ISSN: 2473-3857
v3 n4388

Mastering Data Warehouse Maturity Concepts Using a Serious Game: Design and Implementation of Emerge2Maturity

Mohammed Khojah
Mohammed.Khojah@ucdenver.edu

Michael Mannino
Michael.Mannino@ucdenver.edu

Business School
University of Colorado Denver
Denver, CO 80217, USA

Abstract

Data warehouse development is a complex process involving several related factors and extended time periods to reach a stable solution. Learners face challenges to observe changes, determine key success factors, and understand project relationships involving costs and benefits. Benefits of data warehouse deployment are often intangible especially during initial periods of usage. In contrast, costs are tangible and high during data warehouse development especially with uncertain levels of data quality. Learners need to balance benefits and risks as organizations acquire capabilities to support an organization's strategy for data warehouse development. This paper describes the design and implementation of Emerge2Maturity, an innovative serious game that addresses these learning difficulties. Emerge2Maturity uses two novel models to support decision making by players, the Capability Assessment Model for decisions about extraction, transformation, and integration levels of data sources and the Configuration Model for transition among decision making phases involving constraint levels, learning effects, and random events. Implementation of Emerge2Maturity uses JavaScript libraries, an Oracle database model, and a simple interface to show game progress and decision details.

Keywords: Data Warehouse, Maturity, Business Strategy, Serious Game, Simulation, Capability Assessment

1. INTRODUCTION

Using serious games to facilitate learning through simulating real life events is a fundamental goal of design science in the information systems discipline (Kankanhalli, Taher, Cavusoglu, & Kim, 2012; Zichermann & Cunningham, 2011). Serious games have a long history in business education. Since the Beer Game (Anderson & Morrice, 2000), scholars and students observed the potential benefits of using games to deliver knowledge and skills to participants. Serious games facilitate learning about areas difficult to grasp using traditional learning practices without practice and experience (Lainema & Makkonen, 2003). In business education, these areas include impact of information technology (Monk & Lycett, 2011), strategy, collaboration, integration, and development maturity (Leger, 2006).

With complexity and ambiguity in designing and implementing data warehouses, students and IT professionals struggle to understand relationships between business strategy and capability assessment in an organizational

 ISSN: 2473-3857
v3 n4388

setting. Although many university courses cover data warehouse design and implementation, traditional learning approaches fail to capture complexity and challenges that occur in real situations. Traditional university courses do not reveal ambiguity in real work (Lee, Koh, Yen, & Tang, 2002) and do not integrate knowledge and skills through experience (Boyle & Strong, 2006; Mackrell, 2009). Clearly, there is a strong demand for innovative learning approaches to help students experience complex relationships involving technology and organizational structures.

In this paper, we present the design and implementation of a serious game named Emerge2Maturity. As a business strategy game, Emerge2Maturity involves alignment of an organization's business intelligence strategy with its data warehouse capabilities. Players manipulate capabilities to maximize expected benefits subject to organizational constraints on budget and resources. Emerge2Maturity features two novel decision models and simulation of player choices to provide a serious game experience. In each decision phase, the Capability Assessment Model (CAM) calculates expected, simulated, and optimal results of player choices for resource choices. To transition between decision phases, the Configuration Model (CM) revises cost and benefit levels based on organizational learning rates and constraint levels based on occurrence of events. The game has been implemented using modern technologies such as Node.js, Aurelia.js, Express.js, Javascript-lp-solver, and Oracle database.

This work contributes to both research and practice. From a research perspective, Emerge2Maturity employs two novel analytical engines (CAM and CM) to help players evaluate tradeoffs among resource levels as an organization evolves to a mature state. Planned experimental evaluation of Emerge2Maturity will combine outcomes of player satisfaction, perceived learning, and task performance, a comprehensive approach not typically used to evaluate business strategy games. Both game development and evaluation adhere to design science principles. This approach corresponds with the view of Benbasat and Zmud (2003) regarding the need for more IT artifacts in the information systems discipline.

From a practice perspective, Emerge2Maturity is the first serious game developed for data warehouse strategy and capability assessment, an important yet difficult learning area. Existing approaches to maturity and capability assessment for data warehouses are descriptive lacking precision and manipulation ability for students. Emerge2Maturity allows players to experience simulated evolution of data warehouse infrastructure and thus increase understanding of the relationship among architecture and capability assessment.

