Comprehensive Design for Information Systems with Integrated Systems Design Methodology

Dr Charles C. Willow
School of Business Administration, Monmouth University,USA

Abstract

Systems Analysis and Design (SAD), by far, has been the choice of many practitioners for articulating a gamut of requirements and processes associated with Information Systems Design. In the midst of the Internet-driven real-time and dynamic information era, however, the focus of many information systems managers has shifted from traditional design of systems to making timely, qualitative decisions pertaining to the choice of the fittest information system for their organizations. Moreover, the demand for embedded systems has brought insurmountable time constraint on such management parameters as product turnaround or lead times to market.

This paper presents 'Integrated Systems Design Methodology (ISDM)©, a unique systems design apparatus, which may be employed as an alternative to SAD in many instances. Unlike SAD, ISDM provides the managers with a balance of *qualitative* and *quantitative* aspects as variables for a wide range of information systems, both generic and customized. Two major critiques of SAD from the literature comprise: difficulty of quantification for the entire process and prohibitively long lead times for design completion.

Keywords: Systems Analysis and Design, Information System, Information Systems Design, Software Engineering, Qualitative Decision Making, Quantitative Decision Making, Multiple Criteria, Utility Functions, Ranking and Selection, Operations Research.

I. Introduction

A common problem associated with designing and/or modeling systems is the lack of interest for and insufficient measures taken to both verify and validate the design or model itself before implementation to seek its solution(s). Importance of an *a priori* testing mechanism for the design of systems was emphasized by Willow (1999, 2006) and earlier by such critics as Wymore (1981) and Schmidt (1986).

A *system* is defined as a combination of elements and components – mechanical, electrical, thermal, hydraulic, pneumatic, and even people and information – synthesized into a complex for performing a function and satisfying a need (Willow, 1999). As a consequence, there is a constant need for an *wholistic* approach comprised of designing, engineering, implementing, and testing a system. In addition, for the generally qualitative models of a system to be reusable and to be provided as references for similar efforts, there is almost always the requirement for its *quantification* to a larger extent.

The objective of this paper is to introduce a general-purpose systems design apparatus, the 'Integrated Systems Design Methodology (ISDM)© (Willow, 1999) and its extensive applications to a wide variety of systems. In particular, an application of ISDM to information system will be the primary focus. ISDM provides both the designers and management with a balance of *qualitative* and *quantitative* aspects as variables for a wide range of information systems, both generic and customized.

This paper is organized as follows:
In section II, extensive review of literature for information systems design is conducted. Section III delivers the overview of ISDM© and its implications to the design of systems. Section IV, which is the key section, provides the application of ISDM to information systems design. Conclusions and notes for future research subsequently follow in section V.

Ii. Design of information systems

Design of information systems has interested innumerable number of researchers both in the academe and practice for the past five decades. With the advent of the 21st century, dynamic real-time information systems have become the prerequisite resource if not the norm for most organizations. Moreover, the demand for complex systems applications such as embedded systems has brought insurmountable time constraint on such management parameters as product turnaround or lead times to market (Hellestrand, 2005*b*). As a consequence, building and reconstruction of designs for these information systems is the challenge which requires attention from every aspect of the organization.

Systems Analysis and Design (SAD), by far, has been the choice of many practitioners for articulating a gamut of requirements and processes associated with Information Systems Design (ISD). Willow (2004) describes SAD to be a method which employs a traditional cascade structured model for eliciting and formulating the necessary data for an ISD problem, followed by identifying the relationships and processes for the data entities before implementation. Verification and Validation (V&V) pursue to terminate an iteration of the SAD cycle. Figure 1 depicts the milestones of the SAD lifecycle.

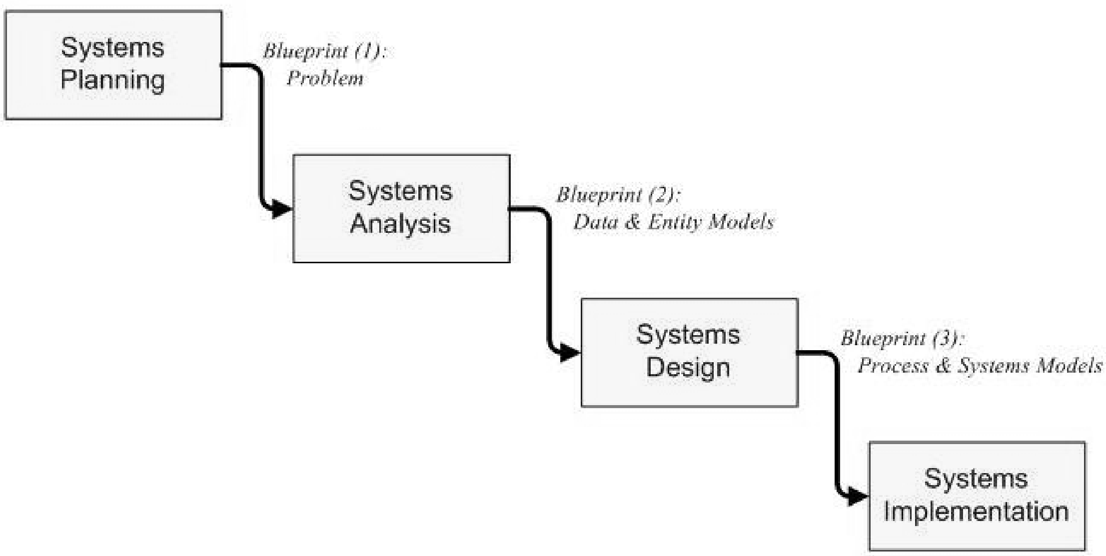


Figure 1. Systems Analysis and Design Lifecycle

There is a precedence relationship between the four major steps of Figure 1: Systems Planning, Systems Analysis, Systems Design, and Systems Implementation. Hence, the modularity of SAD remains relatively low, inhibiting parallel development. This resonates to difficulty in real-time development for and dynamic upgrades to 21-century information systems using SAD as the primary vehicle for modeling. Each step of the SAD lifecycle has a set of activities to be completed recursively, which follow in Figure 2 (Willow, 2004).

A formal design of an information system is acquired at the completion of the 'Systems Design' activities. In particular, the entire flow of data, as well as the process of information, are well specified and at times, articulated in the Data Flow Diagram (DFD). The DFD in turn serves as the blueprint for developing the β-version of the information system in 'Systems Implementation', followed by V&V. There is a major drawback however, implicit in the 'Systems Design'; there seems to be no stopgap for neither verifying nor validating the design itself, before being presented to 'Systems Implementation'. This is due in part to inherently qualitative nature of the DFD, but in most part to little if not no attention to the subject in the literature. This paper proposes a method to both verify and validate the Information System Design (ISD) itself by applying the 'criterion models' of the 'Integrated Systems Design Methodology (ISDM)© (Willow, 1999) to DFDs. The ISDM© is expected to generate *quantified* DFDs for comparisons, as well as providing *a priori* V&V on a more objective scale.

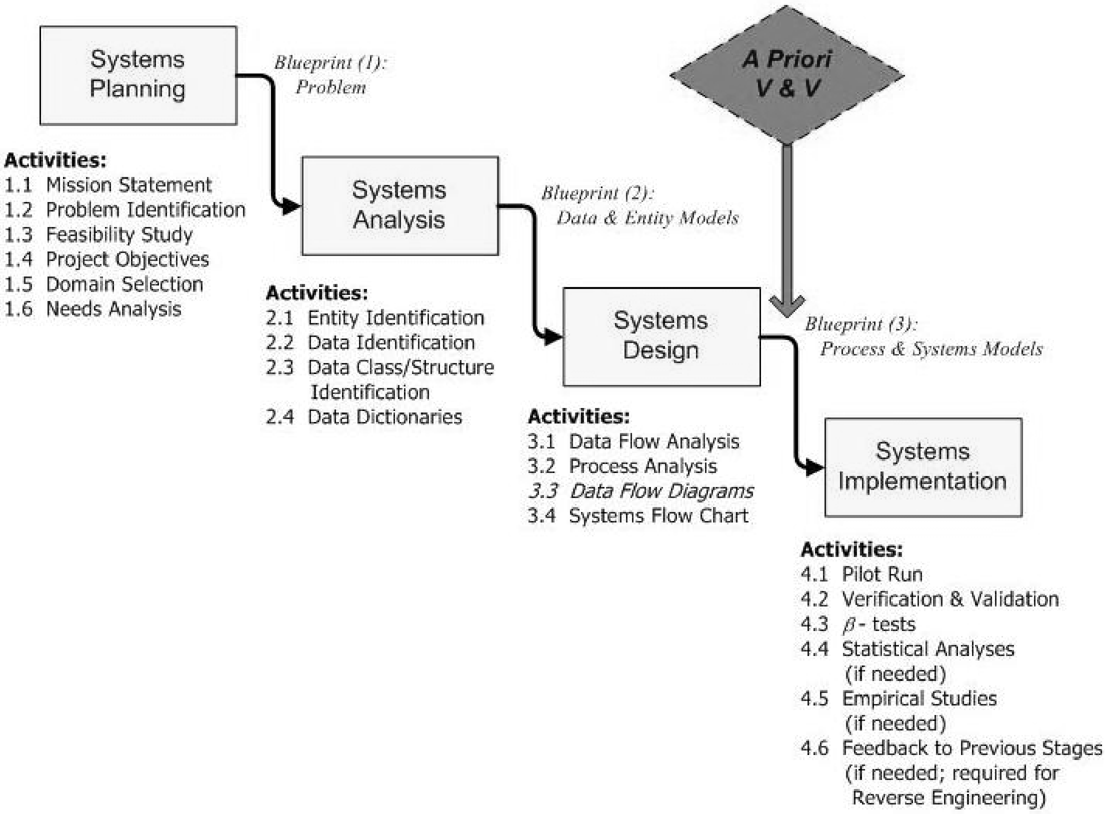


Figure 2. Activities of the Systems Analysis and Design Lifecycle

Kendall *et al* (2005) portrays similar view on SAD, however, seem to emphasize implementation issues during the entire design process. For example, Computer Aided Software Engineering (CASE) is stressed, which is explicitly included in the SAD lifecycle as 'Upper CASE Toolset' and 'Lower CASE Toolset', respectively. Further emphases on systems implementation by Kendall *et al* (2005), perhaps to expedite real-time prototyping, follow by introducing alternatives to SAD such as the Rapid Application Development (RAD). Whitten *et al.* (2001) also illustrates a waterfall model of the SAD, based on early ideas of Davis (1983) and Yourdon (1984).

