View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2000 Proceedings

International Conference on Information Systems (ICIS)

December 2000

Game Playing Behavior in Requirements Analysis, Evaluation, and System Choice for Enterprise Resource Planning Systems

Daniel O'Leary
University of Southern California

Follow this and additional works at: http://aisel.aisnet.org/icis2000

Recommended Citation
O'Leary, Daniel, "Game Playing Behavior in Requirements Analysis, Evaluation, and System Choice for Enterprise Resource Planning Systems" (2000). *ICIS 2000 Proceedings*. 35.
http://aisel.aisnet.org/icis2000/35

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

GAME PLAYING BEHAVIOR IN REQUIREMENTS ANALYSIS, EVALUATION, AND SYSTEM CHOICE FOR ENTERPRISE RESOURCE PLANNING SYSTEMS[1]

Daniel E. O'Leary
Marshall School of Business
University of Southern California
U.S.A.

Abstract

Historically, using legacy software, each major branch of a company could have their own system that met their particular needs. Unlike legacy software, enterprise resource planning (ERP) software requires that the same software be implemented in each branch or office. As a result, now branches must somehow come to agreement on software choices. A number of firms have employed ranking mechanisms where branches or their representatives effectively vote, via their rankings, to determine which software is used. Unfortunately, this can mean the introduction of gaming behavior as branches try to get the software that they think best meets their particular needs. The purpose of this paper is to review some of that gaming behavior and investigate the impact of those behaviors in the ERP requirements analysis process, ERP evaluation process, and with ERP system choice, based in the context of three real world cases.

Keywords: ERP systems, requirements analysis, game playing behavior, system evaluation, system choice

1. INTRODUCTION

This paper investigates the game playing behavior that can manifest itself in the requirements analysis process, evaluation process, and choice of enterprise resource planning (ERP) software, based on three real world cases and two sets of requirements analysis software.

Historically, since computing has been done locally, different branches of firms have been free to choose the software that best met their needs. As a result, legacy software was chosen to conform to local requirements. However, with global competition, and the ability to link together different branches, firms have increasingly turned to enterprise resource planning systems to link the branches together. Unlike legacy software that could be chosen to meet local needs, ERP software ultimately requires potentially disparate branches or offices of the same firm to choose the same software. Since some software differentially impacts each branch, branches have incentives to execute behavior that can result in the choice of software that they think best fits their individual branch needs. Accordingly, in contrast to legacy systems, at some point, different branches and offices must come to agreement in their choice of an ERP system. Getting agreement may take a dictatorial central choice or branches or companies may vote on it. For example, recently some diversified firms that contain thousands of decentralized companies announced that SAP is the preferred software choice. Alternatively, other firms such as Chesapeake, discussed below, apparently vote on different choices.

The findings in this paper are important since they provide insights as to the types of game playing behavior that can take place in ERP selection. These insights can be critical for project management and possible mitigation of their impact.

[1]The author would like to thank the referees and the associate editor for their comments on an earlier version of this paper.

Although perhaps the best known ERP software is SAP, within ERP software there are the "big five" vendors, that include BAAN, J. D. Edwards, PeopleSoft, Oracle, and SAP. Different ERP systems are seen has having different strengths and weaknesses, thus appealing to different groups differentially.

Typically, ERP software includes purchasing, production, financial accounting, management accounting, materials handling, treasury functions, and a wide range of other activities. ERP software is process-based, rather than function-based. Further, ERP systems integrate those processes across different branches or offices of the company. Accordingly, much of the benefit of ERP systems is in their ability to integrate a firm.

Although ERP systems facilitate integration and provide substantial overall benefit to a particular firm, the individual branches may benefit differentially from different ERP systems, or from an ERP altogether. As a result, in some settings, firms may find themselves divided between different branches as to which ERP system they should chose.

2. SAMPLE FIRM CASES[2]

In this paper, three cases are used to illustrate the gaming concepts inherent in choosing between three different ERP systems: Chesapeake Display & Packaging, Financial Services (an alias) and Timberjack. This section provides a general overview of the three companies and their requirements analysis.

