A Holistic Model for Enterprise Development (III)

Hong Li
Water-Logic, Inc., Phone: 614-296-7644, Fax: 866-206-9568, Email: hong@water-logic.com

Abstract: The life cycles of products and services, as well as their providers and customers, can be all treated as enterprise entities on a chain of enterprises that is causal by nature. Once the development programs of the products, the services, and the enterprises where the products and services are generated or consumed are modeled under Aristotelian-Rosennean causal framework, the underpinning relations within and between enterprises on the enterprise chain can be studied to reveal key institutional patterns, organic vs. inorganic, which will either facilitate or limit the enterprise capabilities to change. This study of PERA concludes that organic approach as defined by Rosen is not an option but necessity in sustainable enterprise development in a changing environment.

Keywords: Enterprise Engineering, Systems Engineering, Relational Model, Complexity Theory, Enterprise Reference Architecture, PERA (Purdue Enterprise Reference Architecture), Systems Science

1. Introduction

The M-R model of Robert Rosen presents the key entailment relationships associated with the three bio-functions, metabolic, repair, and replicate, which make organisms alive from within. The previous two articles of this series have demonstrated that the same relational paradigm in terms of Aristotelian-Rosennean (A-R) causal framework [Rosen, 1991 and 2000] is also embedded within PERA (Purdue Enterprise Reference Architecture) [Williams, 1989, 1992, 1996, and 1998] life cycle as a model for enterprise development. As a result of organic enterprise development, an organic enterprise represents an individual institution where strategic development and associated executions become a coherent whole based on institutional and mutual relationships between management and operations. A machine enterprise on the other hand is an individual institution where strategic development and strategy executions remain separated, and merely hierarchical and command-and-control relationships are maintained between management and operations.

Based on this study of enterprise development, with help of Rosen, the following conclusions can be further reported through this series of articles:

- *Basic concept of organic development should be reconsidered.*
  The current concept of organic approach in enterprise development should be reconsidered. The characteristics of open systems, iterative and incremental development, or life cycles, etc. epistemologically represent certain aspects of organic behaviors. But no ontological insight can be drawn from them to explain why organic systems are intrinsically different from inorganic ones since the same characteristics can be found in inorganic ones as well. If the end result of organic development had nothing to do with the inner capabilities but were randomly decided by blind natural selection of business environment, the conscious efforts made by human agents in enterprise development would have made no difference.
- *Machine model applied in enterprise development may only offer limited flexibility in enterprise operations.*
  The metabolic function of the M-R model can be considered equivalent to the common operations function in enterprise development, and the repair function equivalent to the common strategy function. Without the replicate function, the combination of the two functions, metabolic and repair, represents an improved machine model. The flexibility of enterprises that follow the machine model will be limited by a few optional packages of operations available within the existing strategies. Once environmental changes are beyond the reach of the existing strategies, enterprises that follow the rigid machine model even if it is an improved one will at best be led into infinitive and reactive chase after unattainable business sustainability.
- *Organic model applied in enterprise development represents strategically anticipatory enterprises grown from within.*
  In the context of enterprise development, the replicate function of the M-R model represents the organic growth function of enterprise management. The replicate function plays the key role to organically

integrate all three bio-functions of the M-R model into a coherent whole. The existence of the replicate function demands reciprocal and mutually complementary interactions between the operations team and the enterprise management, which cannot be imported from outside. These organic relations represent the organizational infrastructure that an enterprise needs in order to become strategically proactive instead of reactive. Such an organic pattern may only be identified with the full life cycle of PERA, particularly its Human and Organizational Architecture.

- *An enterprise that follows the organic model may sometimes behave like one that follows the machine model.*
  Since the organic model contains the machine model, when the replicate function is suppressed for a limited period of time, or when an observation is mainly focused upon the metabolic and/or repair functions of an enterprise, an organic enterprise may epistemologically look like one that follows the machine model while it still ontologically retains its full organic potentials. An enterprise that only follows the machine model will however never be strategically ready for a changing environment beyond its existing strategies because of the lack of inherent strategic change management. Mechanical or digital machines may facilitate the organic relations of an organic enterprise, but they can never substitute these human relations. Unfortunately when the thinking of machine models dominates, the desires for the productivity and efficiency seemingly promised by machine models frequently outweigh the efforts for developing organic relations within an enterprise. Even a fully organic enterprise will partially lose its organic potentials if its enterprise replicate function is weakened. The area between organic enterprises and machine enterprises is not black-and-white, but gray.

Decisions on organic development have to be verified by business reasons. This article will demonstrate how to apply the M-R model in enterprise development to evaluate not only intra-enterprise relations but also inter-enterprise relations under the A-R framework. The impacts of machine models vs. organic models on the business decisions will be discussed. Once the organic model is adopted for product or service development, an iterative and incremental approach will become a natural conclusion to develop customizable solutions. As a result, development methodology involved will become organic as well.

This article is an extension of previous work [Li, 1995, and 2000] where the authors discussed the types of connections between enterprises, which are formed during enterprise development. This article further explores the organic vs. inorganic driving forces behind the enterprise connections. Robert Rosen's Abstract Block Diagram (ABD) will be the major tool together with the mathematical understanding gained from previous discussions in this series. This article will first establish the entailment connections within and between enterprises through product models under the A-R framework. It will also further discuss the consequences of the entailment impacts. The discussions will reference the authors' industry experiences wherever they are appropriate. Discussions and conclusions of this article are applicable to virtual enterprises as well.

2. Inter-enterprise Entailment Relations With Product Model

[Li, 1995, and 2000] demonstrated that customer products, including service products, can be treated as a special type of enterprise entity with their own life cycles as defined by PERA. They generate connections between the life cycles of the product providers and the life cycles of the product customers, and then a chain of the enterprises involved is formed. Now equipped with the A-R causal framework, relationships entailed through these products and services can be further studied from a new angle, since the chain of enterprises is causal by nature.

The customers or customer entities generally represent the ultimate purpose of the development programs of the products involved, and therefore should be somehow related to the final cause of the product development life cycles. Table 1 presents a list of such examples focused on the identification of the final cause of products in question. Two related but different views of the final cause are shown side by side. When the entailment pattern of a single product life cycle is the focus of study, the simple view of the final cause listed in Table 1 may be sufficient as a generic concept that represents the purpose of the product life cycle as shown in the first article of this series [Li, 2005a]. However, a more detailed study of the entailment chain of the enterprises will need more information to identify the specific entailment roles that the products may play for the customer entities as listed under the more detailed view in Table 1. Then the types of relational models that the customer entities may follow can be further studied, where the original customer product consequently becomes an entailment component of the customer model.

Table 1. Final cause of product or service that is associated with customer's business model

Product or Service (*B*)	**Product or Service Provider**	**Final Cause (Φ) of the Product or Service (*B*) Related to Customer**	
		A Simple View	**A More Detailed Entailment View***
Production /Service Facilities	Engineering & Construction Company	Customers Business	Assets or Asset Management as Efficient Cause of Customer Business
Car	Car Manufacturer	Customer Transportation	Efficient Cause of Customers' Life
Paint	Paint Manufacturer	Protection and Decoration of Subject Surface	Material Cause of Customer Business, say, Auto Manufacturing
CRM Software	CRM Software Company	Business Needs for Automated Customer Information Management	Formal Cause and Efficient Cause of Some Business Purpose, Say Field Service, Which Needs to Build and Maintain the Desired Customer Relationships
Home Design Software	Home Design Software Company	Customer Needs for Computer Aided Tools for Home Design	Formal Cause and Efficient Cause of Customers' Home Improvement
Business Plan	Business Consultants	Business Customers	Formal Cause of the Business Product or Service Program Offered by the Customer
School Class	Class Instructor	School Students	Formal Cause and/or Efficient Cause of the Students' Life to Fulfill Some Personal or Professional Purpose

∗ Note that the more detailed entailment view demonstrates that the final cause (Φ) of the product (*B*) in question may not be the same as the final cause of the customer's business, which will be fulfilled by whatever products or services generated by this customer. The more detailed entailment view reveals the underpinning reasons why and how the missions of a product or service should be connected with the missions, business or personal, of the customer. Since further discussions on this topic are not the focus of this article, interested readers may be referred to [Levitt, 2004] and its references to see how some business scholars view the roles played by products or services, which are related to business strategies and business models, not only where the products or services are generated but also where they are utilized.

