DEFINING FLEXIBILITY FOR INFORMATION SYSTEMS

by

LIEVEN L. VANMARCKE

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING IN PARTIAL
FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF

BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1988

Signature of Author____

Department of Mechanical Engineering
May 6, 1988

Certified by____

Michael Packer
Lecturer in Mechanical Engineering
Thesis Supervisor

Accepted by____

Peter Griffith
Chairman, Department Committee

DEFINING FLEXIBILITY

FOR INFORMATION SYSTEMS

by

LIEVEN L. VANMARCKE

Submitted to the Department of Mechanical Engineering
on May 6, 1988 in partial fulfillment of the
requirements for the Degree of Bachelor of Science in
Mechanical Engineering

ABSTRACT

Due to a misunderstanding of flexibility requirements in the analysis and design stages of system development, many managers find that newly implemented information systems evolve into a constraining facet of their organization.

In attempting to formulate a logical, cohesive definition for the concept of flexibility as it relates to information systems, research included conducting interviews with academia, industry, and information systems consulting firms as well as a literature search. Very little has been written on the subject, and the interviews also showed a technological confidence in new technology as a primary means to solve inflexibility problems.

After outlining the objectives of and existing definitions for flexibility and proposed remedies for inflexibility, the thesis defines flexibility for information systems in terms of goals, scope, means, and action. A formal categorization of important issues is supported with examples.

By recognizing and considering the business assumptions which are likely to change, system designers can better balance trade-offs by limiting embedded assumptions in unknowable areas and revisiting business strategies before final system design. By matching the number and type of functionality requirements built into the architecture to the level of certainty in the requirements, and building systems basic capabilities or "hooks" to anticipate certain unknown requirements, the result is increased flexibility in information systems.

Thesis Supervisor: Michael Packer

Title: Lecturer in Mechanical Engineering

Table of Contents

Objectives

Since the advent of the computer, the use of information systems to "gain competitive advantage" has become a popular goal. There is little disagreement about the strategic importance of these systems. Many managers are finding, however, that newly implemented information systems often evolve into a constraining facet of their organization. To adapt to changes in the business environment, a company may spend millions of dollars more than competitors with more flexible systems. Another company may lose its competitive edge in a market due to the greater lead time associated with adapting inflexible systems to changing business needs.

Much of this inherent inflexibility stems from a misunderstanding of the flexibility requirements for information systems and the organization. In particular, this lack of understanding occurs during the analysis and design stages before implementation. Ideally, information systems must be able to match business needs from the outset, and, more importantly, be able to grow and adapt as the organization changes.

This thesis attempts to develop a theoretical definition of flexibility as it relates to information systems for business applications. After looking at existing definitions of flexibility and proposed remedies for inflexibility, the thesis defines flexibility for information systems in terms of goals of flexibility and scope of the flexibility definition. It then discusses means to achieve this flexibility and actions required to attain flexible systems.

Existing Definitions of Flexibility and Proposed Remedies for Inflexibility

In trying to define any term, it is often useful to consider both the explicit definitions of the term itself, and implicitly its use in practice. These existing definitions provide a foundation for further analysis.

Let us first examine the explicit definitions of the word "flexible." Webster defines the word "flexible" as "responsive or adaptable to change." The Oxford American Dictionary defines "flexible" as "adaptable; able to be changed to suit the circumstances." Its definition for the word "adaptable" is "to make or become useful for a new use or situation." It is clear from these explicit definitions that the core of the definition lies in the connection of "flexibility" with "change."

An implicit definition relating flexibility to information systems can be inferred from the above. Using the literal definitions, information systems are flexible if they can adapt to change. It is this "change" that must be examined and understood to define formally the term flexibility. For our purposes, what does change mean in terms of business needs and adaptability requirements?