2.1 Chesapeake

Chesapeake Corporation (CC) is a multidivisional company, which includes Chesapeake Display and Packaging (CD&P) and Chesapeake Packaging Corporation (CPC). As seen in the companies names, Chesapeake focuses on packaging and displays. Cheimis (1999), Vice President and Chief Information Officer for CD&P, set up a five step process for choosing enterprise software for CC:

1. Form small blue ribbon team
2. Contact vendors to arrange demos
3. Ask the vendor for proof of rapid implementation capability
4. Vote
5. Make the choice based on the vote.

The group of voters spanned each of the two participating divisions and Chesapeake Corporate. The voters and the result of the voting are summarized in Table 1.

The software with the highest average vote is chosen. In this case it is clear that the vote yields a choice between BAAN and J. D. Edwards (JDE), and ultimately, J. D. Edwards.

2.3 Financial Services, Inc.[3]

Financial Services (FS) is an alias for a large financial services firm. FS chose and implemented SAP in the mid-1990s. FS ultimately chose between SAP and Oracle. FS had five branches that were to use the same software. FS issued a request for proposals with a list of over 1,200 requirements they had determined were important. Features were assigned a number from 1 to 5 indicating the importance of the feature, where 5 meant that the feature had to be in the software or that the software could be easily modified to accommodate that feature. For each requirement, FS decided to either have it be required for the software (a "deal killer" if the software could not accommodate) or a requirement rated for its importance. A single branch of the company could indicate that lack of a requirement was a deal killer, resulting in selection of alternative software. Beyond, critical, deal killing requirements, the importance of each requirement was rated as high (3), medium (2), or low (1). For each branch impacted by the requirement, consensus was sought by averaging across the branches.

---

[2]The use of these cases does not imply that each game playing behavior reviewed here was actualized in the cases. However, the cases do show that firms potentially face the entire range of behaviors discussed here.

[3]Unfortunately, I am not able to disclose the name of the firm.

Table 1. Software Preferences at Chesapeake

		Best Functional Fit			Best Implementation Personnel		
Who	Division/Position	BAAN	JDE	SSA	BAAN	JDE	SSA
Gary Cheimis	CD&P VP CIO	2	1	2	3	1	
Elvis Brannam	CD&P IS Mgr	2	3	1	3	2	1
Linda Witter	CD&P Sales Support	3	2	1	3	2	1
Ted Samoits	CD&P Production Mgr	2	3	1	2	3	1
Jack Kirk	CD&P VP CFO	2	3	1	2	3	1
John Polgar	CD&P Sr. Analyst	2	3	1	2	3	1
Carl Wilcox	CD&P Design Adm	3	2	1	3	2	1
Richard Hastings	CD&P Oper Mgr	3	2	1	3	2	1
Mary Gene Simmions	CC Dir Shrd Svc	2	3	1	2	3	1
Dick Fuss	CC Analyst	3	3	1	2	3	1
Randy Grahm	CC Internal Audit	1	3	2	1	3	2
Bill Tolley	CC CFO	2	3	1	2	3	1
David Spencer	CPC Engineering Mgr	2	3	1	2	3	1
Ann Walsh	CPC IS Mgr	2	3	1	2	3	1

2.3 Timberjack

Timberjack is a company that was faced with the choice of the same ERP system in two locations, one in the United States (USA) and one in Sweden (Romanov 1998). Each of the two offices has substantially different needs, and the software differentially meets the needs of the two branches. Timberjack first issued a request for proposals with a list of over 1,000 requirements. Timberjack developed over 1,000 individual requirements for modules including sales order processing, a dealer system, sales analysis, inventory management, warehousing, accounting, and master file data. Requirements were categorized as either "must," "comply," "enhance," "custom," and "N/A." After receiving the proposals, Timberjack created a short list, where the USA branch was given Oracle and Computer and Associates (CA) to evaluate, whereas the Swedish branch was given the Lawson and QAD software to evaluate. Ultimately, the USA branch chose Oracle and the Swedish branch chose QAD, and then the two branches had to make the final choice. The Timberjack case provides an excellent illustration of some of the most critical issues that can be involved in multiple branch ERP system choice.