Earlier research completed by Thomas *et al.* (1981) and Fritz (1987), respectively, illustrate the effectiveness of SAD. While Thomas *et al.* (1981) compare the SAD side by side with the Problem Statement Language (PSL) developed by the University of Michigan team in the late 1970's, Fritz (1987) demonstrates the effectiveness of SAD instruction at a higher educational institute.

Go *et al.* (2004) introduce temporal addenda to classical SAD, specified as 'scenarios'. As accurate as it may be, however, a scenario may award the designer with higher subjectivity in completing a SAD lifecycle, thereby increasing the immeasurable qualitative feature of the overall systems design.

Gabbert (2001) attempts to integrate the two major methods in Information Systems Design (ISD), 'SAD' and 'Object-Oriented Analysis and Design (OOAD) (Coad et al., 1991a and 1991b)'. Critiques are made regarding the current practice of employing these two methods in that they are necessarily complementary not contrasting building blocks for ISD. For example, Gray (1988) has been an advocate for a complete migration to OOAD from any previous ISD methods. In addition, he believes the OOAD to become the de facto standard for ISD. Unfortunately, Gabbert (2001) failed to realize some critical problems associated with the integration of the two: SAD and OOAD. On the one hand, 'reusability of design' for real-time information system development is not guaranteed simply by using OOAD as the mainstream. On the other hand, by increasing the impact of OOAD on the overall system design, where applicable, quantification becomes almost impossible to attain, despite the benefits she outlined. Importance of 'design quantification' was stressed earlier in this section.

Marculescu et al. (2001) present an excellent approach to and demonstration of a systems design for an embedded system by using a classical stochastic model. As a result, design quantification was partly achieved with the conjunction of probabilistic reasoning with ISD.

Cheung et al. (1994) complete an ISD to yet another frequent real-time application, 'control system'. Their provision of mathematical insights was relatively sound and clear. However, the quantitative model was insufficient to safeguard such key issues as reusability and comparison for ISD.

By and large, topics comprised of visualization, graphical representation, Human-Computer Interaction (HCI), and/or Computer-Supported Cooperative Work (CSCW) seem to dominate the recent ISD-related literature. In contrast, relatively small proportion of published findings is dedicated to analyzing quantitative models for ISD. Moreover, to date, most ISD literature have not carried the motivation to quantify the design per se for more objective comparisons and testing. Early research by Botting (1986) pinpoints the need for automation, as well as graphical representation and restoration of ISD. Turetken et al. (2004) propose a graphical development tool for ISD which delivers multi-dimensional pane structure, similar to frames of the web browser. Hahn et al. (1999) take on more psychological view to examine the various HCI factors by introducing different sets of symbols utilized in ISD. Similar research objective that concerns HCI and CSCW are met by Felder et al. (2002) with an application to real-time systems. Singh et al. (2003) conduct an ad hoc comparative analysis of the ISD methodologies in terms of HCI factors involved. A classical CSCW research is reported by Pernici et al. (1989), which proposes a dedicated ISD tool for the Office System. A more generic suite of software tools are developed by Mayer et al. (1995), entitled the IDEF series. However, Mayer et al. (1995) themselves have admitted in their research that even semi-automated design comparisons are unavailable. Thus the validity of an ISD still remains at the discretion of the human designer.

Another mainstream group of ISD research is deemed to have been formed by extending the application domains for existing design methods. Baskerville (1993) presents an ISD for system security by incorporating conventional Data Flow Diagrams (DFDs) and Data Dictionaries (DD's) for functional representation of the system. One of the earlier ISD applications to embedded systems is supported by Yue (1989), in which he embarks on functional decomposition inspired by DFD to characterize behavioral constraints of the complex real-time system. Application of DFD to aeronautic safety analysis is demonstrated by Fenelon et al. (1994).

In essence, today's information systems managers are challenged by the Internet-driven, real-time, dynamic, and even ubiquitous demand created by their clients. As a consequence, the focus of many information systems managers has shifted from building traditional design of systems to making timely, qualitative decisions pertaining to the choice of the fittest ISD for their organizations. Applications of the ISDM© to ISD are sought in the subsequent sections. Section III introduces ISDM©, followed by its application in section IV.

Iii. Integrated systems design methodology

When large-scale systems are approached, decisions made by the designer-modeler-planner almost always are made under the limitations of incomplete data, inadequate information, and the uncertainties associated with the randomness of the processes or activities under study (Willow, 2006). These limitations usually exist to some extent in each decision step of the process, and hence are compounded as many times as there are steps in the process by the degree of inaccuracy or uncertainty associated with each decision. Thus it is relatively easy to understand why any process can produce results which, at best, minimize the risks associated with decisions made under uncertainties. Hence, there is a constant need to develop an integrated, systematic, yet robust methodology for the design of systems.

The Integrated Systems Design Methodology (ISDM)©, originally developed by Willow (1999), allows the designer to organize his/her qualitative thoughts into a structured waterfall model, while quantitatively mapping the properties of the design extensively. It is a general-purpose modeling framework for building a wide variety of system designs, as illustrated in the examples in Willow (2006). ISDM is distinguished from traditional and even the latest ad hoc structured design techniques such as IDEF suite of methods (Mayer et al., 1995) in that it does not exclusively restrict itself as a modeling method. Instead, a set of probabilistic criterion models, which provide a balance between qualitative and quantitative decision making, equip the designers with the selection for optimal solution. A 'design criterion' is a global attribute or property of the system, usually mapped from directly measurable 'design parameter(s)'. Nomenclature, mathematical insights, and algorithm for the optimal system selection with criterion models of ISDM follow in sections III-1 and III-2. The object is to generate an 'evaluation function' to attain an ordinal ranking of the alternatives on a cardinal scale. The scope of the ISDM usage for this paper is illustrated in Figures 3 to 5.

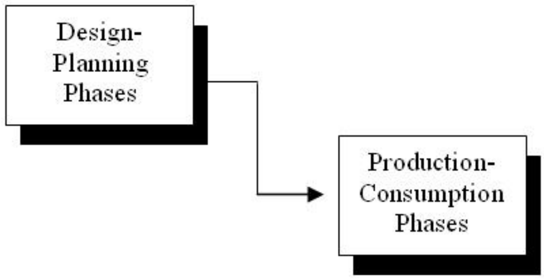


Figure 3. Phases of the Integrated Systems Design Methodology (Level-0)

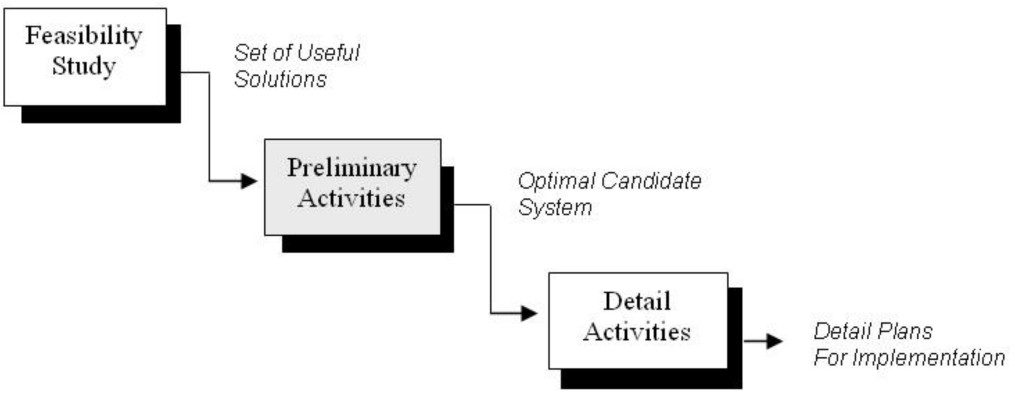


Figure 4. Decomposed Steps of the Primary Design-Planning Phases (Level-1)

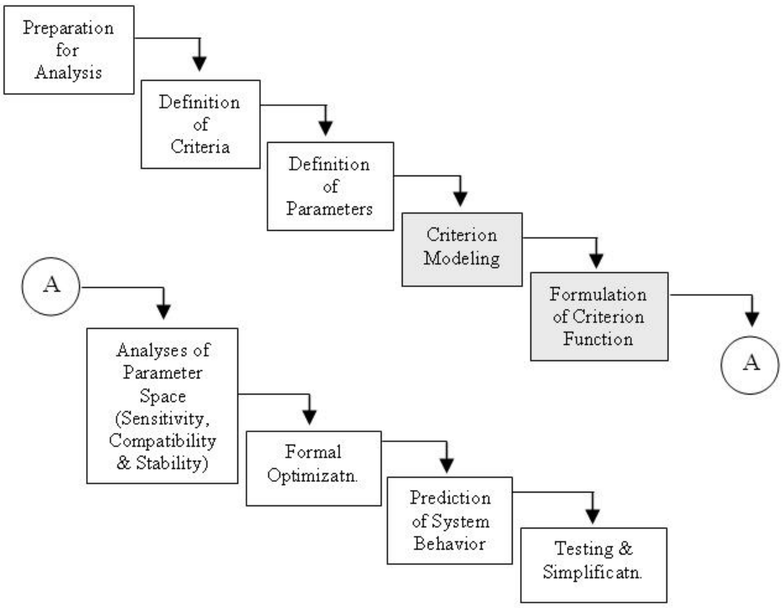


Figure 5. Decomposed Steps of the Preliminary Design (Level-2)

Note the optimal selection-based criterion functions pertain to the activities, 'Criterion Modeling' and 'Criterion Function Formulation' of the 'Preliminary Design'. The reader is referenced to Willow (2006) for a gamut of activities associated with the entire phases of the ISDM.

