchange language would facilitate the transfer of system dynamics models from one vendor's software platform to another. If I ignore the stock of models that could be seamlessly transferred, the system dynamics community would benefit from an XL for several reasons.

Analysis shows that the decision to allocate resources depends upon:

1. the level of effort required to develop the XL
2. the relative resource base of the vendor in customers and models
3. the relative cost per function that the software provides
4. and all factors that control the growth of these resources

Recommendations

Given the complexity of this decision for the SD software firms and the resulting market turmoil that may ensue I recommend that:

1. A trusted outside party should be responsible for the development of XL. Unfortunately, that will only lessen the resource cost to the software firms involved however since the outside party will need access to proprietary information provided by each vendor.
2. The vendor that decides it either has the software engineering surplus (i.e. the leader in functionality) or the largest customer base take on the community, agreed-to format and produce XL.
3. An alliance can be formed between two or more firms to build the XL knowing that each firm's product occupies a specific niche in the system dynamics modeling life cycle. This of course assumes that the typical customer has the resources to afford multiple products.

In either case I believe that the market will eventually favor that firm that has the highest degree of functionality at the lowest cost regardless of marketing, educational use introduction of the software, or industry alliances. All this discussion may be moot; it may be that what makes people smart is an intelligent person manipulating the right tools (Norman, 1993). In that case it is the consulting success of the system dynamics software firm that determines its market success no matter what, if any, system dynamics software is applied.

References

Beeson, Stanley M. and Joseph Farrell, "Choosing how to compete: Strategies and Tactics in Standardization", The Journal of Economic Perspectives, Vol. 8, No.2 (Spring 1994), 117-131.

Beggs, A. and P. Klemperer (1992). "Multiperiod Competition with Switching Costs." Econometrica **60**: 651-666.

Burnham, Thomas A., Judy K. Frels, and Vijay Mahajan, "Consumer Switching Costs: A Typology, Antecedents, and Consequences", Journal of the Academy of Marketing Science, Vol. 31 No. 2, 109-126.

Coyle, R.G., System Dynamics Modeling: A practical approach, Chapman & Hall, 1996.

Diker, Vedat G. and Robert B. Allen, "XMILE: towards an XML interchange language for system dynamics models", System Dynamics Review, Vol. 21, No.4 (Winter 2005): 351-359.

Farrell, J. and C. Shapiro (1988). "Dynamic Competition with Switching Costs." RAND Journal of Economics **19**: 123-137.

Floribert, Françios, "Dynamics of Diffusion and Competition in the Market for Software" MS Thesis, MIT 1993.

Gandal, Neil. "Competing Compatibility Standards and Network Externalities in the PC Software Market" Review of Economics and Statistics, Vol. 77, No. 4 (Nov 1995), 599-608

Grindley, Peter, "Winning standards contests: using product standards in business strategy", Business Strategy Review Spring 1990, 71-84.

Hess, Mike and Joan Eric Ricart, "Managing Customer Switching Costs: A framework for competing in the networked environment" Research Paper No 472, October 2002, IESE Research Division, University of Navarra, Barcelona, Spain.

Katz, Michael L. and Carl Shapiro, "Systems Competition and Network Effects" The Journal of Economic Perspectives, Vol. 8, No.2 (Spring 1996), 93-115.

Klemperer, P. (1987). "The Competitiveness of Markets with Switching Costs." RAND Journal of Economica **18**: 138-150.

Klemperer, P. (1987a). "Markets with Consumer Switching Costs." The Quarterly Journal of Economics **102**(2): 375-394.

Klemperer, P. (1987b). "Entry Deterrence in Markets with Consumer Switching Costs." Economic Journal **97**: 99-117.

Klemperer, P. (1989). "Price Wars Caused by Switching Costs." Review of Economic Studies 56: 405-420.

Klemperer, P. (1995). "Competition when Consumers have Switching Costs." Review of Economic Studies 62: 515-539.

Myrtveit, Magne, “Models crossing the boundaries of tools”, ISDC ’95 Conference Proceedings

Nahm, J. “The Effects of Backward Compatibility on Technology Adoption in Systems Markets”, 2005

National Economic Research Associates, “Switching costs”, Economic Discussion Paper 5, Part one: Economic models and policy implications, April 2003.

Nicholson, Walter, Microeconomic Theory, Dryden Press 1995

Norman, Donald A., Things That Make US Smart, Perseus Books, 1993.

Warren, Kim, Competitive Strategy Dynamics, John Wiley & Sons Ltd. 2002