The items in Table 1 represent some of the important properties of the relational objects under the A-R causal framework: relativeness, recursiveness, and multiplicity. The role of a relational object on the causal chain is always relative to the different context of its position, and therefore, the object usually plays multiple roles on the same enterprise chain. The role of a product or service (*B*) for a customer, for example, must be somehow linked with the particular mission of the customer, which is usually represented by an associated product or service generated by this customer. In this context within the customer entity, the same *B* will most probably be considered neither as a product of this customer entity, nor its final cause.

As listed in Table 1, the paint is a product that is produced by a paint manufacturer. However, the product paint is a material cause of the car to protect and decorate a car, which is then produced by a car manufacturer. Note that equipment is considered an efficient cause in terms of Aristotelian causalities, which represents the material and physical extension of manpower or brainpower [Li, 2005b]. Therefore, a car as a transportation tool is an efficient cause for the customer who wants to achieve whatever purpose or mission that customer needs to carry out with this transportation equipment.

Similarly, production/service facilities, as products or services of an engineering and construction company, will be an efficient cause of a customer business for whatever missions that business is set off. In enterprise development, in consideration of the life cycle of the customer business, the production or service facilities actually represent the needs for asset management throughout the life cycles of the customer business in question. The product or service provided by this customer business can be anything from office rentals, petroleum products to fast-food services.

Business or personal software represents another case of possible roles played by products. When the software automates the otherwise manual procedures of data analysis and information processing, such as sales forecasting and sales reports with CRM software, it can be considered as a formal cause and efficient cause of sales management. Similarly, the software assisted calculation and visualization of home design can be another example of formal cause and efficient cause of a home improvement project.

The service products such as a business plan or a school class listed in Table 1 are somewhat similar with software. Unless they are connected with the business, professional, or personal missions of the customer involved, their impacts as a formal cause or even an efficient cause for the instructed will most probably be underestimated. Particularly, if the business consultants or the class instructors fail to help their audiences establish the causal links between their services and their customers' missions, the quality of these services will become questionable.

No matter how the entailment impacts are appreciated, it is ultimately a business and strategic decision of management to draw the line where the final cause of a product should be extended in the entailment context of customers' business. Once the causal links through products are established, the same A-R causal framework as shown in Table 2 can be recursively applied to further model each of the enterprise entities, including the product entities, throughout the chain of enterprises. The only needed modification of the taxonomy is the additional concept of products or services as shown in Table 2. Figure 1 further demonstrates a modeling framework of product or service development.

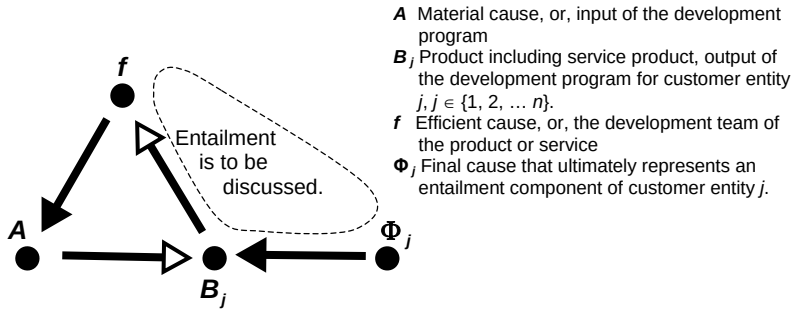


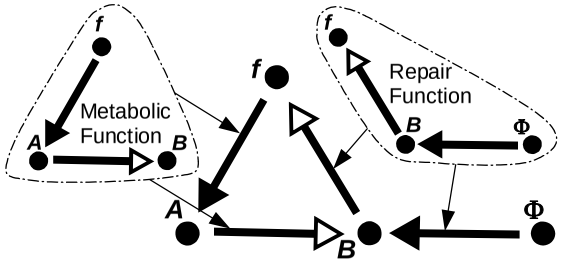
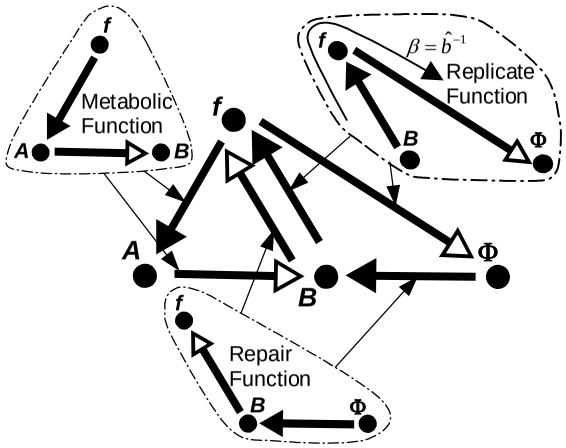
Figure 1. Relational framework of enterprise modeling for a product entity

Figure 1 stresses the possible impacts of a customer entity by adding a customer index, subscribe *j*, to the product *B* and the final cause Φ to show explicitly that it is customer entity *j* that needs the product for certain reasons. For the business of the customer entity *j*, the product B_j may play different roles in the causal terms, such as efficient cause, formal cause or material cause as listed in Table 1. Since further discussions on the reciprocal interactions between enterprises as well as environmental impacts will be needed to decide whether the product development needs to be organic, the possible entailment component, i.e. the replicate function, within the dashed area in Figure 1 has to be decided in the next session.

However, by showing the impact of customer entity *j*, the final cause Φ_j of the product development in Figure 1 actually stands for a proxy of interested customers *j*, $j = 1, 2, \ldots n$. As shown in Figure 2, the causal input from the customers to the development team, i.e., the product requirements and the orders, etc. now becomes the result of the repair function in the relational model. For the same reason, since the replicate function, if existed, would also represent relations between the development team and the customer, the area surrounded by the ellipse in Figure 2 actually represents the interactions between enterprises, i.e. the product provider and the product customers in question.

Please note that under the generic model of enterprise development as shown in Table 2, it may look like that enterprise management has full control over the financial consequences of enterprise operations, i.e., the results of the metabolic function. However, as shown in Figure 2 once the customer impact is introduced through the model of product development into the discussions, it turns out that customers of the product should also have certain influence over the enterprise development involved. The next session will establish the relationship between customer entity, production entity and its hosting entity. And then intra- and inter-enterprise relations as well as their market impacts can be further explored.

Table 2. Recursive A-R causal framework for enterprise development including products and services as enterprise entities

	Model Presentation		**Causal Taxonomy For Enterprise Development**
	Categorical Diagram	**Abstract Block Diagram**	
Improved Machine Model (Rosen's Simplest M-R System)	$A \xrightarrow{f} B \xrightarrow{\Phi} H(A,B)$ where, f – Metabolic map, $f \in H(A,B)$; Φ – Repair map, $\Phi \in H(B, H(A,B))$ $H(A,B)$ – A Hom set, which includes a collection of all morphisms between Domain *X* and Codomain *Y*.		***A*** *Material cause*, or, input of the development program ***B*** *Product* including service product, or output of the development program ***f*** *Efficient cause*, or, the development team of the program **Φ** *Final cause*, enterprise management, or customer missions of a product program Note that when the discussions move from subject enterprise being developed to customer products or services, the only necessary modification of the taxonomy is to explicitly add the customer association with the final cause because the products or services have to be generated for the customer or customer entities, instead of enterprise management [Li, 1995, 2000, and 2005b].
Organic Model (Rosen's M-R Model)	$A \xrightarrow{f} B \xrightarrow{\Phi} H(A,B) \xrightarrow{\beta} H(B, H(A, B))$ where, f – Metabolic map, $f \in H(A,B)$; Φ – Repair map, $\Phi \in H(B, H(A,B))$; β – Replicate map, $\beta \in H(H(A, B), H(B, H(A, B)))$.		