2. RELATED WORK

The literature shows two approaches to determine the process of adopting data warehouse architecture, architecture selection and maturity models. Many researchers investigated the factors that affect architecture selection. Choudhary (2010) examined 11 factors hypothesized to affect architecture selection. Only 7 were found significant: resource constraints, perceived IT skills, need for integration, level of sponsorship, strategic view, urgency and need for information flow between organization units. The study identified factors influencing architect selection. Lower integration architecture is more likely to be selected if the organization has high resource constraints and low perceived IT skill among its staff. Moreover, the organization should select the moderate integrated architecture if the organization has low resource constraints, high need for data integration, and high sponsorship level. Finally, higher data warehouse architecture should be selected if the organization perceived data as a strategic resource. Ariyachandra and Watson (2010) found that information interdependence, task routineness and the level of sponsorship affect architecture selection through the perceived strategic view. The study concluded that organizations should upgrade to higher integration level architecture (enterprise data warehouse) as the strategic perception about their data increases. Thus, the strategic view is the key driver for architecture selection.

Maturity models provide a roadmap to evaluate and understand an organization's progress over time especially for technology capabilities and deployment. A maturity model consists of a sequence of levels representing an anticipated, desired, or typical evolution path of objects developed in discrete stages (Becker, Knackstedt, & Pöppelbuß, 2009). Sen, Sinha, & Ramamurthy (2006) proposed a capability maturity model for data warehouse development. Their model imitates the capability maturity model, a well-known maturity model for software development. The five levels from the capability maturity model have been applied

ISSN: 2473-3857
v3 n4388

but different key process areas (KPAs) and features were used. These KPAs and features are unique for the data warehouse development process.

Three features are considered in this paper: data warehouse size, data quality, and data integration. As the capability of an organization increases, organizations tend to process and store larger amounts of data. Organizations need to extract data from various sources into the data warehouse structure. Data quality becomes a major issue in larger data warehouses. Thus, data quality correlates with the data size. Data transformation process insures that the data has the right quality level. The architecture of a data warehouse by itself is not enough to determine the level of an organization's capability. Higher levels of capability require organizations to have higher alignment between their DW architectures and their business strategies. The strategy determines the integration level of a data warehouse.

We adopt aspects from both research streams about data warehouse architectures. For architecture selection, an architecture evolves from data integration levels influenced by constraints about resources and budgets over decision phases and events both internal and external to an organization. Choices for three capability features (data warehouse size, data quality, and data integration) determine the progress of an organization for achieving an architecture.

Serious games aid instruction about complex concepts and provide experience to participants. In higher education, serious games support interactive learning and engagement through entertainment (Prensky, 2007) with advantages over traditional teaching approaches (Pivec, 2004). Typically, serious games imitate real life events and simplify relationships among constructs. Serious games developed through this paradigm provide benefits other than entertainment (Michael & Chen, 2006). Serious games have been found effective in teaching college level students about business processes and integration of IT goals and business strategy (Monk & Lycett, 2011). Serious games can facilitate learning about aligning information systems design with business strategy, conceptualizing a holistic view of an enterprise system, grasping required technical skills, and showing effectiveness of collaborative work (Leger, 2006).

3. GAME DESIGN

This section presents the design of a role-playing game named Emerge2Maturity. In a role-playing game, participants make decisions in a realistic simulation. Participants can observe the effect of their current decisions and either commit their choices or revise their choices. The game continues over a number of periods or phases with the environment of the game progressing over the phases.

This section begins with motivation of the game about learning difficulties and objectives. The other parts of this section present the game flow and two models providing the foundation of the game to realize the learning objectives.

Learning Difficulties and Objectives

Students as well as professionals struggle to understand development of data warehouses in organizations over time. Data warehouse development is a complex process involving several related factors and extended time periods to reach a stable solution. Learners face challenges to observe changes and determine key success factors in data warehouse projects. Learners have difficulty understanding project relationships involving benefits and costs. Benefits of data warehouse deployment are often intangible especially during initial periods of usage. Benefits become tangible and increase as organizational units increase usage. In contrast, costs are tangible and high during data warehouse development especially with uncertain levels of data quality. Costs declines as benefits increase during usage of a data warehouse over time. Learners need to balance benefits and costs as organizations acquire capabilities to support an organization's strategy for data warehouses.