It is precisely this balance of qualitative and quantitative decision making that ISDM provides, which may allow a series comparisons of Information System Designs (ISD) for testing. More often, however, this may bring a validation ground for ISDs before their physical implementation, as an end effect. Thus the primary contribution of this paper is to extend the application of ISDM to the area of ISD validation (i.e., *a priori* V&V) and possibly to the selection of optimal ISD for real-time implementation.

III-1. Nomenclature

III-1-1. Subscripts and Superscripts

- i — Index for the criteria and relative importance/weight associated.
- j — Subscript for submodels.
- k — Subscript for parameters.
- v — Superscript for *intervals* in the range of criterion i; $v = 1, \ldots, \xi_i$; applicable for Classes II, IV, VI, and VIII (Figure 6).
- V — Superscript for intervals of the system model; $V = 1, \ldots T$; for II, IV, VI, and VIII.
- α — Candidate system or alternative index.

III-1-2. Parameters

- m — Total number of candidates or alternatives for the (design-planning) model; $\alpha = 1, \ldots, m$.
- n — Total number of criteria in the model; $i = 1, 2, \ldots, n$.
- T — Total number of *intervals* of the system model; $T = 1$ for Classes I, III, V, and VII; varies for Classes II, IV, VI, and VIII.

III-1-3. Variables

- x_i — i-th criterion. A (global) attribute of the system. Examples of criteria comprise economic value of a system, customer satisfaction, and so forth.
- $R(x_i)$ — Range of x_i; $R(x_i) = x_{i,\,max} - x_{i,\,min}$
- $\mathbf{X}_i$ — Normalized x_i in the *probability* space. Two primary methods of normalization are:

  1) Linear Interpolation (*Absolute*);

$$\mathbf{X}_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} = \frac{x_i - x_{i,\min}}{R(x_i)}$$

  2) *cdf* Method (*Relative/Statistical*);
     i. Classical – Probability distribution fit, followed by Goodness-of-Fit (GOF) tests, such as the Kolmogorov-Smirnov (K-S) or Chi-square tests. *Statistical independence* among data is assumed.
     ii. Empirical – Direct samples from data; Pr{E} = Occurrence(E) / Ω, where, Ω is the sample space.

- a_i — Relative importance or weight of the i-th criterion; $a_i = f_j(x_i)$.
- ξ_i — Number of *intervals* for the i-th criterion; for Classes II, IV, VI, and VIII (Figure 6).
- β_i^v — Variable for designating the v-th interval for the i-th criterion; applicable for Classes II, IV, VI, and VIII.
- y_k — k-th parameter. A parameter is the (primitive) attribute or feature of the model. It should be *directly* measurable, and should be compact, consistent across all candidates, and complete (*viz*. exhaustive).
- z_j — j-th submodel. A submodel is a set of mapped parameter(s). Hence, $x_i = f_i\{z_j\}$; $z_j = g_j\{y_k\}$; $xi = f_i\{\, g_j\{y_k\}\}$.
- B^V — Variable for designating the V-th interval for the model; applicable for Classes II, IV, VI, and VIII.

CF_α Criterion Function for the α-th candidate (system) or alternative;
$CF_\alpha = h_\alpha\{a_i, x_i\} = h_\alpha\{a_i, f_i\{g_j(y_k)\}\}^1 \Rightarrow h_\alpha\{a_i, \mathbf{x}_i\}$

Theoretical foundation for the indices, parameters, and variables aforementioned is

$$= h_\alpha\left\{a_i, \frac{f_i\{z_j\} - x_{i,\min}}{x_{i,\max} - x_{i,\min}}\right\}$$

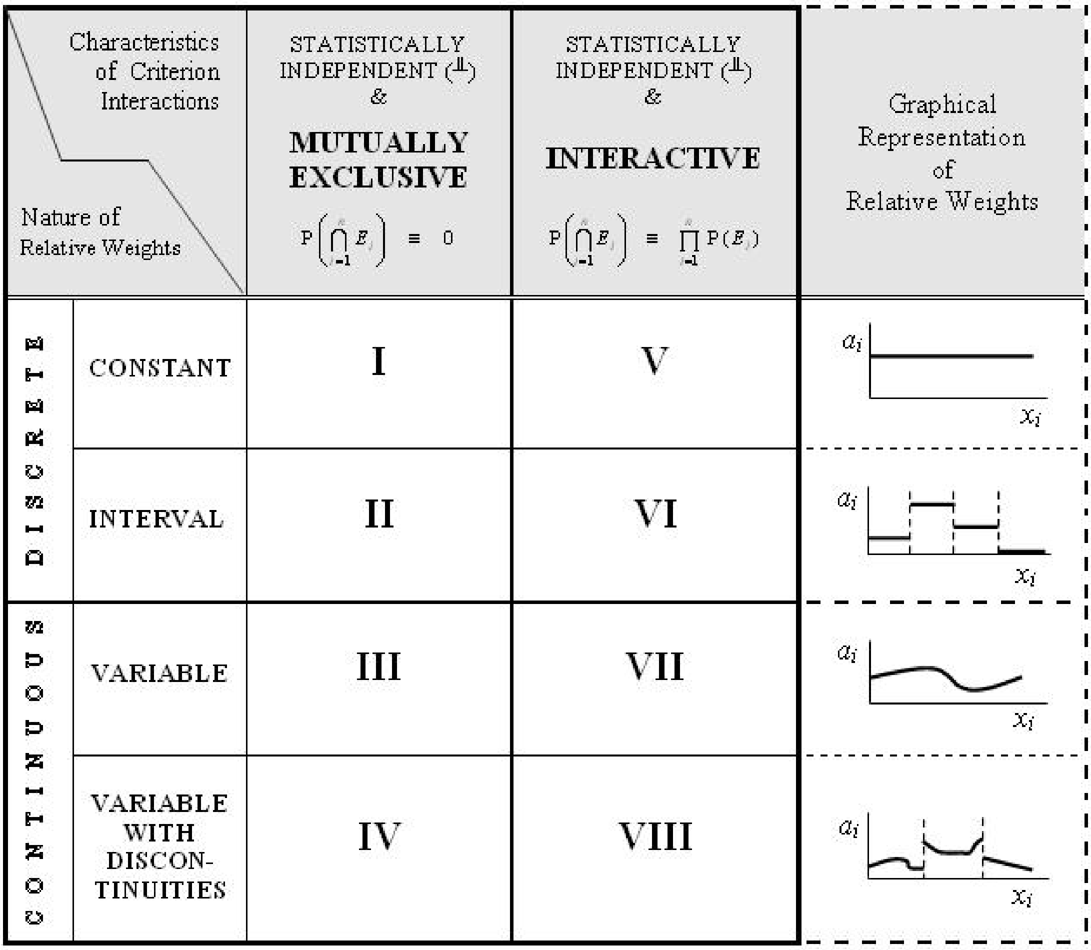
$$= h_\alpha\left\{a_i, \frac{f_i\{g_j(y_k)\} - x_{i,\min}}{x_{i,\max} - x_{i,\min}}\right\}$$

$$= h_\alpha\left\{a_i, \frac{\prod_{\forall j} g_j(y_k) - x_{i,\min}}{x_{i,\max} - x_{i,\min}}\right\} : f_i = \Pi$$

$$= \sum_{\forall i} a_i \left[\frac{\prod_{\forall j} g_j(y_k) - x_{i,\min}}{x_{i,\max} - x_{i,\min}}\right] : h_\alpha = \Sigma$$

detailed in Willow (2006).

Eight possible classes of criterion function models are hitherto discovered, relative to the mathematical representation of each criterion relative weight. The relative weight, a_i, represents the *utility* (Chung, 1994) of the i-th criterion, x_i. Henceforth, the weight, relative importance, and utility will be interchangeably used throughout this paper. Figure 6 summarizes the classes of possible criterion function models.



1 For Mutually Exclusive Criteria, x_i, only. Criterion interactions will involve a_{ij}… and x_{ij}…

Figure 6. Classes of Criterion Function Models

Each class represents criterion-to-relative-weight function model. Models V through VIII are simply extensions to Models I through IV, respectively, in that (higher-order) *criterion interactions* are included in the analysis. Figure 7 depicts Model VI, a *linear* version of Model VIII in Figure 9.