3. REQUIREMENTS ANALYSIS FOR ERP SYSTEMS

Requirements analysis for ERP can range from a small, general set of requirements to a long list of requirements, detailing specific data capabilities, e.g., whether the human resources data captures country of citizenship (Financial Services, Inc.). It can include what subsystems are supposed to interact with what other subsystems and a range of other issues. Given the set of requirements, each is then rated for its importance. This rating approach is well-known. In particular, there are firms that specialize in software that can be used to generate the importance ranking of different requirements and compare the requirements to existing software (e.g., Excelo Corporation and The Requirements Analyst, discussed below). However, integrating the views of multiple branches has not been addressed by these developers. Typically, it is assumed that firms can specify the priority or importance of each requirement, *a priori*.

3.1 Excelo Corporation

Excelo Corporation (2000, p. 1), provides a questionnaire that can be used to help select software. The questionnaire has 1,850 questions that are scored between 1 and 9, in order to facilitate the ranking and selection process. As noted in their documentation,

> If a feature, function, or report is of interest, indicate the degree of interest with a weighting factor from "1" indicating a low and "9" (critical) in the position provided. Remember to use 9's sparingly as TAL [name of

> the questionnaire] rejects all systems that do not have the feature covered by the 9 you are using. We suggest that you weight features you really need with 8's and reduce the weighting of all other less significant questions.

As a result, use of the extreme value "9" could result in disqualification of some software from consideration.

3.2 The Requirements Analyst

The Requirements Analyst—TRA (http://www.ctsguides.com/accounting/index.html.) also provides a questionnaire that can be used to help select software. TRA also has more than a 1,000 requirements to choose from. Unlike the questionnaire available from the Excelo Corporation, it has an easier to use user interface. Further, rather than ranking from 1 to 9, TRA uses rankings from 0 to 3. As an example, consider the illustration given in Table 2.

Table 2. Requirements Rated by Importance

Job Cost	
General Features	
Integration with ...	
Accounts payable	2
General ledger	2
Payroll	3
Accounts Receivable	3
Progress Billing	
Time and Materials	
Inventory	1
Equipment	1
Purchasing Orders	
Service Management	
Estimating	
Scheduling	
Industry Standard Codes	
Detailed Transaction	

4. ERP REQUIREMENT ANALYSIS GAME BEHAVIOR

Perhaps the most critical ERP requirements analysis game is the "deal killer game," where a branch can kill adoption of a particular ERP system through requiring a particular requirement available in some but not all systems. In addition, there is a "requirement averaging game" where branches use their ratings on requirements to influence the importance of a requirement. In this section, we assume that each division, (branch or individual) does a requirements analysis and that the individual analyses are then aggregated, in some way.

4.1 Deal Killer Requirements

Assume that the requirements are analyzed so that, at a first cut, all those packages that have some deal breaker in them from some branch are eliminated from consideration. As we saw with the software from Excelo, if a single requirement is put in as a 9 and the software does not have that requirement, then that software will be eliminated from consideration. Further, as seen with FS, companies do build deal killer capabilities in their software selection processes at the branch level.

Assume that the requirements are established as the result of a joint effort of two branches. If one branch is more interested in implementing software package A as opposed to C, potentially it can influence the choice by changing its evaluation of a requirement. For example, if branch 1 wants software package A rather than C, then it can facilitate that choice by finding some requirement B-j (j = 1, 2) that is in A, but not C, and label it as a necessary requirement. The inability of C to meet the requirement would then be viewed as a "deal killer."

In the game layout in Figure 1, the tuples capture, respectively, the actions of the first and second branches whether the attribute is actually a requirement or not, for the respective office. U indicates that the branch (office) j falsely indicates that requirement B-j is a deal killer, when it is not. F indicates that the "fair" strategy is taken, not indicating the requirement is a deal killer, when it is not. Branches (individuals, players) would have utility payoff's in blocks 1 and 4.

	U	F	
Office 1	(B-1 Required, B-2 Required) 1	2 (B-1 Required, B-2 Not Required)	U
	3 (B-1 Not Required, B-2 Required)	4 (B-1 Not Required, B-2 Not Required)	F
	Office 2		

Figure 1. "Deal Killer" Strategies Game

Conceivably, both branches could find requirements that would be deal killers, ultimately resulting in the choice of a third package D that meets all the "deal killer" needs, but is not the preferred choice of either branch 1 or branch 2.