The design of Emerge2Maturity addresses these difficulties. The game decomposes the complexity of data warehouse development into a sequence of standard steps. To help focus learners on key factors, the game provides common factors across organizations. Learners remain focused on important aspects of data warehouse development without distractions of other elements related to specific situations. The game combines aspects of strategy and capability to help learners understand the relationship between them. The game simulates the development process to show trends, costs, and benefits with increased profits and decreased costs over time. Simulation provides a real-like situation where learners can observe results of their decisions before implementing

ISSN: 2473-3857
v3 n4388

them. The simulation uses models to quantify costs and benefits related to choices. These models depict the relationship between the costs of acquiring capabilities to develop a data warehouse and benefits for using a data warehouse.

Game Flow

Emerge2Maturity involves two models for transition among decision phases (Configuration Model) and capability assessment in each phase (Capability Assessment Model). Before describing these models, we present the game flow showing usage of the models.

Emerge2Maturity provides decision making over a number of phases as depicted in Figure 1. In each phase, players make sequential or joint decisions about capabilities for extraction, transformation, and integration. Players attempt to maximize net benefits using details about costs, benefits, and constraints. The demand for information assets provided by capabilities is stochastic so players deal with uncertainty in assessing capabilities. The game evolves over a number of phases representing budgeting or decision-making periods. The learning effect progresses over the phases, impacting coefficients for costs and benefits. As an organization acquires capabilities, costs decrease and benefits increase. Events influence coefficients and constraints on capabilities. The game terminates after a specified number of phases when an organization reaches its highest maturity level. The Capability Assessment Model (CAM) provides the framework about decision-making in each phase, while the Configuration Model (CM) supports transition among phases.

To manage complexity from a large number of data sources, Emerge2Maturity groups data sources into categories. Categories facilitate determination of cost and benefits derived from individual data sources as all data sources in a category have the same feature values. Emerge2Maturity uses features for technology, complexity, and size to define data source categories. Feature values determine levels of fixed costs, variable costs, production, benefits, and risk. For example, legacy technology involves higher fixed costs and complexity influences production, variable costs, benefits, and risks.

Capability Assessment Model (CAM)

The Capability Assessment Model (CAM) provides the foundation for decision making in Emerge2Maturity. The CAM uses a cost-benefit, demand-driven approach, maximizing profit from capabilities, subject to constraints on capabilities and budget. The CAM is an educational model to demonstrate relationships among important variables of data warehouse capabilities.

To simplify presentation in this paper, Figure 2 shows a conceptual representation of the CAM. We omit detailed math to focus on the major elements.

The CAM manipulates three decision variables (data size *X*, transformation level *Y*, and integration level *Z*) used in processes for extraction, transformation, and integration affecting an organization's capabilities. Extraction involves selecting data sources and transporting data to include in a data warehouse. Transformation involves increasing data quality through operations on individual data sources. Integration involves combining data from different sources, matching and consolidating common data.

Due to embedding CAM in the Configuration Model, Δ represents incremental capabilities added in a phase. Each decision variable involves levels for each data source category.

To indicate the contribution of decision variables for costs and benefits, the CAM uses functions for production, costs, demand, benefits, and profit. Production (*P*) is the number of queries that each data source category can support. Total costs (*TC*) are a summation of variable and fixed costs for each data source category. Demand (*D*) represents expected production plus stochastic risk for each category. Benefits (*B*) involves a benefit rate for each category applied to stochastic demand for each data source category. Profit is revenue (R) from benefits minus total costs (TC), the sum of functions for fixed costs (FC) and variable costs (VC). The Configuration Model determines the values for coefficients and weights used in functions of the CAM.

The CAM uses stochastic demand, common in models in operations management (Schmitt, Snyder, & Shen, 2010; Miranda & Garrido, 2004) and econometrics (Ben-Daya & Hariga, 2004; De Castro, Tabucanon, & Nagarur, 1997; Browne & Zipkin, 1991) Demand is a function of production plus risk. Expected demand is the production level determined by values for decision variables and the uncertain risk or error term. Risk is modeled as a Normal distribution with mean of 0 and standard deviation of *r*, a function of features of a data source category.