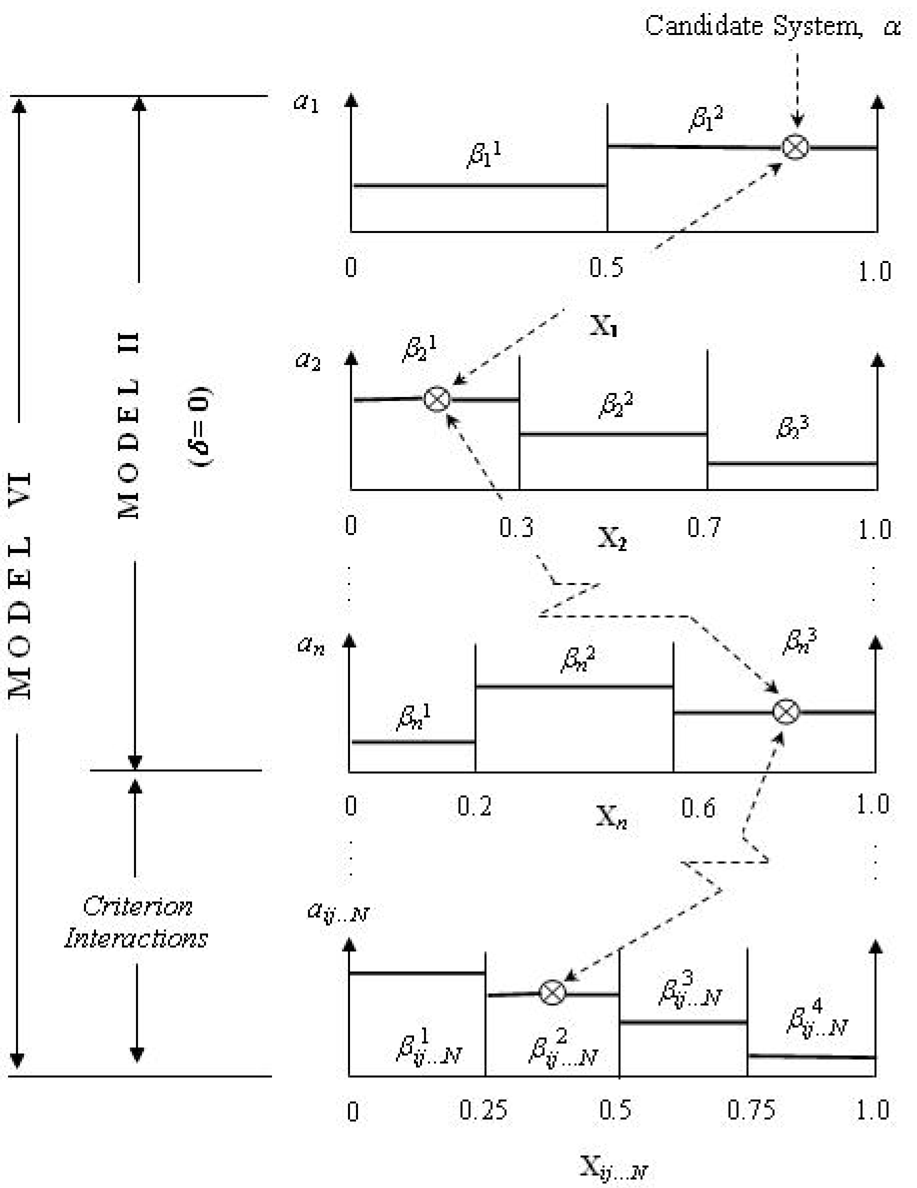


Figure 7. Constant Relative Importance, Mutually Exclusive Criteria with Multiple Intervals (Model II); Interactive Criteria with Intervals (Model VI)

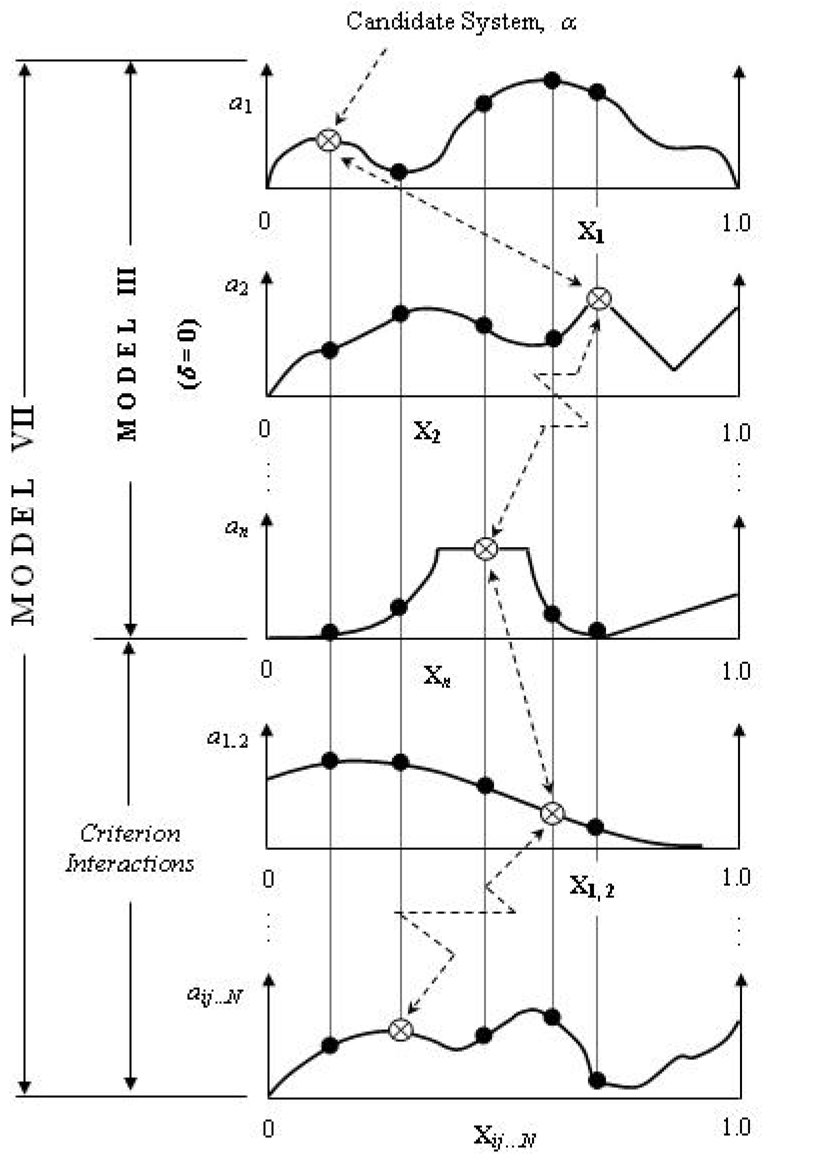


Figure 8. Varying Relative Importance, Mutually Exclusive Criteria (Model III); Interactive Criteria (Model VII)

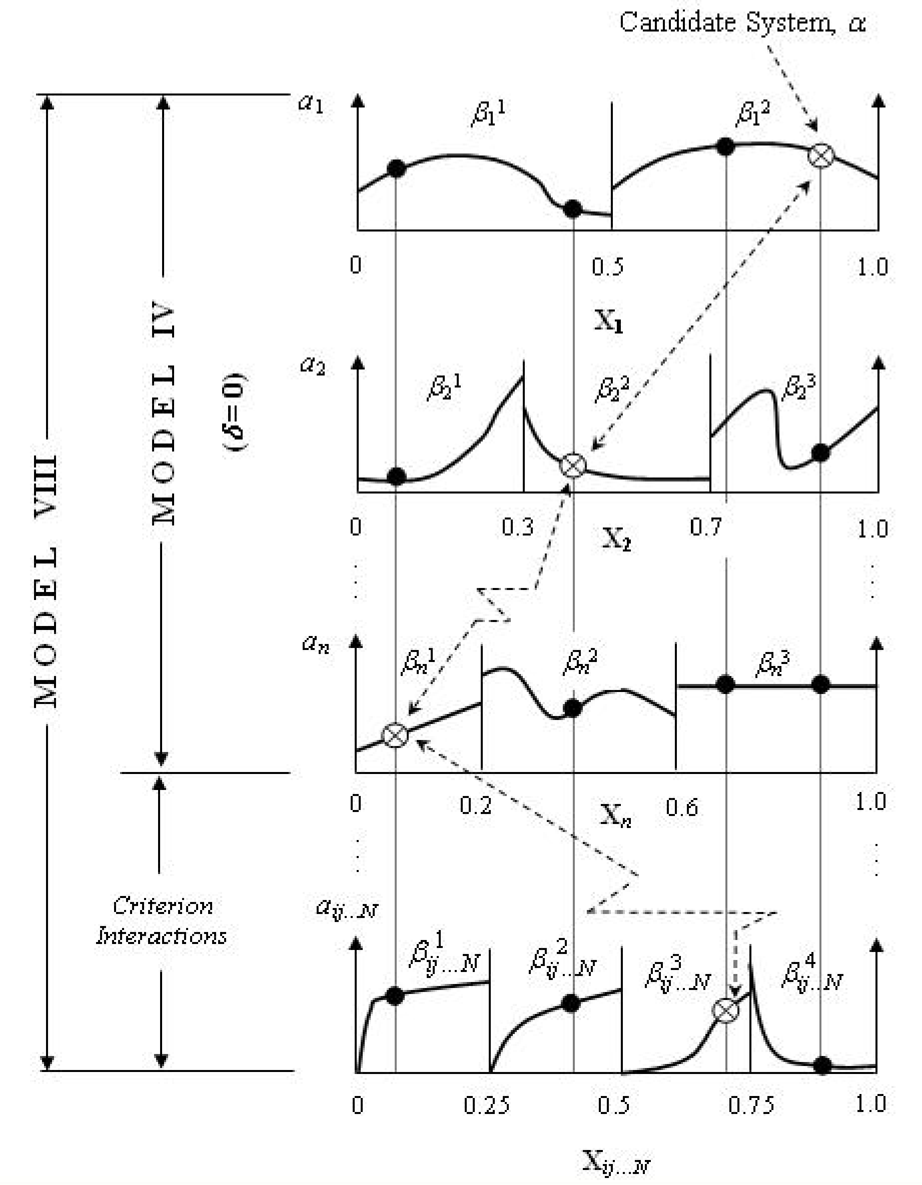


Figure 9. Varying Relative Importance with Multiple Intervals, Mutually Exclusive Criteria (Model IV); Interactive Criteria (Model VIII)

In essence, the evaluation function, CF_α, reflects a perfect balance between as well as a blend of qualitative (a_i) and quantitative (x_i) properties of the system design.