John Kogan, a Partner with Arthur Andersen, believes that "flexibility supports growth given changes in volume, functionality, new needs, product lines, and business needs."(1) Tom Davenport, Director of Research at Index Group, gives a more general description of change: "Flexibility lets you do what you want to do when you want to do it."(2)

It is important to consider some of the proposed remedies and barriers for the present inflexibility in many information systems to understand fully the reasons for examining changes as they relate to flexibility. Many claim that new and future technological tools - such as relational database management systems and fourth-generation languages - will make systems so flexible that any changes can be accommodated without difficulty. Some believe that these new tools will make the concept of adaptability a non-issue. Technical innovations and new data management tools do seem to promise a degree of adaptability previously non-existent, but do not alone provide the flexibility we need to meet changing business needs. There is one fundamental reason why these new tools are not a panacea in developing a flexible system.

Ultimately, most inflexibility problems have their roots in the design stage of system development. In every system, assumptions are made during this design stage with respect to organizational structure, business strategies, marketing issues, etc. These assumptions are essentially "hardwired" into the system architecture. As a result, the directions of the firm and the future needs in relation to adaptiveness become embedded in the structure of the systems. Often decided on by system managers alone, these "hardwired" assumptions carry implicit references to the business managers' general competitive and management approach. For example, a system manager under time constraints may implicitly decide a firm's future product offerings through embedded assumptions.

The fanciest relational databases and fourth-generation languages will not overcome the constraints that result from a poorly designed system architecture. From as early as the analysis stage in development, flaws in the design process can misrepresent the firm's true business needs. Robert Anthony proposes that these business needs, or Management Functions, fall into three categories: generating strategic plans, performing tactical plans, and controlling operations.(3) Each category represents a set of functionality requirements with specific levels of certainty associated with each requirement. To meet these needs in the present and the future, the systems developed must match the requirements that accompany them, with consideration for the certainty associated with specific requirements.

Toward a Definition of Flexibility

A coherent discussion of flexibility encompasses four issues: the goals of achieving flexibility, the scope of the changes required for flexibility, the means to obtain flexibility, and the necessary first actions.

GOALS

What are the goals of information systems flexibility? Greater flexibility generally translates into improved performance for a company. This change in performance can be measured by reductions in time to perform business activities, reductions in cost, and reductions in management aggravation. In addition, flexibility can be measured by the degree to which systems built for one set of business activities can effectively support other new activities. It is useful to clarify these concepts with examples.

To demonstrate the use of flexibility in reducing time to perform business activities, let us look at a firm manufacturing motorcycles. New emission standards force the company to change the design of the muffler assembly. Instead of starting the new design from scratch, the flexible CAD system allows alterations to be made much more quickly without the costs associated with a brand new design.

Assuming time is inherently connected with money in today's business world, this time reduction example can also be viewed as an example which demonstrates cost reduction. The changes to be made, as a result of the flexible system, cost less to implement than drafting an entirely new drawing.

Although overlapping the previous categories slightly, the goal of reducing management aggravation can easily be described by looking at the case of a company manager trying to master a graphics program on his Apple Macintosh personal computer. Already familiar with a word processing program using the same front-end pull-down menu, he adopts the new package easily because fewer new features need to be learned. Where cost or time to learn may not necessarily be issues for any paticular user, in any case, the less hassle there is in learning a few new features as compared to learning an entirely different program (with those same features), the better.

The last category, measuring the degree to which systems built for one set of business activities can effectively support other new activities, can be illustrated using the the previous motorcycle company example. If the company decides to produce scooters (with similar but fewer parts) instead of motorcycles, its remaining systems may be too large and complex to support efficiently the scooter producing function. In this case, the required systems must be flexible enough to meet the new, smaller production needs efficiently. Basically, this need for adaptability means the systems must have the capability of bypassing the resulting unused system modules.

In summary, a system is flexible if the cost to implement a given change is low compared to the value of that change in terms of reduced business cost, time, aggravation, and support of new activities.

SCOPE

Having evaluated the goals of flexibility in the context of information systems, it is now useful to consider the business needs or "Management Functions" which must be incorporated into company systems to achieve flexibility. The scope of these needs must be categorized in order to focus our definition. These catagories essentially overlap the previous assertions about scope (Anthony's definition) with different labels given for each. Systems flexibility can be viewed as falling under three categories: Strategic, Market-side, and Internal.