ISSN: 2473-3857
v3 n4388

Data size has a separate impact in each function as well as combining with transformation level and integration level. Because of these dependencies, the CAM is not separable on data size.

The optimization model involves profit maximization for each data source category subject to constraints. The CAM contains constraints on the budget for total costs, minimum capability levels (data size, transformation level, and integration level) for each data source category, dependency of integration on transformation for each data source category, and maximum capability levels (data size, transformation level, and integration level) for each data source category. The optimization procedure solves the CAM model for the expected demand without the risk term.

Configuration Model

Configuration of a phase involves new levels for constraints about budgets and capabilities, revised weights applied to coefficients for costs and benefits, and random events that influence budget constraints. Constraint levels are determined dynamically based on organizational strategy and capabilities achieved in previous phases.

Coefficients for costs and benefits have base values. However, cost and benefit coefficients change during the game based on organizational learning. Weights are applied for capability costs and benefits to reflect learning effects. As an organization acquires capabilities, it becomes more efficient with decreasing costs for deploying resources and effective with increasing benefits. Learning curve has been used to explain the relationship between relative efforts and cost reduction and increased performance in several disciplines such as software development (Pendharkar & Subramanian, 2004) and help desk support (Deng, 2005). Emerge2Maturity uses Wright's cumulative average model (Wright, 1936) for cost and benefit coefficients,

$$W = a * (Effort)^{b}$$

where a and b are parameters. The value of W starts at 100% with zero effort. The a parameter involves the initial cost or benefit, while the b parameter involves the log-log slope of the function.

Events are occurrences of actions with long-term consequences initiated externally or internally by an organization. An internal event is an occurrence of actions within an organization such as a merger or divestment. An external event is an occurrence of actions that an organization has no control such as a recession, regulation, or litigation. An organization reacts to events by adjusting their strategic view and/or capabilities. Emerge2Maturity uses a small set of events with a probability of occurrence. If an event occurs, phase configuration randomly adjusts budget constraints.

4. GAME IMPLEMENTATION AND DEMONSTRATION

This section describes the game controller, the database model, and game scoring details. Then game implementation details using JavaScript and Oracle database are summarized. Finally, this section demonstrates the game interface showing results from an actual game play.

Game Controller and Database Model

The game controller uses a database with static configuration details and dynamic tracking of game play. The data model in Figure 3 shows the internal database structure of the game. The database has configuration tables in the top part of the database diagram and operation tables in the bottom part of the database diagram. The game controller can provide various experiences based on the configuration tables for phases, category features, phase constraints, and category constraints. Operation tables contain results of a game for the overall game, phases, and simulation attempts in a phase for each decision variable. Emerge2Maturity tracks decision variable choices by players as well as profit (expected, actual achieved by player choices, and optimal).

In a simple game type, players can provide capability decisions in a sequential order for all data source categories in the order of extraction, transformation level, and integration level. Players get the correct solution and continue to build on it for the next phase. In the more complex game type, players make decisions jointly for each category with the chance to continue on committed decisions. Phases can vary from 3 to 10 providing a short or long-term progression from an initial strategic view into the highest strategic view. Players are provided several simulation attempts to revise capability choices for each decision variable.

During game play, the game controller uses recorded values to calculate scores. Game scoring tracks progress of players across phases. It also encourages players to play again to beat their previous scores. During the game, players

ISSN: 2473-3857
v3 n4388

may simulate their data warehouse design. However, when they commit their decision, demand is sampled from a demand distribution. Both the profit from simulated reality and the profit from expected demand are shown and compared. In more advanced version of the game, scores may be used to compare efforts among competitors and players may play against each other or collaboratively to achieve goals.