III-2. Evaluation Function

The essence of the evaluation function, CF_α, is to acquire an *ordinal ranking* from a set of alternatives against a normalized, *cardinal scale*. A generalized algorithm for evaluating Class VI is summarized. Other Classes follow comparable logic:

1) Map all x_i into $\mathbf{X}_i$ (*i.e.,* normalize to unity).
2) Derive all *criterion intersections*, $\mathbf{X}_{ij\ldots n}$.
3) Obtain the values of $a_i^{\beta_i^v}$ and $a_{ijk\ldots n}^{\beta_i^v}$ for all intervals, β_i^v.
4) Find the smallest *first* interval, B^1, from among the $\mathbf{X}_i$ and $\mathbf{X}_{ij\ldots n}$. Note that B^1 accounts for the smallest fragment for all i, and should be distinguished from β_i^v.
5) Establish B^1 for all $\mathbf{X}_i$ and $\mathbf{X}_{ij\ldots n}$. (In Figure 7, B^1 = [0, 0.2) from $\mathbf{X}_n$).

6) Establish the second overall interval, B^2, from the end of B^1 to the next β_i^v limit. (In Figure 7, B^2 = [0.2, 0.25) from $\mathbf{X}_{ij\ldots n}$). Hence B^2 extends to all $\mathbf{X}_i$ and $\mathbf{X}_{ij\ldots n}$.
7) Establish the next interval from the end of B^2 to the next β_i^v limit. Thus B^3 = [0.25, 0.3) from $\mathbf{X}_2$ for Figure 7.
8) Repeat step 7 until the last interval, B^T, is reached. To verify T (V = 1, …, T), the following formula is employed:

$$T = \left(\sum_{i=1}^{n} \sum_{v=1}^{\xi_i} \beta_i^{\,v} \right) - (n-1) \qquad (1)$$

where, n = total number of criteria in the model;
ξ_i = number of intervals for criterion i.

9) Vertical Normalization among Criteria (Inter-Criteria): Normalize the a_i values in each interval B^V, so that

$$\sum_{i=1}^{n} a_i^{B^V} + \sum_{\substack{\forall i,j,\ldots N \\ i \neq j \\ \vdots \\ N-1 \neq N}} a_{ij\ldots n}^{B^V} = 1.0; \quad V = 1, \ldots, T. \qquad (2)$$

That is, we are *vertically* normalizing to unity, the a_i values for each (overall) interval, defined by B^V. Recall that V is the interval index across all $\mathbf{X}_i$ and $\mathbf{X}_{ij\ldots n}$. For instance, the relative importance for $\mathbf{X}_1$ in the first overall interval (V = 1) determined is defined by

$$a_1^{B^1} = \frac{a_1^{\beta_1^1}}{\sum_{\forall i} a_i^{B^1} + \sum_{i,j,\ldots,n} a_{ij\ldots n}^{B^1}} \qquad (3)$$

10) Repeat step 9 for all $a_i^{B^1}$ and $a_{ij\ldots n}^{B^1}$, and verify the values by using equation (2).
11) Repeat step 9 for all intervals, V = 1, … T.
12) Identify all $a_i^{B^V}$ for each candidate system, α. That is, identify the $a_i^{B^V}$ and $a_{ij\ldots n}^{B^V}$ corresponding to each $\mathbf{X}_i$ and $\mathbf{X}_{ij\ldots n}$, respectively, within a candidate.
13) *Horizontal Normalization within a Candidate* (*Intra*-Candidate):
For a given candidate system, α, normalize $a_i^{B^V}$ so that

$$a_i^* = \frac{a_i^{B^V}}{\sum_{\forall i} a_i^{B^V} + \sum_{\forall i,j,\ldots n} a_{ij\ldots n}^{B^V}}, \qquad (4)$$

or

$$a_{ij\ldots n}^* = \frac{a_{ij\ldots n}^{B^V}}{\sum_{\forall i} a_i^{B^V} + \sum_{\forall i,j,\ldots n} a_{ij\ldots n}^{B^V}} \qquad (5)$$

14) Normalize interaction terms, subject to the constraint $a_{ij\ldots n} \geq a_{ij\ldots(n-1)}$ (*i.e., conjunction rule*; see section B that follows).
15) Compute for each candidate, α,

$$\mathbf{CF}_{\alpha} = \sum_{\forall i} a_i^* \cdot \mathbf{X}_i + \sum a_{ij\ldots n}^* \cdot \mathbf{X}_{ij\ldots n} \qquad (6)$$

Thus the Criterion Function (CF_{α}) serves as a trade-off, evaluation function. The general form for the criterion function, CF_{α} follows in equation (7).

$$
\begin{aligned}
CF_{\alpha} &= \Pr\left(\bigcup_{i=1}^{n} \theta_i\right) \\
&= \sum_{i=1}^{n} \theta_i - \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \delta_{ij} \cdot \theta_{ij} \quad : \quad \text{1st Order of Interaction} \\
&\quad + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^{n} \sum_{k=1}^{n} \delta_{ijk} \cdot \theta_{ijk} \quad : \quad \text{2nd Order of Interaction} \\
&\quad \vdots \\
&\quad + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j \\ j \neq k \\ \vdots \\ N-1 \neq N}}^{n} \cdots \sum_{N}^{n} \delta_{ijk\ldots n} \cdot \theta_{ijk\ldots n} \quad : \quad N = (n\text{-}1)^{\text{th}} \text{ Order of Interaction}
\end{aligned} \qquad (7)
$$

where,

$$\delta = \begin{cases} 1, & \text{when } \exists \text{ Interaction} \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad \theta_i = a_i \cdot \mathbf{X}_i .$$

The 'axioms of probability' (Mood *et al.,* 1974; Casella *et al.,* 1990; Taha, 1992) serve as the theory behind normalization, both vertical and horizontal. The cardinal scale for the evaluation function was chosen to be the *probability* space, ranging in the interval [0, 1], for x_i as well as a_i.

An application example of the criterion function method to a set of Data Flow Diagrams (DFD) follows in section IV. Notice the DFDs were selected from Information System Design (ISD) exercises with existing data.

IV. APPLICATIONS OF ISDM TO INFORMATION SYSTEMS DESIGN

In this section, both the efficiency and effectiveness of ISDM are demonstrated through a numerical example, in which a DFD containing qualitative data and information is mapped to a quantitative cardinal scale. The quantitative model may then allow itself to be semi-automated, which is expected to reduce the lead time to design considerably. Interestingly enough, the ISDM has proven to both verify and validate DFDs. Verification refers to the general test of performance, during which one asks the question, "Is this system functioning correctly?" On the other hand, validation tests for integrity, with a more profound implication, "Is this the right system to begin with? Is this the system we desired, which meets the objectives set forth and the needs required?" Hence, checking conformity with the general syntax of the DFD suffices to *verify* an Information Systems Design (ISD). Validation may be more elusive, in that the logic behind an ISD needs to be checked for. In effect, the ISDM may be applicable to design comparisons on a *cardinal* (*i.e.,* under comparable domain within a problem context) scale. Needless to mention, design comparisons across multiple (problem) domains may confound even the expert human designers.

IV-1. Design Verification with ISDM

A generic DFD exercise for a payroll system, prepared by Kendall *et al.* (2005), is selected for the numerical example. Given a DFD in Figure 10, the 'Criterion Function Methods' of the ISDM may proceed as follows. The external and/or internal 'entities' along with 'data stores' make up the criteria (x_i), while the 'data flows' represent corresponding relative weights (a_i and $a_{ij\ldots n}$). The 'processes' may indicate interactive criteria ($x_{ij\ldots n}$). Note the DFD-generated criteria are now indicator variables, with implicit sigmoid-function behavior. There is no need for normalization, and thus,

$$\mathbf{X}_i = x_i = \begin{Bmatrix} 1, & \text{if it exists in DFD} \\ 0, & \text{otherwise} \end{Bmatrix} \quad \text{for every} \quad i = 1, \ldots, n. \qquad (8)$$

An incoming data flow to a criterion is considered to have a negative magnitude, while an outgoing data flow is regarded as positive. The designer may assign a value of preference for the data flow in the range of 0 to 1. The object of verification by using ISDM, therefore, is to seek a balanced convergence to the equilibrium (*i.e.*, $a_i = 0$; $a_{ij\ldots n} = 0$) for all utilities, while detecting those outside the range of the probability space.

In essence, Class V of Figure 6 governs the nature of the ISDM applications to DFDs. Table 1 follows to summarize the criterion and utility mappings.