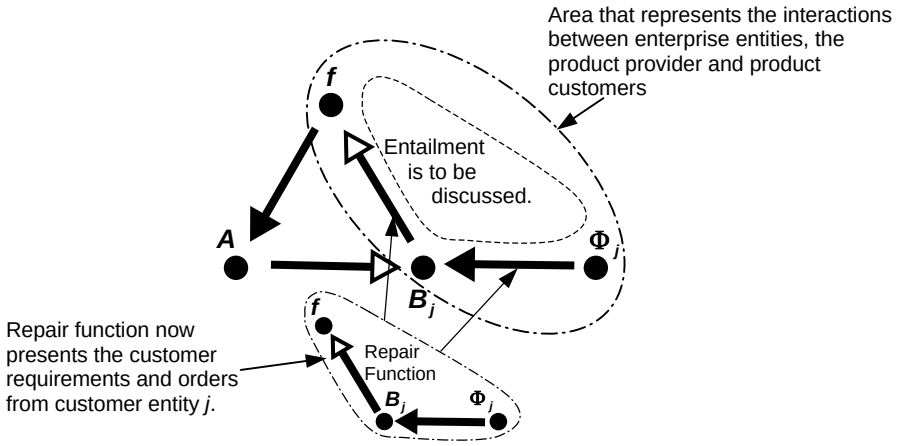


Figure 2. Inter-enterprise relations between product provider and product customers in the relational model of product development

3. Models of Intra- and Inter-Enterprise Relations

3.1 Presentation of Intra- and Inter-Enterprise Relations with Development Models

Following the graphic presentation in Table 2, let E_0 represent an enterprise development program that develops a subject enterprise B_0 as shown in Figure 3. The options for the entailment relations in the area surrounded by the dashed line in Figure 3 will be discussed in the following paragraphs. Whether E_0 is organic is therefore to be decided. One of the missions of E_0 is to further launch product development programs with B_0, such as the one shown in Figure 4.

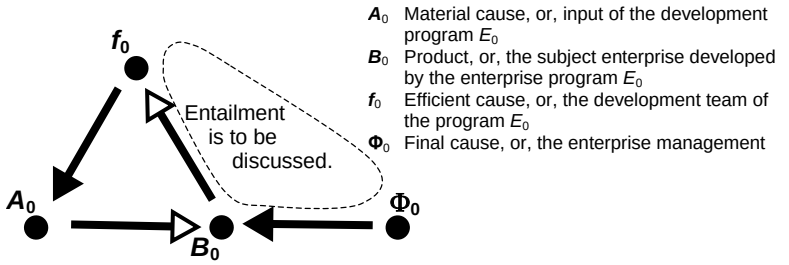


Figure 3. Relational model of enterprise program E_0

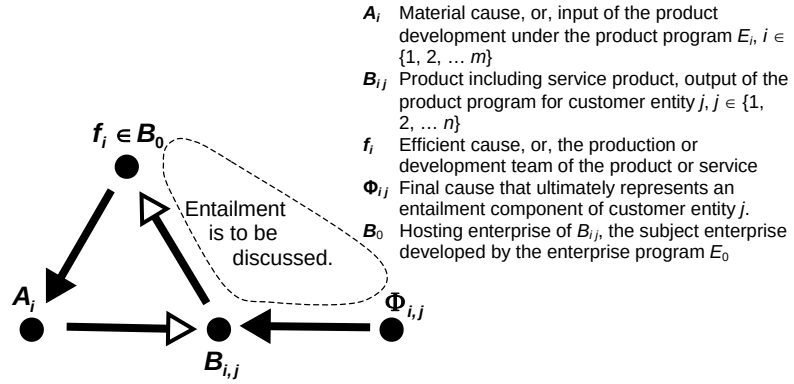


Figure 4. Relational model of product program E_i under its parent program E_0 in Figure 3

Assume that $B_{i,j}$ represents a product, where the subscribe i is a provider index, which indicates that a child program E_i, $i \in \{1, 2, \ldots m\}$ of the program E_0 in Figure 3 will provide the product $B_{i,j}$ through a

subunit of B_0, which is in turn represented by the production team $f_i \in B_0$; and the subscribe j is a customer index that indicates that another enterprise $j \in \{1, 2, \ldots n\}$ needs the product $B_{i,j}$. Since the enterprise j generates a purpose to set up the child program E_i to produce $B_{i,j}$, the final cause of the product $B_{i,j}$ is represented by $\Phi_{i,j}$. And then the product development program in Figure 1 can be further modified as shown in Figure 4.

Because f_i is a subunit of the hosting enterprise B_0, $f_i \in B_0$, Figures 3 and 4 can be combined in Figure 5. Since the relational model of the child program E_i includes the interactions between the customer entity j and the product development team f_i, as shown in Figure 2, Figure 5 illustrates how both intra- and inter-enterprise causal relations may be typically modeled.

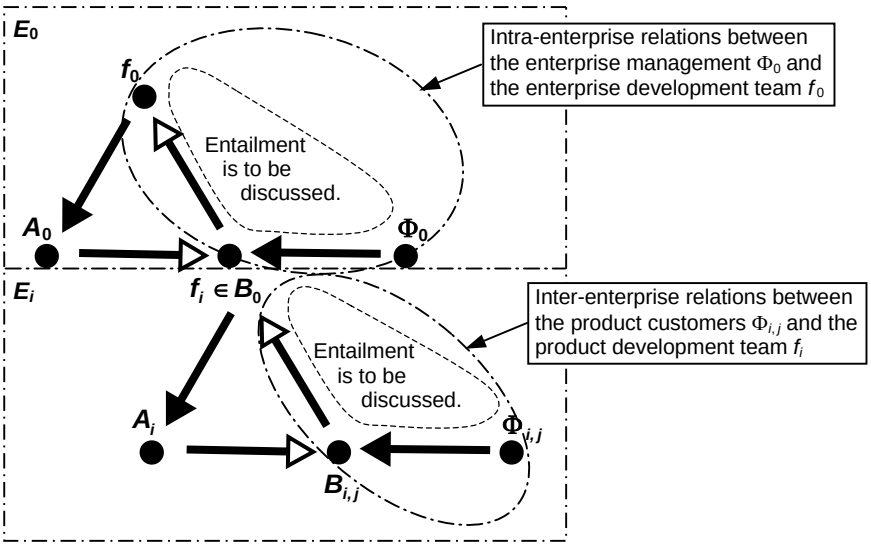


Figure 5. Relational model of enterprise program E_0 and its relationships with customer $\Phi_{i,j}$ under a child program E_i

Please note that an assumption of the intra-enterprise relations here is that the enterprise management Φ_0 is organizationally within the same enterprise as the development team f_0, though a machine model does not necessarily require this condition. Such organizational relations do not require a physical presence either. Therefore, the discussions of this article are applicable to virtual enterprises.

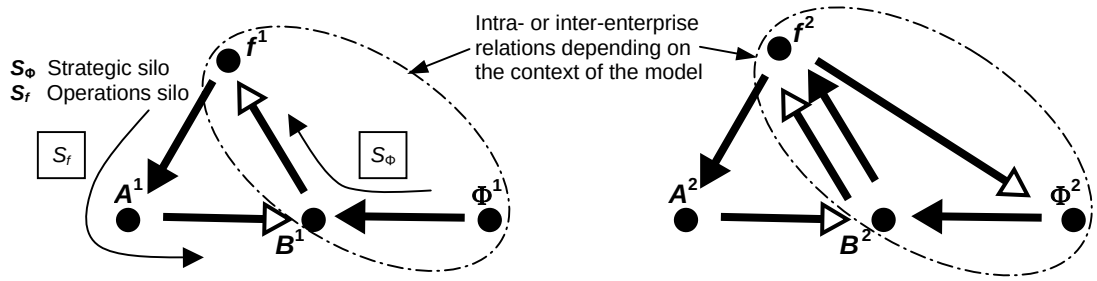
Similar consideration is applicable to possible inter-enterprise relations as well. The relational discussions are focused on the causal relations between enterprises, i.e. the actual entailment relations between existence, actions and consequences of enterprise development. Although terms like hosting unit and subunit, or parent and child, are frequently mentioned, they do not necessarily prescribe one specific financial or organizational arrangement between the two programs, E_0 and E_i, as long as these specific physical or financial forms do not alter the causal relations. For example, the two programs may represent a program and its project within a company, a pair of parent and child companies, or even strategic partners between two different companies.

Given the conditions above, Φ_0 and $\Phi_{i,j}$ represent enterprise identities of related development programs. In terms of Purdue Methodology, the identity of E_0 represented by the management Φ_0 is typically presented in the Mission, Vision and Values, or MVV of the program. The identity of the child program E_i should be compatible with the MVV of the parent program, and at the same time, $\Phi_{i,j}$ must also entertain the MVV of the customer entity j. Although the MVV of the customer entity j is not explicitly presented in Figure 5, examples in Table 1 have shown that MVV of E_i must also be related with the MVV of the customer entity j through certain entailment roles played by the product $B_{i,j}$ for j. The following discussions will focus on the case where the enterprise program E_0 and the customer entity j have

to keep their own MVVs separate. The void in entailment patterns in Figure 5 will be specified as needed, as the discussions in the next session clarify the internal and external conditions in enterprise development.