Consider the payoff matrix in Figure 2. As an example, if the utilities were such that the payoffs became, p = 0, q = 6, t = -2 and r = 4, then the payoff matrix would represent a prisoner's dilemma. The Nash solution for this game is where both offices choose U as a strategy. As noted by Morrow (1994) communication alone cannot solve the dilemma. Instead, the players need a binding agreement or some way to enforce a deal. If players pursue their own goals they will end up with an inferior solution. Branches must cooperate. Accordingly, an ERP system choice should have to account for this finding as part of the project management.

	U	F	
Office 1	(p,p) 1	2 (q,t)	U
	3 (t,q)	4 (r,r)	F
	Office 2		

Figure 2. Payoff Matrix for Deal Killer Strategy

The situation where both branches play the deal killer strategy can also be viewed as a "crossing game" (Moulin 1986). In the crossing game, two motorists approach an intersection simultaneously. If both stop (F,F), then both get a payoff of say p = 10. If one stops and the other does not, then say t = 20 and q = 9, since their pride may be injured. If neither stop, then the strategy pair (U,U) results in an accident with a cost of say r = -20. Similarly, if one branch creates a false deal killer but the other does not, then t = 20 and q = 9. If both create a false deal killer, then the cost may be another package that is worse for both, resulting in a cost of p = -20. Finally, if neither use the false deal killer, then both go with the consequences and get a benefit of r = 10. The game can be generalized by moving from a general p to p_1 and p_2, where the two are not equal, and the other payoffs are also generalized.

4.2 Requirement Averaging Game

Next assume that requirements for individuals, or branches, are gathered and averaged, as was done at Chesapeake. In this situation, a branch can either over or under represent the weight on a requirement in order to push the requirements analysis in the direction of its preference. For example, suppose an office (branch) O_1 has a preference for one ERP package over another, say A over C. However, suppose that there exists some requirement that is in C, but not in A. In this case, O_1 can rank the requirement as not important. Then when the branch rankings are averaged, the importance will be minimized. A similar situation occurs with evaluation weights and is discussed further in the next two sections.

5. EVALUATING ERP SYSTEMS

Sometimes after requirements analysis, addition evaluation is done. Requirements analysis would be used to trim the number of options, and then evaluation would be used to account for factors other than functionality. As noted by Anderson Consulting, (Flaatten et al. 1992, pp. 260-261) a common way of evaluating software is to use different "scoring methods, where criteria are assigned weights and ranges, and the evaluation process consists in assigning values to the criteria by consensus of the evaluators." Then, the software with the highest total weight is chosen. Both Timberjack and FS ultimately used evaluation criteria and weights as part of the evaluation process.

5.1 Timberjack

In order to illustrate the use of evaluation criteria and weights on ERP systems, Table 3 summarizes a set of evaluation criteria and weights used in Timberjack's comparative evaluation of QAD and Oracle (Romanov 1998, p. 21). In that case, Timberjack's offices in Sweden and the USA had to choose a single system. Ultimately, the choice was between QAD and Oracle. Timberjack employed the evaluation criteria and weights to facilitate their choice process. A representative from Coopers & Lybrand, the large consulting firm, helped them with their analysis. Using the numerical approach, they had to achieve consensus with choice of a single system.

Table 3. Evaluation Criteria and Weights

Factor	Weight	QAD	Oracle
Support	20		
Functionality	30		
User Interface	10		
Flexibility	20		
Future Prospects	10		
Reliability	20		
Integration	30		
Platform	20		
Total	160		

5.2 Financial Services

Evaluation at FS was very similar to that at Timberjack. As part of the choice process for their ERP software, FS developed evaluation criteria and allocated points in order to help the firm decide which ERP system would best fit their needs. Unlike Timberjack, FS was to choose between SAP and Oracle, rather than QAD and Oracle. In addition, their choice criteria included some different categories, e.g., maintenance requirements. Maintenance requirements are those requirements designed to facilitate system maintenance after implementation. Like Timberjack, FS worked with a large consulting firm. In addition, one of the project leaders had previous experience with a large consulting firm. Exploiting its consulting background, FS was very systematic and forward looking in its ERP choice and implementation processes. Consensus on the choice of the weights was attained by averaging the weights of the multiple branches.