Strategic flexibility is basically the ease with which a firm can change its business strategy. From a systems point of view, because strategies are often uncertain or unclear, this type of change is one of the most traumatic to accommodate in design. Strategic flexibility should accommodate a change in strategy, such as from a concentration on profitability to a focus on market share leadership, and encompass a wide variety of the potential business changes that accompany a strategy change. Another example might be a strategic diversification change from selling software to a concentration on catalogue selling where the products being sold include office supplies as well as software; the subscription service offering might require extensive rebuilding of systems. These types of strategic change can require huge systems changes, from the most basic architectures to the most detailed codes for product definition. The examples are endless, the resulting need for flexibility is immense, and strategic scope is difficult to limit unless the business is very clear about its strategy and believes it will not change.

Market-side flexibility refers to the business decisions relating to marketing issues. The ability of a firm to change its pricing policies, its product lines (add products, drop products, diversify), and its customer base depends on the market-side flexibility of its systems. For example, when Apple Computer decided to alter its primary target market (a strategic change in itself) to include businesses as well as individuals, their in-house systems had to accommodate and adapt to the new market demands. The changes in marketing (advertising, pricing, promotion, etc.) needed in-house systems support. In addition, if a chemical company decides to consolidate its different divisions and track only high volume sales in order to alter marketing approaches to these specific customers, its accounting systems have to match to be of use.

Internal management flexibility relates to the internal structure of the company's organization and its ability to assess its own performance. If a company wants to change from a geographic management structure to a brand management structure, this change constitutes a need for internal management structure flexibility. If a company wants to change its evaluations of salespersons from number of actual sales to number of new customers or net profits, it would need internal flexibility for its performance assessment procedures.

MEANS

How do we achieve this flexibility in information systems? Let us assume that the overall flexibility of a system is some product of its technical flexibility (i.e. the flexibility of the systems themselves and its "architecture") and the flexibility of the organization itself (the firm's inherent ability to adapt to both internal and external changes).

(Technical Flexibility) X (Organization Flexibility) = Overall Flexibility

Relational databases, fourth generation languages, and modular systems are all means of attaining greater technical flexibility. Hypertext, a software package for the Macintosh computer, is an example of a technically flexible word processing system. Because the text is defined and treated as an object, sentences, paragraphs, and entire sections can be moved around freely within the body of the text. The most important part in designing any modular system, however, is to correctly define the boundaries of the modules (see figure 1). If these boundaries are initially incorrect (inconsistent with business assumptions) the "modularity equals flexibility" concept does not hold. For example, the director of marketing at a quickly growing bank wishes to look at all accounts with balances over a specific amount. The "modular" systems, however, were only designed to call up accounts based on name or account number. The resulting inflexibility demonstrates the need for carefully defined functionality requirements from the the outset.

Basically, the more functionality requirements properly built into the system, the more flexible. The greatest trade-off or cost of building in flexibility lies in the excess data needed to support multiple views of the organization and the business. The data storage capacity, the necessary algorithms, and the people (entry) time all contribute to the final limitations on any system. For example, a firm wishing to retain flexibility to reorganize around products, customers, geography, and organizational unit would have to track every expense by product, customers, geography and organizational unit - an impossibly costly endeavor.

The organizational flexibility category relates very much to the structure of the organization (management processes - paperwork, bureaucratic regulations, heirarchical structure) and its people (number, qualifications, culture). If the organization is inherently inflexible, even great technical flexibility will not provide great overall flexibility.

Using the above simple model to organize our definition, let us assume that the technical flexibility issues (which may include both tools and embedded assumptions) and the organization flexibility issues are both included in the concept of "overall flexibility." We must now decide on the actions needed to build a flexible system successfully.

ACTIONS

Having addressed the scope of, goals for, and means to build flexibility in information systems, it now becomes important to examine the specific actions required to make a system flexible. The building of a flexible information system must take into consideration business assumptions to understand fully and select specific flexibility requirements and must maintain a "feedback loop" (sending information back to the requirements stage before final implementation). The process can be roughly diagrammed as in figure 2.

Let us make some more formal characterizations of the business assumptions on which we are concentrating. These assumptions can be split into two general categories: managerial business assumptions and assumptions related to the environment.

Managerial business assumptions include decisions regarding competition: How do we compete - Low price? Better service? They may include money-making assumptions: An insurance company makes money by spreading its risk over many clients and minimizing its overall risk. Even the most basic business assumption must be evaluated: What business are we really in?