Let numeric superscripts be the organic index for the relational models of enterprise development: 1 stands for the machine model, and 2 for the organic model. The two models in Table 2 can then be presented in Figure 6 with the added numeric superscripts. The organizational silos, operations silo S_f and strategies silo S_Φ, with the machine model are formed due to the rigid hierarchical entailment embedded within this model as discussed in [Li, 2005b].

Once the nature of the relational model is clarified as organic or machine as shown in Figure 6, whether the relations between the efficient cause *f* and the final cause Φ represent intra- or inter-enterprise relations will be context dependent. As noted in Figure 5, the relations will be intra-enterprise if the model represents the development life cycles of a subject enterprise. The relations will be inter-enterprise if the model represents the development life cycles of a product.



(a) 1 stands for machine model of development program E^1

(b) 2 stands for organic model of development program E^2

Figure 6. Numeric superscripts as organic index for development models

Material, energy, and information flows in the physical world are part of the interactions between and within enterprise entities. Although they can be expressed in the relational terms as well, they are the results of the causal relations, and particularly in the context of organic enterprise development, they should support, but may never replace the replicate function that represents active, reciprocal, and complementary interactions at the strategic level either between the development team and the enterprise management in case of the organic enterprise development, or between the development team and the product customers in case of organic product development. It is the causal relations of the replicate function in the relational world as shown in (b) of Figure 6 that hold the key to the organic development strategy in enterprise development in general so that the development teams, the management, and the customers may all become active and organic agents of the development in question.

3.2 Entailment Patterns of Developmental Programs that Follow Machine Model

If every one of the clients or customers, j = 1, 2, … n, in Figure 5 is able to provide their product specifications, and if these specifications $B_{i,j}$ can be fulfilled within the production capacities of some subunit f_i of the hosting enterprise B_0, Figures 5 and 6 then can be specified as Figure 7 following the improved machine model from Table 1.

As a result, four organizational silos that are associated with the machine model in (a) of Figures 6 can be identified as $S_{\Phi,\,0}$, $S_{f,\,0}$, $S_{\Phi,\,i}$, and $S_{f,\,i}$ in Figure 7. The enterprise management Φ_0^1 in silo $S_{\Phi,\,0}$ is responsible for preparing business strategies and plans, and managing the enterprise development team f_0^1 in silo $S_{f,\,0}$. The development team f_0^1 is responsible for implementing the business strategies and plans to develop and physically maintain the enterprise operations, which results in the existence of operational B_0^1. The product team f_i^1 in silo $S_{f,\,i}$ as a subunit of B_0^1 is expected to follow the product strategies included within the business strategies of the parent program E_0^1. f_i^1 is responsible for developing and generating customer products represented by $B_{i,j}^1$. Customer entities represented by $\Phi_{i,j}^1$ in silo $S_{\Phi,\,i}$ are expected to provide their requirements for $B_{i,j}^1$.

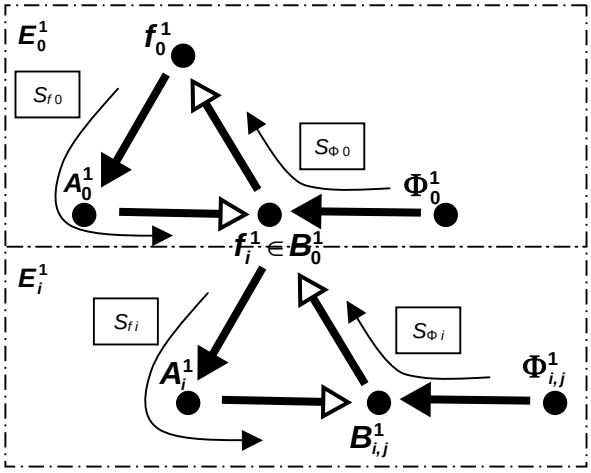


Figure 7. Developmental model of enterprise relations following machine models

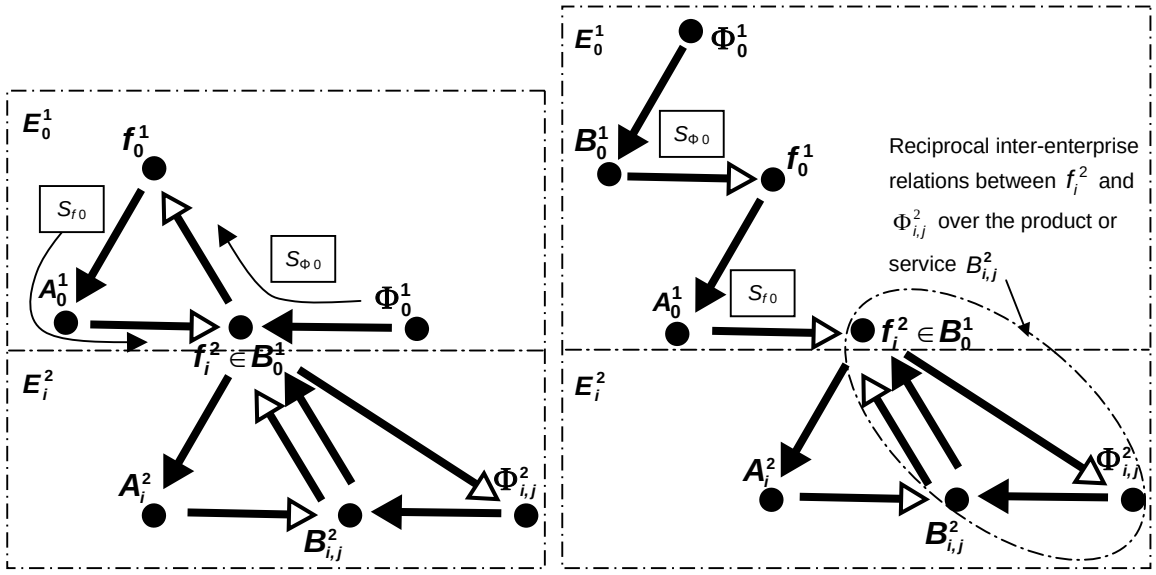
In the relational model of child program E_i^1 in Figure 7, the outside customer entities play the role of the final cause $\Phi_{i,j}^1$ of product $B_{i,j}^1$. The only causal links that connect the organizational silos between the enterprise and the customer in product development are $B_{i,j}^1$, which represents the product orders together with other customer specifications from $\Phi_{i,j}^1$, $i = 1, 2, \ldots m$, $j = 1, 2, \ldots n$. In a sense, the product team f_i^1 serves two "masters": it must follow instructions from enterprise management who decide the product strategies and production plans, and also from customers who demand certain features and functionalities of the product. As long as these instructions are compatible, clear, and complete as needed, the product team should be able to maintain its productivity given other necessary material resources.

The weak causal links between and within enterprise entities that follow the machine models in Figure 7, if the customer entities $\Phi_{i,j}^1$ are considered as a special type of enterprise entities as well, have presented at least one significant advantage: this organizational structure is simple to assemble and disassemble along the silos. This arrangement will be justified if the given business conditions remain stable and every move in the plans can be "calculated." For example, only limited variations of the product $B_{i,j}^1$ are required and the customer business $\Phi_{i,j}^1$ is mature and stable. In other words, the product life cycles, including their designs, development, production, and delivery processes, etc. are all mature and stable.

The weakness of the machine models in Figure 7 is also evident: these assumed business conditions above are by no means weak. Complete specifications of the product/service $B_{i,j}^1$, specifications of the enterprise $f_i^1 \in B_0^1$ where the product is generated, and specifications of the enterprise development program E_0^1 where the enterprise B_0^1 is built, etc., should all remain stable and unchallenged. Otherwise, such convenient organizational assembly from silos offers little from within to actively compensate for errors and correct mistakes, as revealed by the previous discussions [Li, 2005b]. The development programs in Figure 7 may soon become unstable at best once an unexpected change from environment is introduced into this machine enterprise system.

Based on previous discussions of organic enterprises, when information about the customer entity $\Phi_{i,j}$, and therefore the customer product $B_{i,j}$, is not all immediately available, one possible improvement of the developmental model in Figure 7 is to replace the child program E_i^1 with an organic one E_i^2, which is an organizationally autonomous subunit of B_0^1, $f_i^2 \in B_0^1$ as shown in Figure 8. The added replicate function together with the existing repair function in E_i^2 presents reciprocal relations between the team f_i^2 and the customer $\Phi_{i,j}^2$ as shown in Figure 8. Instead of passively receiving customer specifications from the

customer, the team f_i^2 in Figure 8 is able to interact with the customer $\Phi_{i,j}^2$ over the desired product $B_{i,j}^2$. The added relations also grant more opportunities for the customer to learn about the product from the team, say, through customer training and other educational or communicative activities. Should the customer find that it is necessary to modify their product/service specifications, the customer will have additional opportunities to communicate with the team.