Implementation Details

Emerge2Maturity implementation uses the MEAN stack architecture, a common architecture for JavaScript-based web applications. The MEAN stack is a preferred architecture for web application development due to its light overhead, ease of use and customization, and a large and evolving library of packages that provide a wide range of functionalities to the developer. This stack consists of Node.js, Aurelia.js, Express.js, JavaScript-lp-solver, and Oracle DB. Node.js is a server-side JavaScript framework that provides the game functions and services via a REST API. Aurelia.js is a client side JavaScript web application framework, which consists of the game's views and controls. Express.js is a routing framework used to connect the Aurelia client to the Node server. The lp-solver. a JavaScript package, finds values of decision variables that optimize the profit based on phase constraints. Emerge2Maturity uses an Oracle database to initiate, save and retrieve game related data. The game can be deployed on a Linux or Windows server, and does not require an external web server such as Apache, as a bundled Node.js server handles all API requests from clients.

To extend the implementation of Emerge2Maturity game, two gamification elements were added to the game. Playing games is a daily routine for many people. Enjoyment is the main motivation to play games. Gamification elements, if combined with educational materials, can also bring enjoyment to the learning process and help accomplish learning outcomes. Gamification is defined as "the use of game design elements in non-game contexts" (Deterding, Dixon, Khaled, & Nacke, 2011). According to Werbach (2015), there are 15 game elements that are responsible for creating the enjoyment in games. Emerge2Maturity uses points to reward players for their accomplishments and a leaderboard to show the name and the score of the highest ranked players.

Game Demonstration

The game interface utilizes the Aurelia.js framework. It allows game interactions between a player and the game. The game is available online using any Internet browser.

Initially, a player provides some demographic questions and obtains login credentials. Then, a player chooses a game and phase 1 starts. At the beginning of a phase, the game provides the player with some qualitative information about the data sources categories (Figure 4). Then, the player selects the best combination of data sources that maximizes the organization's profit (Figure 5). Players simulate their decisions and see potential results from a stochastic demand. Players have specified number of simulation attempts before they must commit one answer. After committing an answer, Emerge2Maturity uses the lp-solver to determine the optimal answers and show them to the player. The player continues to the transformation decisions and integration decisions.

At the end of each phase, the game controller saves the player's decisions and outcomes. The game controller initiates the next phase and the player continues the game. At the end, the game controller calculates the total score and shows it to the player. It also compares the score to the previous scores and rank players in the leaderboard (Figure 6).

5. CONCLUSIONS AND FUTURE WORK

This paper presented the design and the implementation of Emerge2Maturity, a serious game for strategy and capability assessment of business intelligence. The game simulates the capability decisions for data warehouse development using the Capability Assessment Model and Configuration Model, novel decision tools implemented using JavaScript and Oracle technologies.

Future research will evaluate Emerge2Maturity using a combination of survey and experiment. Most serious games are evaluated by engagement and learning outcomes (Connolly, Boyle, Macarthur, Hainey, & Boyle, 2012). To evaluate game engagement, the questionnaire method is commonly used. In contrast, experiment is used to measure items that are difficult to evaluate using surveys such as learning outcomes. Hainey, Connolly, Stansfield, & Boyle (2011) used a combination of pre-test/ post-test and experiment to evaluate an educational game. Player's perceptions and knowledge were assessed using the pre-test/post-test while the learning outcomes were assessed using the experiment. Participants

ISSN: 2473-3857
v3 n4388

were randomly assigned to control groups and treatment groups.

To extend Emerge2Maturity, we will develop the Strategy Assessment Model (SAM), a novel strategy model to determine data warehouse development strategy. The current version of the game determines the development strategy by specifying the number of phases and constraints for each phase. SAM will allow players to determine the strategy and provide players the freedom of specifying constraints and number of phases. This higher-level version of the game requires not only capability assessment but also the strategy assessment using information interdependence, task routineness, and level of sponsorship factors.

6. REFERENCES

Anderson, E. G., & Morrice, D. J. (2000). A simulation game for teaching service-oriented supply chain management: Does information sharing help managers with service capacity decisions? *Production and Operations Management*, Spring 2000, 9 (1), 40-55.

Ariyachandra, T., & Watson, H. (2010). Key organizational factors in data warehouse architecture selection. *Decision Support Systems* (49:2) Elsevier B.V., pp. 200–212.

Becker, J., Knackstedt, R., & Pöppelbuß, J. (2009). Developing Maturity Models for IT Management, *Business & Information Systems Engineering* (1:3), pp. 213–222.