Environmental assumptions also play a great role in the early stage of building flexible systems. Considerations must be given for the exchange rate, the inflation rate, and the rate of technological innovation. These are environmental assumptions which, before becoming "hardwired" into the systems, must be considered and evaluated in a manner consistent with the management goals of the firm. For example, if an international securities firm assumes that the dollar will never pass a certain rate compared to the yen, it is constraining its flexibility requirements.

After identifying all the assumptions associated with the managerial and external environments, an in-depth understanding of the areas of required flexibility is needed. Each business requirement can be characterized as a known, unknown but knowable today, unknowable but bounded, or unpredictable (unknowable and unbounded) requirement.

Known flexibility requirements include any clearly defined requirements arising from factors such as present organizational structures, short-term strategies, goals,etc. Examples may include leaving room in database files for a new product line or perhaps an option for a national roll-out. In general, these known flexibility requirements require little if any system flexibility; simply build the stable requirements into the system.

Unknown but knowable requirements include issues presently unclear or simply unaddressed. These issues can become clear (become known requirements) in the near future if the necessary effort is expended.

Examples may encompass such issues as future investment opportunities, new businesses to enter into, or perhaps future mergers. If a consumer products company is unsure whether to incorporate a particular new product into its systems architecture, it can wait for test market data or simply build in the flexibility to accommodate it. These unknown requirements usually call for deferral of a decision about what to build until more is known or the building of these flexibility requirements into the system if the cost of discovery is higher than the cost to build it in..