(a) Organic program E_i^2 under inorganic one E_0^1 (b) Hierarchical view of E_0^1 in (a)

Figure 8. Combination between parent enterprise program following machine model and child product program following organic model

The close relations between the team f_i^2 and the customer $\Phi_{i,j}^2$ however present potential organizational conflicts between the team f_i^2, the hosting enterprise B_0^1, and the parent enterprise management Φ_0^1. As shown in (b) of Figure 8, the hierarchical nature of the siloed development program E_0^1 is change-averse, as discussed in [Li, 2005b], including both Φ_0^1 with silo $S_{\Phi,\,0}$ and f_0^1 with silo $S_{f,\,0}$. Even though the subunit $f_i^2 \in B_0^1$ is organic on its own mission for more customer information needed by the product development, its hosting entity B_0^1, a pre-plan follower with the inorganic parent program E_0^1, is not. Since f_i^2 is a direct report of B_0^1, f_i^2 can only maintain its autonomy within the scope and resource permitted by B_0^1. The performance of f_i^2 with B_0^1 is measured by the standards originally prepared by the management Φ_0^1 under the machine model. The enterprise management Φ_0^1 in silo $S_{\Phi,\,0}$ will most probably not be able to provide timely support for B_0^1 in $S_{f,\,0}$ over an operational issue that falls outside the previously defined production plans and business strategies assigned to B_0^1.

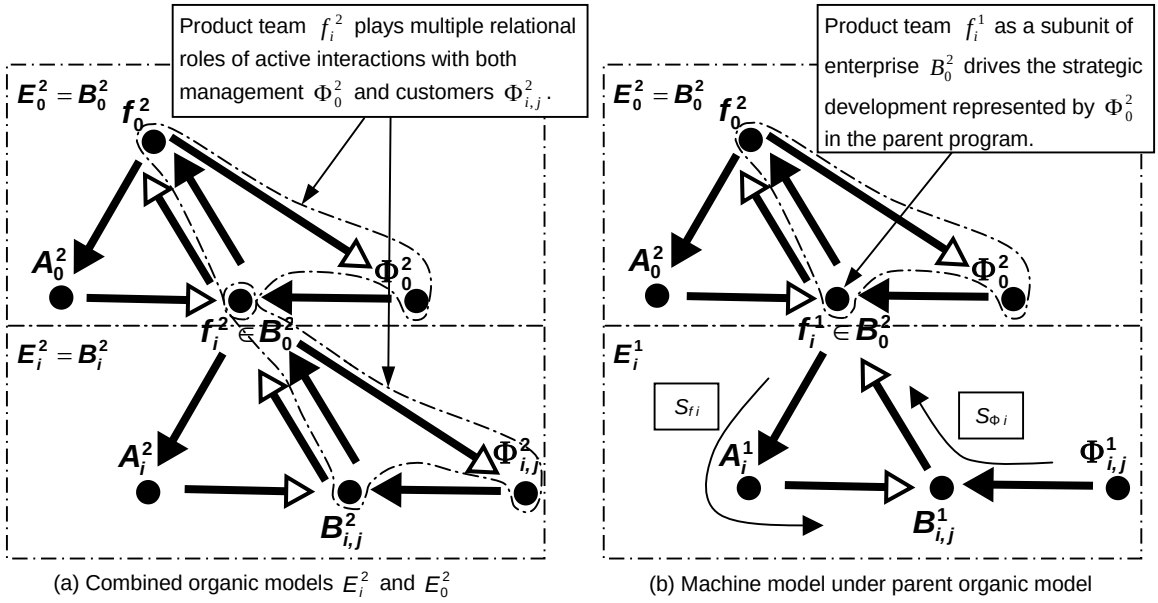
On the other hand, since the organic subunit represented by f_i^2 must actively approach the customer $\Phi_{i,j}^2$, the customer may well expect more than what f_i^2 is actually allowed to provide. In a sense, the different relational orientation between the hosting unit and the subunit will probably create self-conflicting double identities for f_i^2 in front of the product customers. If f_i^2 cannot obtain commitment for necessary resource from B_0^1 to entertain the needs of $\Phi_{i,j}^2$, or if f_i^2 underestimates the product requirements $B_{i,j}^2$, its own hosting unit will inevitably become a constraint of its performance. The quality of $B_{i,j}^2$ will then probably be compromised. Therefore, the inherent organizational incompatibility between

the parent program E_0^1 and its child program E_i^2 may become a bottleneck for the business growth of both sides, f_i^2 and B_0^1, unless the two can find a balanced position.

In order to ease the potential organizational tensions between the two sides, one option is to make E_i^2 a transitional project, for example, as an upgrade of an existing product, or a launch of new product program, and then roll the child program E_i from E_i^2 in Figure 8 back to E_i^1 in Figure 7, that is, to follow the machine model again in a product maintenance mode. Therefore, the relational model in Figure 8 is best recommended for matured production or development processes with evolving product designs.

3.3 Entailment Patterns of Developmental Programs that Follow Organic Model

Please note that the discussions in this session so far have assumed that the customer enterprise may only need either limited variations of the existing product, or may only need to tune their requirements for the product within a given scope of limited change allowed by the existing production plans, facilities, and other organizational constraints with the provider. The assumption leads to the validation of the machine model adopted for the parent program E_0^1. However, when these assumptions do not hold any more, for example, if the customer requires the product $B_{i,j}$ to catch up with their evolving business, or the management Φ_0 of the parent enterprise itself needs to explore and experiment new business ideas outside the current business plan and production facilities, then the development program E_0 of the parent enterprise will have to choose the organic model. Figure 5 should then be specified as (a) of Figurer 9.



(a) Combined organic models E_i^2 and E_0^2 (b) Machine model under parent organic model

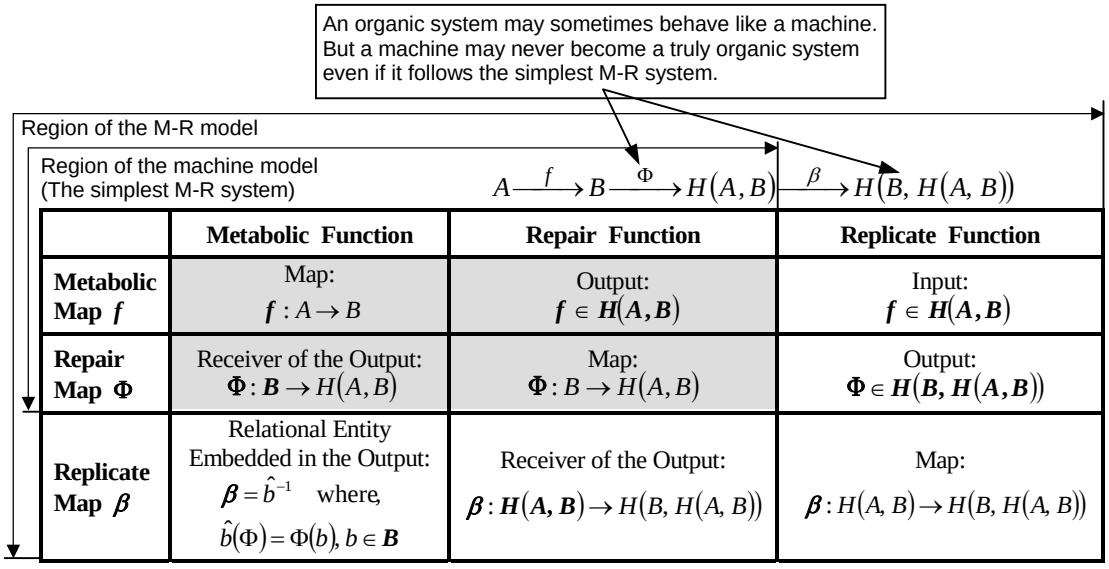
Figure 9. Product/Service development under an organic parent model

When both the parent and the child development programs, E_0^2 and E_i^2, become organic, the organizational silos inherited from the machine model are all removed, as shown in (a) of Figure 9. Following the same developmental model, both the parent and child programs have similar organic growth strategies, performance management and organizational communications, etc. Since there is no inconsistency between the two in their growth patterns, the enterprise management Φ_0^2, as long as its own identity and resource permit, will be able to proactively develop programs E_0^2 and E_i^2 to entertain the evolving needs of the customer entities.