6. ERP EVALUATION WEIGHT GAMES

ERP evaluation weight games depend, in part, on the approach used to generate weights for criteria or choose the criteria for which weights will be assigned. For example, as part of the development of the ERP evaluation and choice analysis, FS decided that they would averaged the weight criteria of the different branches. Thus, each branch was asked to assign 100 points to a set of evaluation criteria.

6.1 Game Assumptions

It is assumed that the "fair" solution is the allocation of points that is optimal for the firm, not the branches. It is assumed that each branch knows the fair solution. It also is assumed that there are two branches and a set number of points are being distributed across two categories. Although the results can be extended to multiple players and categories, the two office and two category allocation more easily illustrates the game playing behavior of different branches.

It is assumed that either a branch plays "fair" (F) by using the fair solution, or uses an "unfair" (U) strategy to have it gain maximal self benefit. Branches that play unfair strategies are assumed to use an "extreme point" assignment of weighting points to maximize the impact of their assessment. For example, if there are x points and the branch will be unfair, then it will assign x points to one criteria and 0 to the other. Criteria 1 is assumed to be more beneficial for branch 1, while criteria 2 is assumed to be more beneficial for branch 2.

In the presentation of the weights in game form, the top line will represent the evaluation point distribution by office 1 and the second line will be the evaluation point distribution by office 2. The third line is the average of the two distributions that is used to generate the weight.

6.2 Fair Solution Is Equal

First, consider a two branch firm, where the "fair" solution is for each of the two branches to equally distribute evaluation criteria points among two different criteria. We find that if both departments choose either the fair or unfair allocation, the same solution will be arrived at. Differences occur only if one chooses the fair allocation and the other does not. If both take the U strategy, each branch's extreme point assessment of self interest is off-set by the other branch. An example game is given in Figure 3, where two offices must distribute 20 points to two different criteria.

	U	F	
Office 1	20,0 0,20 (10,10)	20,0 10,10 (15,5)	U
	10,10 0,20 (5,15)	10,10 10,10 (10,10)	F

Office 2

Figure 3. Equal Weight Values

6.3 Fair Weights Are Not Equal

If the fair weights are not equally distributed between two options, then the greatest distance from the fair solution is not the case where neither submits the fair solution, but is a setting where one of the offices presents a fair solution and the other does not. Further, the "distance" of the average solution from the fair solution is asymmetric, closer when one plays U than when the other plays U. An example is given in Figure 4 where two offices must distribute points to two different criteria.

	U	F	
Office 1	20,0 0,20 (10,10)	20,0 15,5 (17.5,2.5)	U
	15,5 0,20 (7.5,12.5)	15,5 15,5 (15,5)	F

Office 2

Figure 4. Unequal Weight Values

In general, assume each of two offices have K > 0 points to distribute and those points will be added and averaged. Also assume that the optimal allocation for the firm is K ((n-1)/n) and K/n, $n \geq 1$. For example, in Figure 3, K = 20 and n = 2, while in Figure 4, K = 20 and n = 4. In general, for $n \geq 3$, in quadrant 2, where O_2 is F and O_1 is U, the payoffs will be closer to the F, F strategy than if both offices play U. Further, for $1 \leq n < 1.5$, in quadrant 3, if O_1 is F and O_2 is U, then the average payoffs are closer to the F,F strategy than if both offices play U. Thus, in these cases, if one office knows that another office will execute the U strategy, then by executing an F strategy, the two can generate an averaged solution that is closer to an F strategy than if the office executed a U strategy.

6.4 What Can the Company Do?

In the case where the divisions are assumed to "know" the fair weights, the company has made known the "solution" that it thinks is appropriate. If individual divisions break from the ranks, then that will be known. Accordingly, management could structure additional incentives that might guide the division to making the decision it prefers. Alternatively, the company could just execute the strategy it prefers without gathering office estimates (e.g., CIO 1996).

7. SYSTEM CHOICE THROUGH AGENDA SETTING

In some firms, e.g., Timberjack, the choice of an ERP system has been done using "decentralized paired comparisons of systems," not unlike winner take-all sports tournaments designed to choose a "champion." In particular, in these decentralized paired comparisons, the ERP choice process is broken into pieces where one branch chooses between one pair of ERP systems and another branch chooses between another pair of ERP systems. Then the two branches each bring the "winning" ERP system from their branch comparison for further competition as to which is preferred for the company overall.