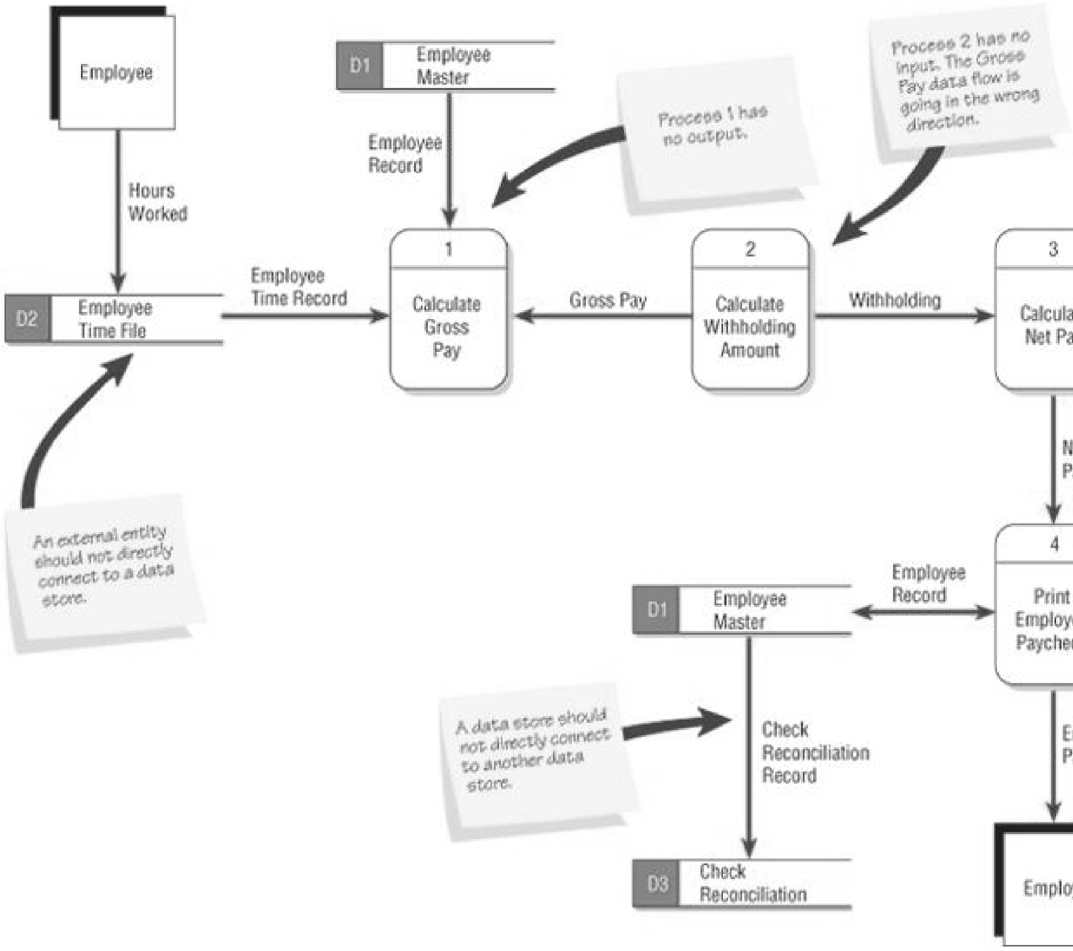


Figure 10. A Data Flow Diagram with Syntactic Errors

Table 1. Criterion and Utility Mappings from Data Flow Diagram

Index	DFD	$\mathbf{X}_i$	a_i	Remarks
1	Employee	1	0.40	-Employee PayCheck + Hours Worked = -1 + 0.6 Designer was 60% (0.6) certain about the egress data flow, 'Hours Worked'.
2	D1	1	0.35	-Employee Record + Employee Record + Check Reconciliation Record = -1 + 1 + 0.35
3	D2	1	0.40	-Hours Worked + Employee Time Record = -0.6 + 1
4	D3	1	**-0.35**	-Check Reconciliation Record + 0 = -0.35
12	Process 4	1	**-0.80**	-Employee Record – Net Pay + Employee Paycheck = -1 – 0.8 + 1 = -0.8
23	P1	1	**-2.60**	-Employee Time Record – Employee Record – Gross Pay = -1 – 1 – 0.6 = -2.6
??	P2	**0**	0.80	-0 +Withholding + Gross Pay = 0.2 + 0.6
???	P3	**0**	0.60	-Withholding + Net Pay = -0.2 + 0.8

Notice there are a number of indications for ambiguities in the design, as expressed in bold types in Table 1. Relative weights, a_4, a_{12}, and a_{23} were out of range. In addition, $\mathbf{X}_{ij\ldots n}$ for Process 2 and 3, respectively, were set to zero despite their presence in the DFD. Their order of interaction was unidentified.

Since the number of intervals for each criterion, ξ_i =1 for every i, there is no need for vertical normalization (with regard to the utilities) for Class V. In fact, zero relative weight for a criterion is considered as having reached the equilibrium. Hence the evaluation function for the verification of a DFD is to seek the following equation, equivalent to (7), to become zero, such that

$$\textit{Minimize} \quad 0 \le \mathrm{CF}_{\alpha} = \sum_{i=1}^{n} a_i^* \cdot \mathbf{X}_i - \sum_{\forall i}\sum_{\forall j} a_{ij} \cdot \mathbf{X}_{ij} + \ldots (-1)^{N-1} \sum_{\forall i}\sum_{\forall j} \ldots \sum_{\forall N} a_{ij\ldots n}^* \cdot \mathbf{X}_{ij\ldots n}, \ \le 1 \qquad (9)$$

where

$$a_i^* = \frac{a_i^{B^V}}{\sum_{\forall i} a_i^{B^V} + \sum_{\forall N} a_{ij..n}^{B^V}} \qquad (10)$$

and

$$a_{ij\ldots n}^* = \frac{a_{ij..n}^{B^V}}{\sum_{\forall i} a_i^{B^V} + \sum_{\forall N} a_{ij..n}^{B^V}} \qquad (11)$$

Equations (10) and (11) allow for horizontal normalization, as illustrated in section III-2.

By using equations (10) and (11), the denominator for horizontal normalization yields

$$\sum_{\forall i} a_i^{B^V} + \sum_{\forall i,j,\ldots,n} a_{ij\ldots n}^{B^V} = (0.40 + \ldots + 0.35) + (0.8 + \ldots + 0.6) = 6.3$$

Thus, (9) above generates the evaluation function for the example DFD as follows.

$$\therefore\ CF_1 = \frac{0.40}{6.30}(1) + \frac{0.35}{6.30}(1) + \frac{0.40}{6.30}(1) + \frac{(-0.35)}{6.30}(1)$$

$$- \frac{(-0.80)}{6.30}(1) - \frac{(-2.6)}{6.30}(1) \qquad : \text{ P2 and P3 unknown}$$

$$+ 2 \cdot [0 \cdot (1)...] \qquad : \text{ Unknown interactions}$$

$$= 0.063 + 0.055 + 0.063 - 0.055 + 0.127 + 0.413$$

$$\cong \mathbf{0.666}$$

Not only do some a_i and $a_{ij...n}$ violate the range constraint, but the evaluation function value *per se* is far from reaching the equilibrium, zero. Notice each DFD must be sequentially verified, one at a time, and semi- to full-automation of the entire procedure may be highly attractive.

IV-2. Design Validation with ISDM

The Data Flow Diagram (DFD) serves as the logical blueprint for an Information System Design (ISD), and is perhaps one of the most critical outcomes in the 'Systems Analysis and Design Lifecycle' as illustrated in Figures 1 and 2 of section II.

Validation of a DFD requires the designer to check for its integrity without the presence of an objective metric system, thereby producing inconsistency, redundancy, and/or at times fatal logical flaws for the information system during implementation. One simply relies on his/her subject-matter expertise, experience, and rules of thumb to gauge the effectiveness of the DFD. In short, human decisions are inaccurate, may be highly subjective, cost inefficient, and most importantly, extremely time consuming. One of the possible solutions to this set of problems is to introduce 'quantification', which in turn may allow automated decision making.

Consider a DFD in Figure 11 [Kendall *et al.* (2005)], which has been validated by a human expert. With insights from 'graph theory applications to systems design' (Willow, 1999) and 'network theory' from Operations Research (Taha, 1992), quantification of DFD (for validation) may begin. Notice the 'entities', 'data stores', and 'processes' map to criteria, x_i, in contrast to DFD verification of section IV-1. The 'data flows' correspond to criterion interactions, $x_{ij...n}$. The criterion values represent sigmoid behavior, as illustrated in section IV-1:

$$\mathbf{X}_{\boldsymbol{i}} = x_i = \begin{cases} 1, & \text{if it exists in DFD} \\ 0, & \text{otherwise} \end{cases} \quad \text{for every } i = 1, \ldots, n. \qquad (12)$$

$$\mathbf{X}_{\boldsymbol{ij...n}} = x_{ij...n} = \begin{cases} 1, & \text{if it exists in DFD} \\ 0, & \text{otherwise} \end{cases} \quad \text{for every } i = 1, \ldots, n. \qquad (13)$$

Hence, an adjacency matrix for the utilities of 'data flows' may be formed which follows in Table 2.