Under the organic approach in (a) of Figure 9, the product development team f_i^2 strategically plays multiple roles with the two programs E_0^2 and E_i^2. In the organic product program E_i^2, it should

proactively approach the customer entities $\Phi_{i,j}^2$ on behalf of the product development. In the organic parent program E_0^2, as an *efficient cause* of the enterprise management Φ_0^2, it should also drive proactive interactions with the management Φ_0^2 when the team finds out that the original plans or business strategies do not fit into customer business or business environment in general.

Please note that since the simplest M-R system, or the improved machine model in Table 2, is included within the M-R model as shown by the gray area in Figure 10, which is a further explanation of Table 1 in [Li, 2005b], the M-R model allows an organic enterprise to operate sometimes like a machine. For example, when an organic enterprise undergoes regular maintenance, or even a resource crisis, it may well slow down its replicate function or even repair function as needed. However, when it looks like a machine this way, the organic system still retains the full organic potentials. The replicate function is still indispensable for being organic.



	Metabolic Function	**Repair Function**	**Replicate Function**
Metabolic Map f	Map: $f : A \to B$	Output: $f \in H(A, B)$	Input: $f \in H(A, B)$
Repair Map Φ	Receiver of the Output: $\Phi : B \to H(A, B)$	Map: $\Phi : B \to H(A, B)$	Output: $\Phi \in H(B, H(A, B))$
Replicate Map β	Relational Entity Embedded in the Output: $\beta = \hat{b}^{-1}$ where, $\hat{b}(\Phi) = \Phi(b), b \in B$	Receiver of the Output: $\beta : H(A, B) \to H(B, H(A, B))$	Map: $\beta : H(A, B) \to H(B, H(A, B))$

Figure 10. A standing alone machine cannot become an organic M-R system

Therefore, the model in (a) of Figure 9 will behave differently from the one in (a) of Figure 8 when the two systems face environmental changes. The latter will offer far fewer options than the former, as discussed in the previous session. Unfortunately, under environment pressure, a common practice is to let the parent enterprise program E_0 be reduced into a mechanical organization E_0^1 as shown in Figure 8 from an organic one E_0^2 as shown in (a) of Figure 9 by simply cutting off those activities that facilitate the replicate function, other than the other way around: growing the replicate function from within so that an enterprise in Figure 8 will become the one in (a) of Figure 9.

Since the M-R model contains the machine model as shown in Figure 10, the organic parent program E_0^2 in Figure 9 is capable of managing a development program that follows the machine model as shown by the gray area in Figure 10. That is to say, it is possible for E_0^2 to generate a child program E_i^1 that only follows the machine model, as shown in (b) of Figure 9. Since the product/service program E_i^1 is focused on a specific product, it will not become an organizational threat to the organic parent E_0^2. What has been reduced is the links with customer $\Phi_{i,j}^1$, as shown in (b) of Figure 9.

This program combination in (b) of Figure 9 may be considered as a case when a version of product design or service package has been stabilized, before the next revision is due. Such a transition back and forth between the machine model and the organic model will not be strategically viable at the global level with the parent program E_0^2, if the enterprise management Φ_0^2 ever needs to maintain the organic identity of the global development.

When the parent program E_0^2 is organic, there will be a strategic advantage to manage changes, even if the child program follows the machine model E_i^1. The hosting entity B_0^2, where the product team $f_i^1 \in B_0^2$ is a subunit as shown in (b) of Figure 9, is a coherent part of the parent program E_0^2. B_0^2 is not only able to offer more facilities than E_i^1 needs, but also allows the product team f_i^1 in the child program to drive the strategic development of the parent program as shown in (b) of Figure 9.

In other words, an organic parent program will ultimately let the product development team in its child program create and develop its own work plans, procedures, and organizational structures, even though the child program itself may follow a machine model. When the parent program becomes truly organic as expressed by $E_0^2 = B_0^2$ [Li, 2005b] as shown in Figure 9, the product team f_i^2 or f_i^1 will be also able to play the role of the development team f_0^2 in the parent program as indicated in Figure 9. In enterprise development in the real world, therefore, the roles that a product development team may play as well as their accountability, responsibility, and authority should become reliable internal indicators for the degrees of organicity of the overall enterprise development.

The organic approach is prepared for imperfect information and changing environment, which are by nature not computable. In other words, it acknowledges up front that no pre-arranged step-by-step plans or predefined strategies will ever solve all problems, though they are still necessary. There are still developmental rules, but they can be very different from those programmable procedures pursued by reductionist approaches. Particularly the siloed concepts about strategies and operations under the machine model are not applicable any more.

As a result of this organic approach, the line between strategies and operations will become blurred. Please also note that since the organic natures are built upon entailment relations between causal objects or entities involved, Rosen's organic model does not impose any material or physical conditions on the model realization. Therefore, organic enterprise development may take many different and innovative approaches.

For example, when the team from the paint manufacturer PPG not only run the paint lines for automobile manufacturer GM, but also managed its employees, the team actually explored organic relations, i.e. the replicate function, with the customers for new and probably better product strategies. Should the PPG team be able to generate new insights of their production and product delivery strategies and bring the ideas back to their reporting unit, the actions would be a typical sign of organic development.[1]

Under the Toyota production system, the concept of a customer is internally extended to any worker who works downstream in the production line. The responsibilities of the members of the production team are more than to deliver defect-free products to the customer downstream, or help this customer solve his problems. They, including this customer, are all in charge of developing better production plans and organizational structures of their own. The internal structure of the production system has every essence of organic development described in Figure 9. [Monden, 1998; Spear, 1999 and 2004]

Both Honda and Toyota have also introduced the same organic practice upstream into their supply chains. As customer enterprises along the enterprise chain, instead of waiting for organic providers, they actively grow and nurture the replicate functions between them and their vendors. Once they identified the vendors that share similar value concerns, Honda and Toyota will send their own people to stay on site with the vendors and understand vendors' business practice, and they also invite these selected vendors into their internal development cycles to learn Honda or Toyota's standards and requirements. In other words, both Honda and Toyota invest heavily and actively in long-term organic relationship building with their supply chains where their value concepts are shared and appreciated, instead of merely letting short-term economic and financial gains guide their business practice. [Liker, 2004a and 2004b; Monden, 1998; Nelson 1998; Shook, 1988]

The history of Microsoft provides another example from software industry. The nature of the software business has granted the company a huge advantage: a big pool of proxy customers out of its more than 30,000 employees for software development, "who often play the role of lead users." [Iansiti, 1998, p. 203] Through all levels of the company from development teams to Bill Gates, efforts made to improve both

[1] Interested reader may find news reports on the strategic partnership between PPG and GM from media records, say, at http://www.findarticles.com/p/articles/mi_m3165/is_n7_v32/ai_18546232. One of the authors worked as a lead business consultant for a market research project of PPG in 1997.

products and processes have been reported frequently by scholars who studied the company for years [Cusumano, 1995; and Iansiti, 1998], where the natures of organic development shown in Figure 9 can be also identified.

Should all tasks throughout life cycles of product development including identification of customer needs, initial design, development, production, and delivery, etc. be all automated by following the machine model, the proliferation of process automation would have most probably kept increasing productivities effectively, since machines usually follow the procedures batter than humans. However, when the developmental life cycles open to the enterprise chains in a changing environment, if it is important to become long-term players, the development programs will have to follow the organic model one way or another, particularly the replicate function represented by human relations within and between enterprises, unless the whole chain contains nothing but machine enterprises in a familiar environment that has little change potential. The process automation will help the organic chain but only after it becomes part of the organic development driven by organic enterprise relationships, even though the main goal may still include elimination of wastes and improvement of accuracy and efficiency.

In order to define and prepare better product, process, organizational strategies, or business solutions, the key relations within and between enterprises facing changing environment have rich perspectives, which are reciprocal, situational, exploratory, and innovative. The line between intra- and inter-enterprise relationships therefore becomes blurred. When these rich perspectives are reduced into merely commercial and technical specifications, face values of financial transactions, or digitized instructions and procedures, what the development programs have lost is the inner capabilities of proactive adaptation realized through intra- and inter-enterprise human relationships that promote and sustain self improvement and innovation right in the product and service development.

Therefore, the replicate function of organic product development and the associated organic Human and Organizational Architecture stand for a relationship infrastructure and business practice to establish, between the provider entities and the customer entities, intra- and inter-enterprise human relationships that promote and sustain self improvement and innovation throughout product development life cycles as defined by PERA.