The decentralized paired comparison approach was used by Timberjack (Romanov 1998). In that set of comparisons, the Swedish branch compared QAD and Lawson, while the USA branch compared Oracle and CA, as illustrated in Figure 5.

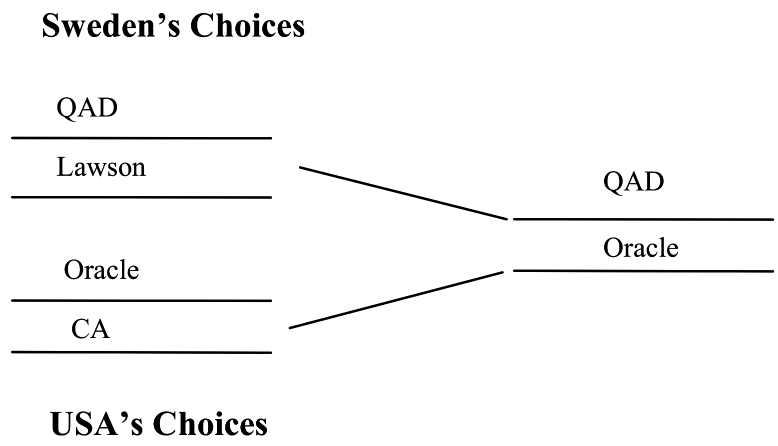

Figure 5. Timberjack's ERP Choice Process

In these kinds of branch comparisons of ERP systems, the order in which system choice is structured can play an important outcome in which ERP system ultimately is chosen by the firm. As a result, the "agenda" (order and pairs of choices) that are set can influence which ERP system is chosen.

Indicate a preference for system A over system B as A → B. Suppose that Oracle → CA and QAD → Lawson, but that Lawson → Oracle and CA → QAD. If Oracle and CA are paired, and QAD and Lawson are paired then we will have the same result summarized in Figure 4. However, if the agenda is changed, the results can change substantially. If instead the systems are paired for comparison as (Oracle, Lawson) and (QAD, CA) rather than (Oracle, CA) and (QAD and Lawson), then the choice will be substantially different, summarized in Figure 6. Agenda pairing makes a difference in the ERP software that can be chosen.

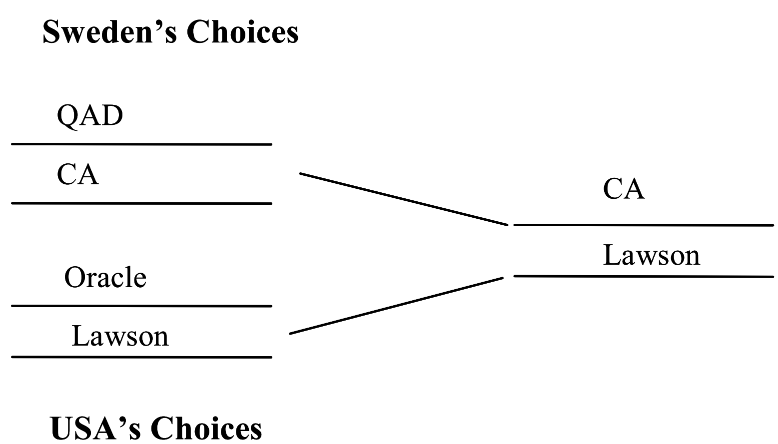

Figure 6. Alternative Possible Agenda

In the case where the preference between software choices is one of "strict dominance," this approach can result in the "iterated elimination of strictly dominated strategies" (Gibbons 1992).

8. STACKING THE DECK

In those settings where votes are gathered from a number of voters, another game can be played: stacking the deck. In this game, the choice of who votes becomes a critical concern. For example, at Chesapeake, there were 14 voters. How do firms decide who becomes a voter? Is there any gaming that stacks the deck in favor of a particular candidate system? This section investigates the possibility of such gaming behavior.