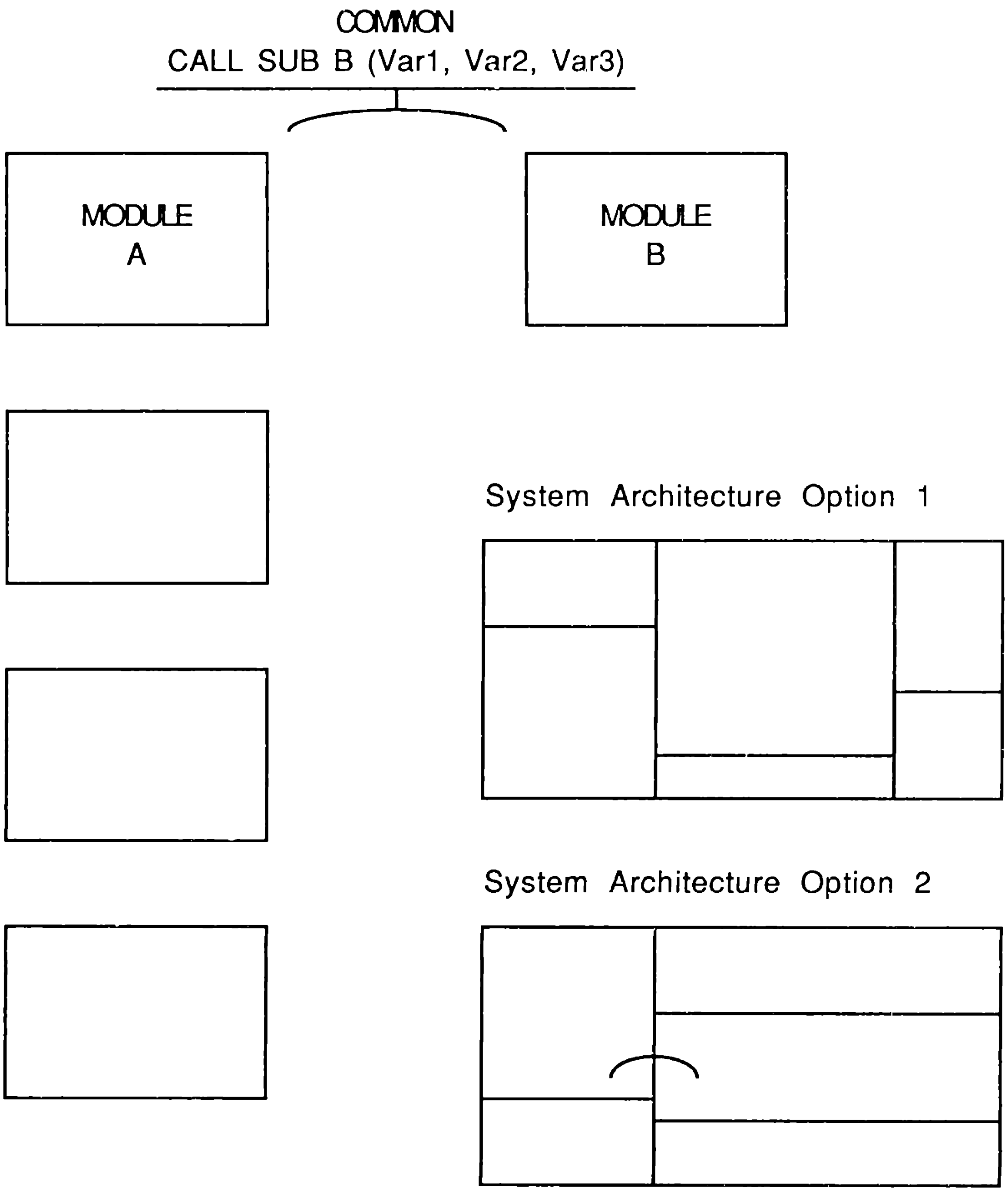
Unknowable flexibility requirements relate to requirements such as new technologies or new regulations for which there is insufficient information to build exact functionality into systems in the near future. The areas in which flexibility is important, however, are known. An example might be an audio electronics dealer being unsure whether or not to prepare for selling the new DAT (digital audio tape) technology. In general for systems design, it is best to consider as many unknowable flexibility requirements as possible. By creating self-consistent business scenarios about the future, and by comparing the cost and value of the different options, one can pick a range of flexibility in which to "buy an option". The car insurance pricing example in figure 3 demonstrates how one might use probability studies and expected cash flow analyses over various years to determine which functionality requirements are best left alone, built in today, or bought as an option. This latter process involves building "hooks" into the system today to be able to invest in additional code to deliver functionality later at less cost. Expected cash flows tend to be flatter in this case because the discounted value of the cost later is little compared to the high cost of initially building in many expected functionality requirements. If few unknown requirements are built in initially, the system is a "quick and dirty fix", meeting needs today, but inflexible and expensive to maintain. If many unknown requirements are built in initially, usually resulting in a more modular system, there is a high initial cost with a smaller "maintenance" cost. "Buying an option," a hybrid between the two alternatives, means investing a little now to gain the right to exercise that option later at a lower price (see figure 4). Outside of the scope of this thesis, a model could be formulated to determine the cost and value of specific options in the same manner as security options are evaluated (see figure 5).

Unpredictable flexibility requirements relate to issues unthinkable or unimaginable. Examples may include legislation far in advance, unusual economic events, hostile takeovers, etc. The cynanide in the Tylenol case is an example of an unknown and unforeseeable issue. In terms of systems design, one can only do what is best in crises given the tools available.

Conclusion/Recommendations

With cost and time constraints in a rapidly changing business environment, no system can be so flexible as to accommodate all changes, internal and external. However, by recognizing and considering the assumptions which are likely to change, system designers can better balance trade-offs by limiting embedded system assumptions in unknowable areas and revisiting business strategies before final system design. By matching the number and type of functionality requirements built into the architecture to the levels of certainty in the requirements, and building hooks to anticipate certain unknown requirements, the result is increased flexibility in information systems.

Figure 1



Areas represent functionality requirements.
Boundaries should be designed initially
to match management functions and
business needs.

"The most important part in designing any modular system is to correctly define the boundaries of the modules."