The acknowledgement of the replicate function in enterprise development does not represent a denial of the existence of different enterprise identities, the potential competitions, and even potential conflicts between enterprises, which are part of the changing environment around an enterprise. On the contrary, this acknowledgement is more than recognition of these facts. It presents a direction of organic solutions: all parties involved should understand their entailment roles along the enterprise chain, and develop the right relations and actions with others, or find better partners. Otherwise the chain of the enterprises together with the individual enterprises that expect others on the chain to become their profitable machines will not be self-sustainable in the long run any way.

The acknowledgement of the replicate function in enterprise development does not represent a denial of the benefits of automated machines and technologies either. A large portion of enterprise operations, i.e. enterprise metabolism in terms of the organic model, including those in product development, are performed through machine equipment. The productivity and efficiency achieved by these machines have become one of the most important foundations of modern civilization. The same economic results feed organic enterprises and their chains of enterprises as well. However, any industry automation established through ignoring, instead of facilitating, the necessary organic replicate functions either within enterprises or between enterprises will suppress the organic potentials, and will eventually turn everything into machines, which do not have inner capabilities to grow and adapt from within in a changing environment. One of the unavoidable consequences is that the enterprise development will then lose the very source to self improve or innovate the machine operations.

When management of an enterprise that follows the machine model is convinced that a digitally integrated performance management system based on IT solutions will by itself make the organization and their supply chain more adaptive, the newly installed information system will probably first give the management within the existing silo a false illusion that they were better connected with their enterprise operations and their supply chains than they really are. Once it is discovered later that such an IT solution does not deliver the desired organizational adaptability, the IT solutions usually have to shoulder the full blame for the failure, while it should probably have been at least a shared responsibility between the management and the solution provider(s).

Such issues in enterprise development represent much more difficult and challenging research subjects than traditional process automation. They have been however largely ignored by mainstream researches of

technology in academia and industry. The conventional approach of process automation will unfortunately represent a dead end if blindly applied to organic development: it is impossible for all organic relations to be pre-programmed, either internally or externally. Or otherwise the enterprise can only become a machine-like entity, when it is controlled and measured as a machine.

3.4 A Market View of Enterprise Developmental Models

Discussions in this article have been so far mainly focused on the entailment roles that a product may play for either its customers or providers. In the course of interactions between customers and providers in marketplace, a customer's selection of a product from a particular provider usually represents preferences that include but are more than these product functionalities. Many of these preferences can be traced back into the business models and cultures of the customer throughout its business life cycle in question.

For example, a customer may choose straightforward orders from a provider whose product development follows the machine model E_i^1, instead of from a provider who follows the organic model E_i^2, which may consequently need certain reciprocal interactions between the customer and the provider. This decision may indicate that the customer could be insensitive to potential functional improvement based on the organic growth model of the product, and therefore the customer could view the product as commodity with functions already sufficient. Or this customer may just prefer to stay with a stabilized product design.

If however a customer has chosen a provider with an organic product development model E_i^2 instead of the machine model E_i^1, the customer is probably be sensitive to the growth model of the product, and may consider the product as a value-added entity with helpful functions. Therefore, this customer will most probably value opportunities of reciprocal interactions with the product provider.

When a customer's decision is influenced not only by the product development model E_i, but also by the enterprise development model E_0, the decision may demonstrate more of the growth orientation of the customer. If a customer has chosen products from inorganic enterprises E_0^1 over organic ones E_0^2, for example, the customer may not appreciate the potential benefits of continuous product and process improvement, and it may not consider any further improvement of the product will be critical to its own business. Therefore, this customer will mainly look for product/service based on contractual relationship, efficiency, and low costs.

If a customer has chosen organic enterprises E_0^2 over inorganic ones E_0^1, the customer will probably appreciate the benefits of continuous product improvement through design and development processes, and this customer may consider that any further product improvement will be critical to its own business development. Furthermore, this customer may be also interested in pursuing strategic partnership with the provider based on existing interactions with the provider.

Table 3 summarizes briefly the discussions from previous sessions. All possible entailment combinations between parent programs and child programs, i.e. the enterprise development program and the associated product development program, are listed. These combinations represent strategic positions of development programs, which a product/service provider, or a customer, may choose. The typical consequences of different developmental strategies are listed together with each of the program combinations. An assumption of Table 3 is that with bounded rationality [Simon, 1982], customer entities in the market generally make their procurement decisions based on limited information available about the products or services and their development strategies.

Please note that for product providers, the organizational transitions between the two columns for child programs in Table 3, E_i^1 and E_i^2, are acceptable, and as discussed previously, may well be necessary when they are justified by the business involved. However, frequent transitions between the two rows for parent programs, E_0^1 and E_0^2, particularly from E_0^2 to E_0^1, are usually not recommended due to the negative global impact on the enterprise identity.

Table 3 lists in the gray area possible considerations taken by the customers when they approach the product providers. The customer preference as discussed above is summarized as customer views. Since Table 3 groups these views according to the developmental models of providers, it is probably more important for the customers to look for balanced decisions based on the different views than just taking actions based on a one-sided view.

Table 3. Positions of product/service providers vs. their customers

		E_i: Product Development Program of E_0		**Customer Views (Row View ←) of Development Organizations**
		E_i^1 (Machine Model)	**E_i^2 (Organic Model)**	
E_0: Enterprise Development Program	**E_0^1 (Machine Model)**	$E_0^1 - E_i^1$ (See Figure 7) For provider's capability, this quadrant typically represents matured product development process and matured, or stabled product design. For repeated customers, this quadrant represents familiar business transactions, and familiar technical specifications of products. The lowest price and highest efficiency are commonly sought by customers here, since providers may have nothing else to offer to compete for customers.	$E_0^1 - E_i^2$ (See Figure 8) For provider's capability, this quadrant typically represents matured product development process and evolving product design For repeated customers, this quadrant represents familiar business transactions, and some familiar technical specifications, except those of new features and functionalities of the product. Certain customer interactions are possible.	If a customer has chosen products from inorganic enterprises E_0^1 over organic ones E_0^2, the customer may not appreciate the potential benefits of further product improvement, and it may mainly look for product /service based on contractual relationship, and low costs.
	E_0^2 (Organic Model)	$E_0^2 - E_i^1$ (See (b) of Figure 9) For provider's capability, this quadrant typically represents evolving product development process and stable product design For repeated customers, although this quadrant may represent familiar business transactions, and familiar technical specifications, the quality of both the services and the products are improved constantly. Customer values and relationships are probably respected.	$E_0^2 - E_i^2$ (See (a) of Figure 9) This quadrant typically represents evolving product development process and evolving product design For repeated customers, although this quadrant may represent familiar business transactions, the service quality, the product quality, and the product features and functionalities may be improved constantly. New business opportunities are frequently explored and experimented. Close customer interactions can be regular activities.	If a customer has chosen organic enterprises E_0^2 over inorganic ones E_0^1, the customer may appreciate the potential benefits of further product improvement in design and development processes, and may also be interested in developing strategic partnership with the provider.
Customer Views (Column View ↑) of Product /Service		If a customer has chosen straightforward customer orders under the machine model E_i^1 for product development over reciprocal interactions under the organic model E_i^2 with the provider, the customer may be insensitive to the growth model of the product, and may view the product or service as commodity with fixed functions	If a customer has chosen reciprocal interactions under E_i^2 with the provider over straightforward customer orders under E_i^1, the customer may be sensitive to the growth model of the product, and may view the product or service as a value-added entity with helpful functions	

For example, when a customer enterprise somehow considers a needed product as a piece of commodity relative to its business strategy, it may have two choices for its purchase decision with provider program E_i^1 in Table 2: vendors that follow the machine model as indicated by $E_0^1 - E_i^1$, or vendors that follow the organic model as indicated by $E_0^2 - E_i^1$. If additional improvement on the product quality or design, including associated service offers, may still carry considerable values to its business, the customer enterprise will probably select its provider based on the hybrid developmental model $E_0^2 - E_i^1$. Otherwise, the pure machine combination $E_0^1 - E_i^1$ will be a natural choice. Please note that the second choice $E_0^1 - E_i^1$ does not necessarily mean that the commodity in question will never be improved or upgraded with E_0^1. The parent program E_0^1 may still move to a developmental combination represented by $E_0^1 - E_i^2$ for the next generation of the commodity. However, the availability of functionality and quality of the product produced under E_0^1 may never be able to match up with those produced under E_0^2, since the latter will most probably push more improvements both in production process and product design than the former.