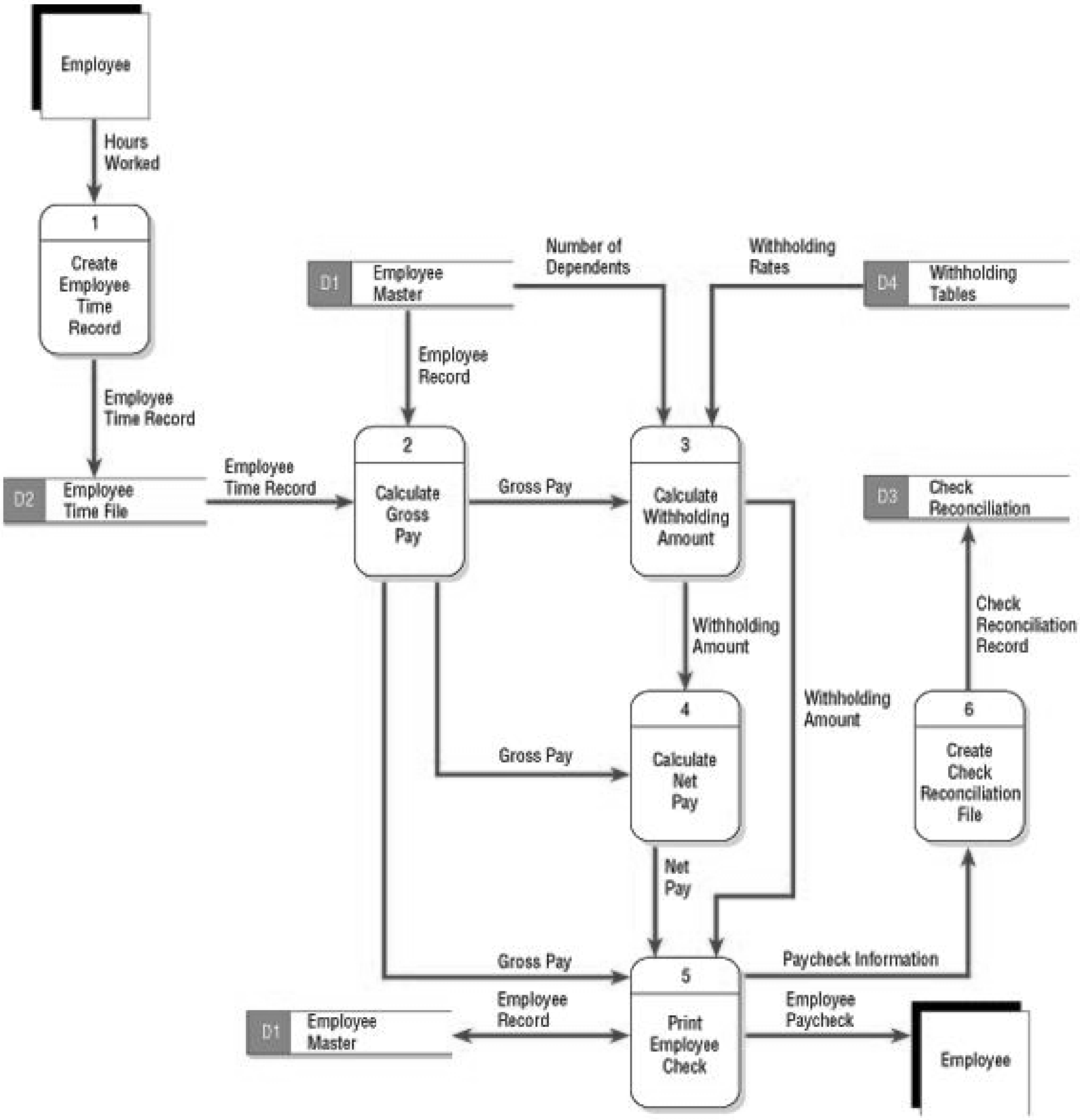


Figure 11. A Validated Data Flow Diagram

Table 2. Utility Adjacency Matrix for Data Flows (DFD Validation)

From \ To	x_1 (Empl)	x_2 (P1)	x_3 (D1)	x_4 (D2)	x_5 (P2)	x_6 (P3)	x_7 (P4)	x_8 (P5)	x_9 (P6)	x_{10} (D3)	x_{11} (D4)
x_1		0.62						X			
x_2	X			0.5							
x_3					0.9	0.82		0.9			
x_4		X			0.5						
x_5			X	X		0.73	0.73	0.73			
x_6			X		X		0.42	0.42			X
x_7					X	X		0.53			
x_8	0.9		0.9		X	X	X		0.4		
x_9								X		0.33	
x_{10}									X		
x_{11}						0.5					

Note most DFDs carry data flows which are of the first-order criterion interactions. That is, up to x_{ij}, where $i = 1, \ldots, n$ and $i \neq j$, are observed in Figure 11. The possible number of first-order interactions for a uni-directional graph is $[n(n-1) / 2] = [11(10) / 2] = 55$ and $n(n-1) = 110$ for a bi-directional in this example. There were 17 data flows altogether, as listed in Table 2, where an 'X' denotes the data flows in reverse directions. Their sum was

$$\sum_{i \neq j}^{11} a_{ij} = 0.62 + 0.5 + 0.9 + \ldots + 0.33 + 0.5 = 10.83 \qquad (14)$$

These weight values in Table 2 were obtained from a pool of ISD experts, upon completion of a short survey which requests a confidence value in range of 0.00 to 1.00 for each data flow when asked to validate the DFD. Similarly, relative weight values for each criterion follow in Table 3.

Table 3. Relative Weights for Criteria (DFD Validation)

Index	DFD	$\mathbf{X}_i$	a_i
1	Employee	1	1.00
2	Process 1	1	1.00
3	Data Store 1	1	1.00
4	D2	1	0.90
5	P2	1	1.00
6	P3	1	0.80
7	P4	1	1.00
8	P5	1	1.00

9	P6	1	0.70
10	D3	1	0.70
11	D4	1	0.90

The sum of the criterion weights are

$$\sum_{i=1}^{11} a_i \;=\; 1.00 + \ldots + 0.70 + 0.90 \;=\; 10.00 \tag{15}$$

Hence, evaluation function for validating a DFD by incorporating ISDM is to

$$MAXIMIZE \quad 0 \le \mathrm{CF}_\alpha \;=\; \sum_{i=1}^{n} a_i^* \cdot \mathbf{X}_i - \sum_{\forall i}\sum_{\forall j} a_{ij} \cdot \mathbf{X}_{ij} + \ldots (-1)^{N-1} \sum_{\forall i}\sum_{\forall j} \ldots \sum_{\forall N} a_{ij\ldots n}^* \cdot \mathbf{X}_{ij\ldots n}, \;\le 1 \tag{16}$$

where

$$a_i^* \;=\; \frac{a_i^{B^V}}{\sum_{\forall i} a_i^{B^V} + \sum_{\forall N} a_{ij..n}^{B^V}} \tag{17}$$

and

$$a_{ij\ldots n}^* \;=\; \frac{a_{ij..n}^{B^V}}{\sum_{\forall i} a_i^{B^V} + \sum_{\forall N} a_{ij..n}^{B^V}} \tag{18}$$

Again, (17) and (18) horizontally normalize all utilities associated with a DFD. Tables 4 and 5 follow to illustrate the *horizontally normalized* utilities. Notice the denominator for both (17) and (18) is the sum of equations (14) and (15), which is 10.83 + 10.00 = 20.83.

Table 4. Horizontally Normalized Utilities for Criteria (DFD Validation)

Index	DFD	a_i	a_i^*
1	Employee	1.00	1.00 / 20.83 = 0.048
2	Process 1	1.00	0.048
3	Data Store 1	1.00	0.048
4	D2	0.90	0.043
5	P2	1.00	0.048
6	P3	0.80	0.038
7	P4	1.00	0.048
8	P5	1.00	0.048
9	P6	0.70	0.033
10	D3	0.70	0.033
11	D4	0.90	0.043

Table 5. Horizontally Normalized Utilities for Criterion Interactions (DFD Validation)

Index	a_{ij}	a_{ij}^*	Index	a_{ij}	a_{ij}^*
12	0.62	0.62 / 20.83 = 0.029	67	0.42	0.020
24	0.50	0.024	68	0.42	0.020

35	0.90	0.043	78	0.53	0.025
36	0.82	0.039	81	0.90	0.043
38	0.90	0.043	83	0.90	0.043
45	0.50	0.024	89	0.40	0.019
56	0.73	0.035	9, 10	0.33	0.016
57	0.73	0.035	11, 6	0.5	0.024
58	0.73	0.035			

The evaluation function thus becomes

$$\begin{aligned} CF_\alpha = {} & a_1\mathbf{X}_1 + \ldots + a_{11}\mathbf{X}_{11} \\ & - [a_{12}\mathbf{X}_{12} - a_{21}\mathbf{X}_{21}] - \ldots. \\ & - [a_{35}\mathbf{X}_{35} - a_{53}\mathbf{X}_{53}] - [a_{36}\mathbf{X}_{36} - a_{63}\mathbf{X}_{63}] - [a_{38}\mathbf{X}_{38} - a_{83}\mathbf{X}_{83}] - \ldots. \\ & - [a_{11,6}\,\mathbf{X}_{11,6} - a_{6,11}\,\mathbf{X}_{6,11}] \end{aligned}$$

$$\begin{aligned} = {} & 0.048\,(1) + \ldots + 0.043\,(1) \\ & - [0.029\,(1) - 0] - \ldots \\ & - [0.043\,(1) - 0] - [0.039\,(1) - 0] - [0.043\,(1) - 0.043\,(1)] - \ldots. \\ & - [0.024\,(1) - 0] \end{aligned}$$

$$= 0.478 - 0.431 \cong \mathbf{0.047}$$

This indicates that there are ample opportunities for the system designers to enhance the DFD in Figure 11 in terms of its completeness and integrity, prior to ‘System Implementation’. The evaluation function value at this point is deemed too low for a critical ISD such as the ‘payroll’. The designer may choose to revise the DFD, seek applications of previously developed templates, or to rebuild it from scratch.

One may argue that the utility values may vary widely between different pools of ISD assessors, which indeed might largely affect the evaluation function. However, this problem of ‘domain dependence’ has plagued almost every application of information system, ranging from databases to embedded systems.

If at all possible and feasible, there should be a library of lookup tables encompassing acceptable CF_α ranges for different classes of problems.

V. CONCLUSIONS

An introduction to the Integrated Systems Design Methodology (ISDM)© and its application to Information Systems Design (ISD) have been detailed in this paper. As an alternative to the Systems Analysis and Design (SAD), ISDM has proven to provide the designer with the following major benefits:

- Evaluation Function: CF_α as the evaluation function assists the designer by allowing the assessments of ISDs on a more objective scale and scope.