Figure 2

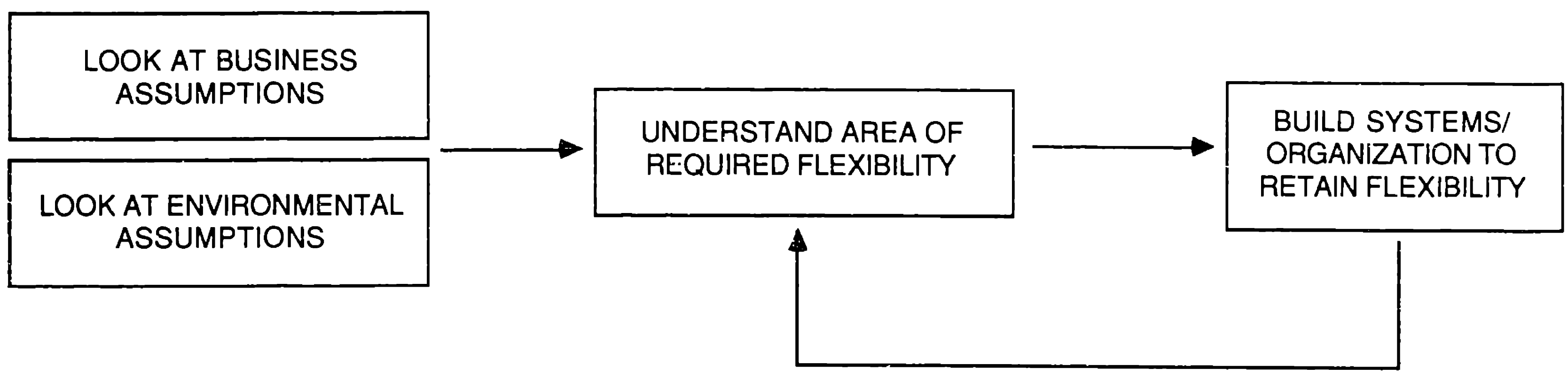

LOOK AT BUSINESS ASSUMPTIONS
LOOK AT ENVIRONMENTAL ASSUMPTIONS
UNDERSTAND AREA OF REQUIRED FLEXIBILITY
BUILD SYSTEMS/ ORGANIZATION TO RETAIN FLEXIBILITY

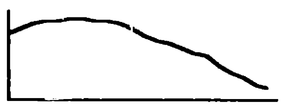
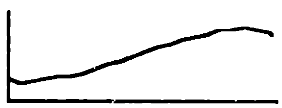
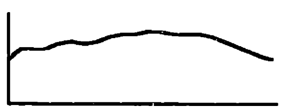
Figure 3

For example: In designing a decision support system for a car insurance company,let us examine one set of "unknowable" requirements, the issue of pricing car insurance.

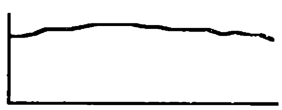
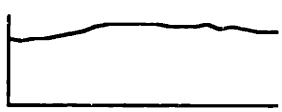
Assume that some known requirements have already been addressed and the system presently incorporates a driver's sex, age, and type of car in determining the final rate.

For different years,

	Year 2		Year 5	Year 10
Functionality Requirements	Probability of needing requirement		Probability of needing requirement	
Keep system as is	0.8		0.7 →	
Include past driving record	0.3		0.4	Probability curves flatten out
Include city	0.1		0.3	

	Investment	Expected cash flows (Return on investment)
Keep as is	$0	
Include record	$5M	
Include city	$2M	

If "buy option" or "build hooks,"

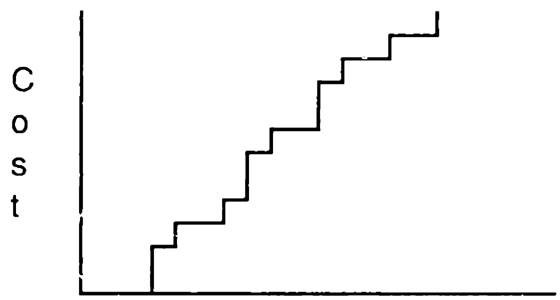
	Investment	Expected cash flows
Include record	$100k today + $2M then	
Include city	$200k today + $1M then	

Cash flows tend to be flatter

Figure 4

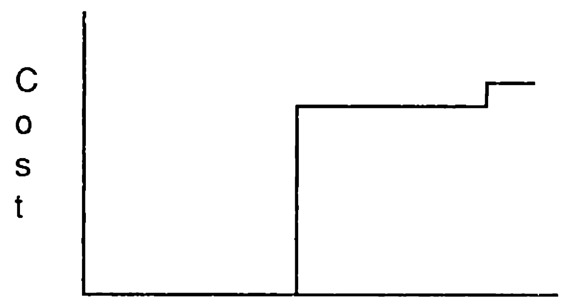
Given unknowable functionality requirements,
we have the option to: Not Build, Build, or Build Hooks