Although the customer that chooses E_0^1 to do business with may not necessarily be an enterprise that follows the machine model, an enterprise that follow the machine model will most probably pick a provider that follows the same model, because the additional features and functionality outside the current specifications of the "machine" will add no values until the "machine" receives its upgrade instructions from an outsider. As an exception, those child programs represented by E_i^2 under machine organization E_0^1 may very well become a customer of an organic enterprise E_0^2 because they will need organic facilities that are largely missing under their own parent programs E_0^1.

As a result of this market selections made by the customers in the market illustrated by Table 3, the customers that stay with the organic providers will most probably be those that themselves also operate on similar organic models. The product selection from E_0^2 does not mean a constant change in product design since the parent program E_0^2 also runs a stabilized developmental program represented by E_i^1, which leads to combination $E_0^2 - E_i^1$ in Table 3. More importantly, the enterprises and their customers that operate based on the organic model E_0^2, if they can be successful in the long run, will eventually become the market leaders in a changing environment. In a sense, such a market movement is led by the overall results of joint efforts made by provider entities and customer entities, which have adopted organic strategies of development.

The organic relationship management is what is missing from the machine model. Both inter- and intra-enterprise relationship management that follow the organic model will need both leadership and enterprise infrastructures that are different from those under machine model. Otherwise, rigid hierarchies inherited from the machine model, which is indifferent to the organic relationships, will neutralize any organic efforts made between management and operations, or between customers and providers. For example, if an employee is expected to fulfill daily pressing quota with no help to solve unexpected problems, or a provider is treated as the same as the commodity he is selling with very thin profit margin left to survive, people who play these roles will be in no position to help their powerful employers or demanding customers improve the product involved. Unfortunately, operated without replicate functions, people with machine enterprises will tend to think that the machine model should be the right one to describe their enterprises, and then easily extend this conclusion to all enterprises.

The organic relationship management and organic Human and Organizational Architecture play the pivotal roles in organic enterprise development. Once they are removed, any enterprise development, inter- or intra-, can only become machine organizations without any other choice in the causal terms of A-R framework, which is unfortunately the only tool that the authors can find to expose the underlying logic structure. With business practice under the machine model, the relational approaches will be inevitably deemed transaction costs unnecessary and unproductive. The technical independence [Li, 2004] of these discussions in this series has demonstrated that technical solutions alone may only facilitate or enhance the resulting "organism" or "machine," but will never be able to convert machines to organisms. It will be the

managerial decisions and practices that ultimately decide what will be implemented in reality. It is our duty to reveal the otherwise hidden causal relations between these decisions, practices, and their consequences, which may represent very different outcomes in different environments.

4. Experience from Enterprise Development in Industry

<Shelby: As the suppliers, manufacturing & service providers, customers, and cost/geopolitical demographics become much closer coupled and interactively involved, the Anticipatory and Dramatic change capabilities in Rosennean replicative feedback may become a dominant feature either limiting or enhancing the life cycle of an Enterprises and even larger complex Endeavors. PERA and Rosennean theory ARE THE ONLY FULL MODELS which can provide the support for addressing these needs.> [Hong: it will be great if you can find some examples for the anticipatory and change capabilities]

<Shelby: On the subject of scalability and flexibility of system architectures, SP-95 Parts1, 2, 3 (IEC/ISO 62264) on integrating the controls, manufacturing and enterprise layers are rapidly becoming an evolutionary factor. Using PERA and Rosennean approaches will be of great benefit to clients in using these approaches on new enterprises and adapting existing enterprises to them. Similarly ISO 15926 on temporal control and management of the enterprise's information life cycle.> [Hong: some examples of evolutionary system architectures would be great here too]

<Shelby: Add some cases and lessons learned from Enterprise Development in Industry - Mike Engineer who is part of our EICS department has some which will be written up for presentation at this years ISA Expo/Conference (See attached abstract). I have quite a few as well.> [Hong: Your examples of importance of PERA Human and Organizational Architecture will be appreciated very much! We may also invite Mike to coauthor this article if we need his materials. Please let me know.]

5. Summary

In a reductionist view, the selection of growth model is probably machine model *or* organic model. But in a holistic view, it is both machine model *and* organic model, with the latter containing the former. Any holistic claim without capability to be inclusive will be questionable.

Organic enterprise development is an organizational association of autonomous individual enterprises where strategies and associated executions become a coherent whole based on institutional and mutual relations, including both inter- and intra-enterprise relationships such as those between management and operations, and those between providers and customer.

The inner dynamics of organic enterprise development is able to co-evolve with its changing environment: its marketplace, social and political conditions, and global economy.

Acknowledgments

6. References

Cusumano, M. and Selby, R. (1995) Microsoft Secrets, with Preface to the paperback edition in 1998, Simon and Schuster.
Iansiti, M. (1998) Technology Integration, Harvard Business School Press, Boston, Massachusetts.
Levitt, T. (2004) "Marketing Myopia", Harvard Business Review, Vol. 82, Nos. 7 & 8, pp. 138 – 149.
Li, H. (2005b) "A Holistic Model for Enterprise Development (II)", work in progress.
Li, H. (2005a) "A Holistic Model for Enterprise Development", BioTheory, First Issue, available from http://www.rosen-enterprises.com/RobertRosen/HongLiBioTheoryPaper.
Li, H. and Williams, T. J. (2004) "A Vision of Enterprise Integration Considerations", 2004 International Conference on Enterprise Integration and Modelling Technologies (ICEIMT '04), University of Toronto, Canada, October 9-11, 2004.
Li, H. and Williams, T. J. (2000) "The Interconnected Chain of Enterprises as Presented by the Purdue Enterprise Reference Architecture", Computers in Industry, Vol. 42, Nos. 2-3, pp. 265–274.
Li, H. (1995) A Formalization and Extension of the Purdue Enterprise Reference Architecture and the Purdue Methodology, Ph.D. Thesis, Purdue University, West Lafayette, Indiana, December 1994. Also published as Li, H. and Williams, T. J., A Formalization and Extension of the Purdue Enterprise Reference Architecture and the Purdue Methodology, Technical Report 158, available at http://iies.www.ecn.purdue.edu/IIES/PLAIC/PERA/Publications, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, IN 47907, April 1995.
Liker, J. K. and Choi, T. Y. (2004b) "Building Deep Supplier Relationships", Harvard Business Review, Vol. 82, No. 12, pp. 104 - 113.
Liker, J. K. (2004a) The Toyota Way, 14 Management Principles from the World's Greatest Manufacturer, McGraw-Hill, New York.
Monden, Y. (1998) Toyota Production System, An Integrated Approach to Just-In-Time, 3rd Edition, Engineering & Management Press, Norcross, Georgia.
Nelson, D., Mayo, R., and Moody, P. E. (1998) Powered by Honda, Developing Excellence in the Global Enterprise, John Wiley & Sons, Inc., New York.
Rosen, R. (2000) Essays on Life Itself, Columbia University Press, New York.
Rosen, R. (1991) Life Itself, Columbia University Press, New York.
Shook, R. L. (1988) Honda, An American Success Story, Prentice Hall Press, New York.
Simon, H. (1982) Models of bounded rationality, MIT Press, Cambridge, Massachusetts.
Spear, S. (2004) "Learning to Lead at Toyota", Harvard Business Review, Vol. 82, No. 5, pp. 78 – 86.
Spear, S. and Bowen, H. (1999) "Decoding the DNA of the Toyota Production System", Harvard Business Review, Vol. 77, No. 5, pp. 96 – 106.
Williams, T. J. (1998) "Characterization of the Place of the Human in Enterprise Integration" In Handbook of Life Cycle Engineering, Concepts, models, and technologies, Molina, A., Kusiaka A., and Sanchez, J. Eds. Kluwer Academic Publishers, Dordrecht, Boston, London, 1998.
Williams, T. J., Rathwell, G. A., and Li, H., Eds. (1996) A Handbook on Master Planning Implementation for Enterprise Integration Programs. Based on the Purdue Enterprise Reference Architecture and the Purdue Methodology, available at http://iies.www.ecn.purdue.edu/IIES/PLAIC/PERA/Publications, Purdue University, West Lafayette, Indiana, US.
Williams, T. J. (1992) The Purdue Enterprise Reference Architecture, Instrument Society of America.
Williams, T. J. (Ed.) (1989) Reference Model for Computer Integrated Manufacturing, Instrument Society of America, Research Triangle Park, NC.