Assume a setting with two divisions, or that the voters can be set in two different groups, A and B. Let |A| be the cardinality of A. Assume that there are two choices. Assume that within group voting for group A is cohesive and all members vote the same. Assume that for group B, the voting is split. One approach to getting the solution desired by group A or the B minority is to either increase the number of voters in A or the minority in group B, or both, so that the numbers exceed those of the group B majority. In general,

$$|A| + |B_{Minority}| \geq |B_{Majority}|$$

As an example, in the case of Chesapeake, the two groups could be A = (CC,CPC) and B = (CD&P). Since there was only one vote for SSA, assume that the choice is between JDE and BAAN. Analysis of those two groups reveals that group A voted J.D Edwards as the top choice, while group B was evenly split between BAAN and JDE. $|A| = |A_{JDE}| = 6$, $|B| = 8$, $|B_{JDE}| = |B_{BAAN}| = 4$, which implies JDE would be chosen. In terms of the story of stacking the deck, assume that the decision was originally only for CD&P. By adding voters from CC and CPC, the JDE contingent was able to get the votes they "needed."

9. SUMMARY, CONTRIBUTIONS, AND EXTENSIONS

ERP systems differ from legacy software in that multiple branches must now chose the same software, rather than choosing software to meet local needs. As a result, this paper examined some of the game playing behavior that can take place in a multi-branch firm that is implementing the same ERP software in each location.

9.1 Contributions

This paper makes a number of contributions, each couched in analysis of potential ERP system evaluation and choice, driven by the need to have the same system in multiple branches. First, branches can use game playing behavior to push the software that they prefer in order to meet their specific needs. In particular, "deal killer" requirements can force the choice of software preferred by a particular branch, unless other branches also pursue the same strategy. Second, using strategies designed to influence the averages of requirement priorities or ERP weights, branches can manipulate the ratings in order to push the software that they prefer. Third, agendas can be set by branches in order to facilitate positive evaluation of the software that they prefer. If ERP software is compared on a pairwise basis to other ERP software, then the order and pairing can influence the ultimate ERP system choice.

It is important to determine where and when such game playing might take place. Project management might be used to mitigate such behavior, to the extent possible, by establishing appropriate rules up front.

9.2 Extensions

The results can be extended in a number of directions. First, the focus of this paper has been on two branches, but can be extended to three or more branches (e.g., Rapoport 1970). Second, in the analysis of evaluation weights, this paper focused on two requirements at a time. However, that analysis can be extended to multiple requirements. Third, although in some cases the payoff weights are presented as equal for different branches, that will not always be the case. However, the results can be extended to different weights for different branches.

Perhaps the most important extension is to further formalize the results in a game theoretic structure. For example, future work could investigate how to align conflicting incentives in order to generate the desired behavior and suggest specific evaluation mechanisms that do not have the shortcomings noted in this paper. This discussion is the focus of a subsequent paper.

References[4]

Cheimis, G. "Shooting the Rapids of a Rapid Implementation," *Shape Your World, Focus '99, J. D. Edwards*, June 1999.
CIO. "SAP Key Roles," http://www.cio.com/forums/061596_sap_roles.html, 1996
Excelo Corporation. "Needs Definition and Evaluation Checklists - Instructions," http://www.excelo.com/allquest.htm, printed February 2000.
Flaatten, P., McCubbrey, D., O'Riordan, P., and Burgess, K. *Foundations of Business Systems*, New York: Dryden Press, 1992.
Friedman, J. *Game Theory with Applications to Economics*, New York: Oxford University Press, 1986.
Gibbons, R. *Game Theory for Applied Economists*, Princeton, NJ: Princeton University Press, 1992.
Morrow, J. *Game Theory for Political Economists*, Princeton, NJ: Princeton University Press, 1994.
Moulin, H. *Game Theory for the Social Sciences* (2nd ed.), New York: New York University Press, 1986.
Rapoport, A. *N-Person Game Theory*, Ann Arbor, MI: University of Michigan Press, 1970.
Romanov, D. "Timberjack Parts: Packaged Software Selection," Harvard Business School 9-398-085, 1998.

[4]This paper and the following reference list contain URLs for World Wide Web pages. These links existed as of the date of submission but are not guaranteed to be working thereafter. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced. The author(s) of the Web pages, not ICIS, is (are) responsible for the accuracy of their content. The author(s) of this article, not ICIS, is (are) responsible for the accuracy of the URL and version information.