"Quick and dirty fix,"
Expensive to build in new
functionality requirements



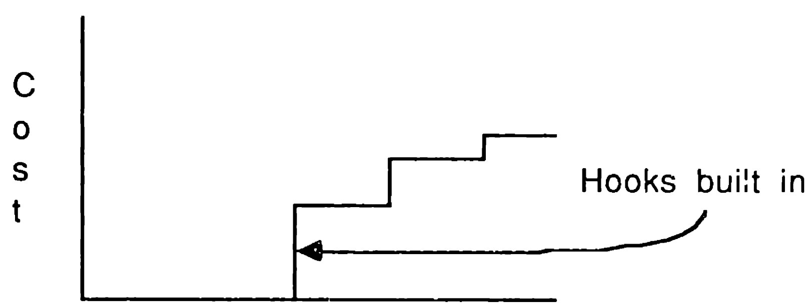
# of functionality requirements
built into system

Initially build in a lot.
Expensive, but less
costly to maintain.



# of functionality requirements
built into system

Build in functionality "hooks"
to later add in at less cost.



# of functionality requirements
built into system

A question of balancing . . .

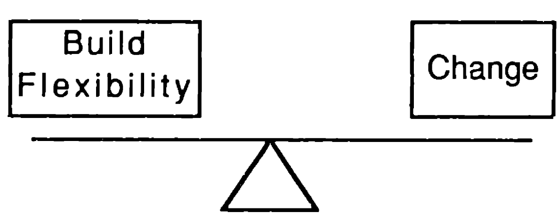


Figure 5

Preliminary Model Attempt

Objective Function

Minimize: w.r.t. C^*

$$\underbrace{\underset{\text{minimum}}{\text{Cost}(C_o)} + \underset{\text{fixed}}{(\text{Slope})(C^* - C_o)}}_{\text{initial cost}} + \underbrace{\underset{\text{discounted}}{(\text{Slope}) \sum_{n \geq n^*} (C_n - C^*) \frac{1}{(1+\beta)^n}}}_{\text{later cost}}$$

C^* - optimal # of correct functionality requirements

C_o - minimal # of correct functionality requirements (meets known needs)

C_n - # of functionality requirements needed at year n

n^* - year when flexibility is required, additional functionality required

n - year

β - discount rate (high for rapidly changing information systems technology)

Slope - cost to build functionality/# of correct functionality requirements (see fig. 5)

THESIS INTERVIEW INSTRUMENT

I. INTRO

My name is Lieven VanMarcke. I'm a senior at M.I.T. and am presently doing an undergraduate thesis studying the concept of flexibility as it relates to information systems. Would you mind taking a few minutes to give your views on several questions I have?

II.ROADMAP TO DISCUSSION

I'm going to start with an open-ended question and then get into a few more specific questions to which your opinions and perhaps supporting examples would be most helpful.

III.QUESTIONS

Open-ended

How would you define a "flexible information system?"

Goals

What in terms of benefits does flexibility give you? Prioritize cost, time, other?

Scope

Where is it most important to build flexibility? Must have the flexibility to change what?

Means

What does it take to make a system flexible? Does the cost to build a system necessarily make it more flexible? How do you justify extra cost to build in flexibility (cost vs. flexibility)? What tools and techniques make a system flexible?

IV.CONCLUSION

Thank you for your time, may I call you sometime later if I have further questions?

Quotes from Interviews

John Kogan - Partner, Arthur Andersen

"Flexibility supports growth given changes in volume, functionality, new needs, product lines, and business needs."

"Goals - Allows company to cut cost, time to support new endeavors."

"Modular system - allows changes to be made with destroying the architecture."

"Must have a systems development methodology to have systems plans congruent with business plans."

"The role of the user is in the functional specification stage, the d/p people take over in the technical stage."

Cecilia D'Oliveira - Director of Information Systems Planning, M.I.T.