- Balance of qualitative and quantitative decisions: ISDM is perhaps one of the first of its kinds to seek and experiment *quantification* of ISDs.

- *A priori* Verification and Validation (V&V): V&V of DFDs prior to ‘Systems Implementation’ may be achieved, saving valuable time for the entire process of systems design. This is expected to increase both the overall efficiency and effectiveness of the design process considerably.

Building ISD itself may involve a series of intricate if not elusive processes, which may never be automated. However, with the increasing demand for timely development for a wide range of information systems, there should be further studies as well as extensions to the research performed in this paper to balance the qualitative and quantitative decisions made in ISD through model generations, algorithm developments, and statistical analyses, among others. In addition, CF_α range analyses for classes of disparate problem structure to validate various DFDs may be an interesting topic for research.

ACKNOWLEDGMENTS

This paper was funded by the 2005 Business Council Research Grant from the School of Business Administration, the Monmouth University, West Long Branch, NJ 07764-1898, U. S. A.
The author wishes to extend his gratification to the anonymous reviewers for their time, assistance, and patience.
Most importantly, this paper would not have been completed without the cooperation from my precocious son, Ocean D. Willow…

REFERENCES

Aukerman, R., Schooley, R., Nord, D., & Nord, J. (1989). The Importance of Selected Systems Analysis and Design Tools and Techniques as Determined by Industry Systems Analysts and University Educators. ACM SIGCSE Bulletin, 21(3), 30-34.

Avrunin, G. S., & Wileden, J. C. (1985). Describing and Analyzing Distributed Software System Designs. ACM Transactions on Programming Languages and Systems, 7(3), 380-403.

Baskerville, R. (1993). Information Systems Security Design Methods: Implications for Information Systems Development. ACM Computing Surveys, 25(4), 375-414.

Botting, R. J. (1986). Into the Fourth Dimension: An Introduction to Dynamic Analysis and Design. ACM SIGSOFT Software Engineering Notes, 11(2), 36-48.

Casella, G., & Berger, R. L. (1990). Statistical Inference. Wadsworth, Berkeley, California.

Cheung, S. C., & Kramer, J. (1994). An Integrated Method for Effective Behavior Analysis of Distributed Systems. Proceedings of the 16th IEEE International Conference on Software Engineering, 309-320.

Chung, J. W. (1994). Utility and Production Functions: Theory and Applications. Blackwell, Oxford, U.K., 1994.

Coad, P., & Yourdon, E. (1991a). Object-Oriented Analysis. Yourdon Press, Englewood Cliffs, New Jersey.

Coad, P., & Yourdon, E. (1991b). Object-Oriented Design. Yourdon Press, Englewood Cliffs, New Jersey.

Davis, W. S. (1983). Systems Analysis and Design: A Structured Approach. Addison-Wesley.

Felder, M., & Pezzè, M. (2002). A Formal Design Notation for Real-Time Systems. ACM Transactions on Software Engineering and Methodology, 11(2), 149-190.

Fenelon, P., McDermid, J. A., Nicholson, M., & Pumfrey, D. J. (1994). Towards Integrated Safety Analysis and Design. ACM SIGAPP Applied Computing Review, 2(1), 21-32.

Fritz, J. M. (1987). A Pragmatic Approach to Systems Analysis and Design. ACM SIGCSE Bulletin, 19(1), 127-131.

Gabbert, P. (2001). Systems Analysis: The Challenge of Integrating Two Competing Technologies. Journal of Computing Sciences in Colleges. 16(2), 195-202.

Go, K., & Carroll, J. M. (2004). The Blind Men and the Elephant: Views of Scenario-Based System Design. ACM Interactions, 11(6), 45-53.

Gray, L. (1988). Transitioning from Structured Analysis to Object-Oriented Design. Proceedings of 5th ACM Washington Symposium on ADA, 151-162.

Hahn, J., & Kim, J. (1999). Why are some Diagrams Easier to Work with? Effects of Diagrammatic Representation on the Cognitive Integration Process of Systems Analysis and Design. ACM Transactions on Computer-Human Interaction, 6(3), 181-213.

Hellestrand, G. R. (1999). The Revolution in Systems Engineering. IEEE Spectrum, 36(9), 43-51.

Hellestrand, G. R. (2005a). The Engineering of Supersystems. IEEE Computer, 38(1), 103-105.

Hellestrand, G. R. (2005b). Systems Architecture: The Empirical Way – Abstract Architectures to 'Optimal' Systems. Proceedings of the 5th ACM International Conference on Embedded Software EMSOFT '05, 147-157.

Kendall, K. E., & Kendall, J. E. (2005). Systems Analysis and Design, 6th Edition, Prentice-Hall.

Marculescu, R., & Nandi, A. (2001). Probabilistic Application Modeling for System-Level Performance Analysis. Proceedings of the IEEE International Conference on Design, Automation, and Test in Europe, 572-579.

Mayer, R. J., Benjamin, P. C., Caraway, B. E., & Painter, M. K. (1995). A Framework and a Suite of Methods for Business Process Reengineering. Knowledge Based Systems, Inc., College Station, Texas.

Mood, A. M., Graybill, F. A., & Boes, D. C. (1974). Introduction to the Theory of Statistics. McGraw-Hill, Singapore.

Ostrofsky, B. (1977). Design, Planning, and Development Methodology. Prentice Hall, Englewood Cliffs, Inc., New Jersey.

Pernici, B., Barbic, F., Fugini, M. G., Maiocchi, R., Rames, J. R., & Rolland, C. (1989). C-TODOS: An Automatic Tool for Office System Conceptual Design. ACM Transactions on Information Systems, 7(4), 378-419.

Rosson, M. B., Maass, S., & Kellogg, W. A. (1987). Designing for Designers: An Analysis of Design Practice in the Real World. Proceedings of the ACM SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface, 17(SI), 137-142.

Roussinov, D., & Chen, H. (2001). Information Navigation on the Web by Clustering and Summarizing Query Results. Information Processing and Management, 37(6), 789-816.

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2002). Systems Analysis and Design in a Changing World. Thomson Learning, Canada.

Schmidt, J. W. (1986). Introduction to Systems Analysis, Modeling, and Simulation. Proceedings of the 18th ACM Winter Simulation Conference, 5-16.

Siddall, J. N. (1982). Optimal Engineering Design: Principles and Applications. Dekker, New York.

Siddall, J. N. (1983). Probabilistic Engineering Design: Principles and Applications. Dekker, New York.

Singh, S., & Kotzé, P. (2003). An Overview of Systems Design and Development Methodologies with regard to the Involvement of Users and Other Stakeholders. Proceedings of the South African International Conference on Systems and Information Technology, 37-47.

Stavely, A. M., Becker, D. C., Orr, S. P., & Titus, G. B. (1985). A Collection of Software Tools for Analyzing Designs of Concurrent Software Systems. Proceedings of the 8th IEEE International Conference on Software Engineering, 111-118.

Suh, N. P. (1990). The Principles of Design. Oxford University Press, New York.

Taha, H. A. (1992). Operations Research, 5th Edition. McMillan, New York.

Turetken, O., Schff, D., Sharda, R., & Ow, T. T. (2004). Supporting Systems Analysis and Design through Fisheye Views. Communications of the ACM, 47(9), 72-77.

Thomas, R. J., & Kirkham, J. A. (1981). Structured Systems Analysis and the Problem Statement Language as a Combined Methodology in the Teaching of Systems Analysis and Design. Proceedings of the ACM '81 Conference, 96-100.

White, K. P. (1998). Systems Design Engineering. Systems Engineering, 1(4), 285-302.

Whitten, L. J., Bentley, D. L., & Dittman, C. K. (2001). Systems Analysis and Design Methods. McGraw-Hill Irwin, Boston, MA.

Wickens, T. D. (1995). The Geometry of Multivariate Statistics. Erlbaum Associates, New Jersey.

Willow, C. (1998). Neuro-Fuzzy Systems in Engineering Design. Doctoral Seminar Report, Department of Decision and Information Sciences, University of Houston, Houston, Texas.

Willow, C. (1998). An Object-Oriented Database for Information Management in Computer Integrated Manufacturing. ACM Transactions on Information Systems, 34(_), 12-39.

Willow, C. (1999). Multiple Criterion Function Methods in Design with Varying Relative Weights. Doctoral Dissertation, Department of Industrial Engineering, University of Houston, Houston, Texas.

Willow, C. (2004). Lecture Notes for MIST-715: Systems Analysis and Design, Dept. of Management Information Systems, New York Institute of Technology, NYIT Press, New York, NY.

Willow, C. (2006 in print). Qualitative Decision Making with Integrated Systems Design Methodology. Journal of Engineering and Technology Management, Manuscript # JET-M MS#2005-03-3738, Elsevier Publication.

Willow, C., & Ostrofsky, B. (1997). Multiple-Criterion Computational Methods in Design. University of Houston Press, Houston, TX.

Wymore, A. W. (1981). Applications of Mathematical System Theory to System Design, Modelling, and Simulation. Proceedings of the 13th IEEE Winter Simulation Conference, 1, 209-219.

Yourdon, E. (1984). Structured Analysis and System Specification Workshop. Yourdon Press, New York.

Yourdon, E. (1993). Yourdon System Method: Model-Driven Systems Development. Prentice-Hall, Englewood Cliffs, New Jersey.

Yue, K. (1989). Validating System Requirements by Functional Decomposition and Dynamic Analysis. Proceedings of the 11th ACM International Conference on Software Engineering, 188-196.