"Goals - here at M.I.T., where things don't change too quickly, cost is the major issue."

"In business, organizational structure constraints exist because of assumptions."

"Design data model with a vision, must provide users with data."

"Design techniques - inputs and outputs well defined, modules allow east change."

"Justify costs by getting others to understand the life cycle of systems."

"Part of this education process - collect historical data to show cost to build (systems) and when it became obsolete."

Tom Davenport - Director of Research (The Index Group)

"Flexibility - Do what you want to do when you want to do it."

"Benefits - Less cost and shorter time to develop, and permits development by non-professionals."

"Trade-off between flexibility and reliability"

"Modularity - Can slide functionality in and out"

"Costs more to build a modular system than an integrated."

"Ultimate flexibility - Quick and dirty fix, then throw systems away and start over."

"Anticipating future needs is more an art than a science."

"The less you invest initially, the more you are willing to throw it away; the more it costs, the less likely you are to change it."

"Costs more to build a modular than an integrated system."

John Henderson - Professor, MIS, Sloan School of Management

"Flexible systems allow me to translate, convert a concept into a product in a minimum amount of time."

"Flexibility should be measured in time to market, not efficiency."

"Reflect on integration of activities, rather than composition; integration is the key."

"Example - Series of sequential tasks never done as fast as in parallel; orders of magnitude impact on time to market."

"When building in flexibility, the coordination cost is higher."

"To get modules, introduce independent problem solvers, decouple; the cost to get these loosely coupled systems is a coordination cost."

"With flexibility, one can take advantage of opportunities. this justifies opportunity costs."

Paul Cunningham - Manager, Touche Ross

"Flexibility reduces time to respond, cost, and lowers technical risk."

"Scope - Data structures, traditionally most difficult to change, functional requirements, and table changes."

"Know when to use table changes; know when to put hardcoded information in a separate table."

"Means - the key is in the design, starts in analysis where you identify needs for change."

"Data structure, after analysis, select tools."

"Use tables for hardwire information that could be external, use modular design and use well structured code."

"Cost vs. Flexibility, no easy correlation."

"Must weigh anticipated risk in ignoring change vs. the time and cost to build up front."

Also spoke informally with:

Frederick B. Baldwin, Quality Assurance Manager, Stone & Webster

Erik H. VanMarcke, Professor of C.E. and Operations Research, Princeton

Footnote

(1) Private Communication, John Kogan, March, 1988

(2) Private Communication, Tom Davenport, March 1988

(3) Anthony, R.M. "Planning and Control Systems: a Framework for Analysis
Harvard University Press (1965).

Journal Resource List

Computer and Control Abstracts - back to 1980

Information Processing & Management - back to 1978

Information Systems - back to 1975

InfoWorld - back to 1985

Management Information Systems Quarterly - back to 1981

Management Information Systems Weekly - back to 1987

Sloan Management Review - back to 1980

References

Anthony, R.M. *Planning and Control Systems: a Framework for Analysis.* Harvard University Press (1965).

Barros, O. *Information Requirements and Alternatives in Information System Design* Information Systems Vol. 12, No 2, pp. 125-136 (1987)

Barros, O. *Management Information Systems Structure, Types and Integration* Information Systems Vol. 6, No.4, pp.243-254 (1981).

Bakos, J.A. *Information Technology and Corporate Strategy: A Research Perspective* MIS Quarterly (1986)

Brancheau, J.C. *Information Architectures: Methods and Practice* Information Processing & Management Vol. 22, No. 6, pp.453-463 (1986)

Kampfner, R. *A Hierarchical Model of Organizational control for the Analysis of Information System Requirements* Information Systems Vol. 12, No. 3, pp. 243-254 (1987).

Kampfner, R. *Formal Specification of Information Systems Requirements* Information Processing & Management Vol. 21, No. 5, pp 401-414 (1985)

Kobayashi, I. *Information and Information Processing Structure* Information Systems Vol. 1 pp. 39-49 (1975).

McKeen, J.D. *Successful Development Strategies for Business Application Systems* MIS Quarterly (1983)

Parsons, G. *Information Technology: A New Competive Weapon* Sloan Management Review (1983).