Ben-Daya, M., & Hariga, M. (2004). Integrated Single Vendor Single Buyer Model With Stochastic Demand And Variable Lead Time. *International Journal Of Production Economics* (92:1), pp. 75–80.

Benbasat, I., & Robert W. Zmud. (2003). The Identity Crisis Within The Is Discipline: Defining And Communicating The Discipline’s Core Properties. *MIS Quarterly* (27:2), pp. 183–194.

Boyle., T. A., & Strong, S. E. (2006). Skill requirements of ERP graduates. Journal of *Information Systems Education* 17 (4), 403-412.

Browne, S., & Zipkin, P. (1991). Inventory models with continuous, stochastic demands. *The Annals of Applied Probability* (1:3), pp. 419–435.

Choudhary, R. K. (2010). Key organizational factors in data warehouse architecture selection. *Vivekanada Journal of Research* (24), pp. 24–32.

Connolly, T. M., Boyle, E. A., Macarthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education* (59:2) Elsevier Ltd, pp. 661–686.

De Castro, E., Tabucanon, M., & Nagarur, N. (1997). A production order quantity model with stochastic demand for a chocolate milk manufacturer. *International Journal of Production Economics*, (49:2), pp. 145-156,

Deng, X. N. (2005). Differentiating the Effect of Cumulative Experience and Learning: A Field Study of Help Desk Support. *AMCIS 2005 Proceedings*, 341.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining ‘Gamification’. *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA* ’11, p. 2425.

Hainey, T., Connolly, T. M., Stansfield, M., & Boyle, E. a. (2011). Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers and Education* (56:1), pp. 21–35.

Kankanhalli, A., Taher, M., Cavusoglu, H., & Kim, S. H. (2012). Gamification: A New Paradigm for Online User Engagement. *Thirty-Third International Conference on Information Systems*: Orlando, FL.

Lainema, T., & Makkonen, P. (2003). Applying constructivist approach to educational business games: Case REALGAME. *Simulation & Gaming: An Interdisciplinary Journal*, 34, 1, 131-149.

Lee, S., Koh, S., Yen, D., & Tang, H-L. (2002). Perception Gaps between IS Academics and IS Practitioners: An Exploratory Study. *Information and Management*, 40, 51-61.

Leger, P. M. (2006). Using a simulation game approach to teach enterprise resource planning concepts. *Journal of Information Systems Education* 17(4): 441-447.

Mackrell, D. (2009). The work readiness of Master of Information Systems International students at an Australian University: A pilot study. *Issues in Informing Science and Information Technology*, 179-191.

ISSN: 2473-3857
v3 n4388

Miranda, P., & Garrido, R. (2004). Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. *Transportation Research Part E: Logistics and Transportation Review*," (40:3), pp. 183-207

Monk, E., & Lycett, M. (2011). Using a Computer Business Simulation to Measure Effectiveness of Enterprise Resource Planning Education on Business Process Comprehension. *International Conference for Information Systems*, pp. 1–10.

Pendharkar, P., & Subramanian, G. (2004). An Empirical Study of Learning Curve Theory ' s application to the Software Development Effort- Experience Relationship in Software Engineering and Project Management An Empirical Study of Learning Curve Theory ' s application. *AMCIS 2004 Proceedings*

Schmitt, A., Snyder, L., & Shen, Z. (2010). Inventory systems with stochastic demand and supply: Properties and approximations. *European Journal of Operational Research*, (206:2), pp. 313-328

Sen, A., Sinha, A. P., & Ramamurthy, K. (2006). Data Warehousing Process Maturity: An Exploratory Study of Factors Influencing User Perceptions. *IEEE Transactions on Engineering Management* (53:3), pp. 440–455.

Werbach, K. & Hunter, D. (2012). For the Win: How Game Thinking Can Revolutionize Your Business. *Wharton Digital Press*, Philadelphia. Zichermann,

Wright, T. (1936). Factors Affecting the Costs of Airplanes. *Journal of Aeronautical Science* 3: 122-128.

Zichermann, G., & Cunningham, C. (2011). Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. *O'Reilly Media*, Sebastopol, CA, ISBN-13: 978-1449397678.

ISSN: 2473-3857
v3 n4388

Appendices and Annexures

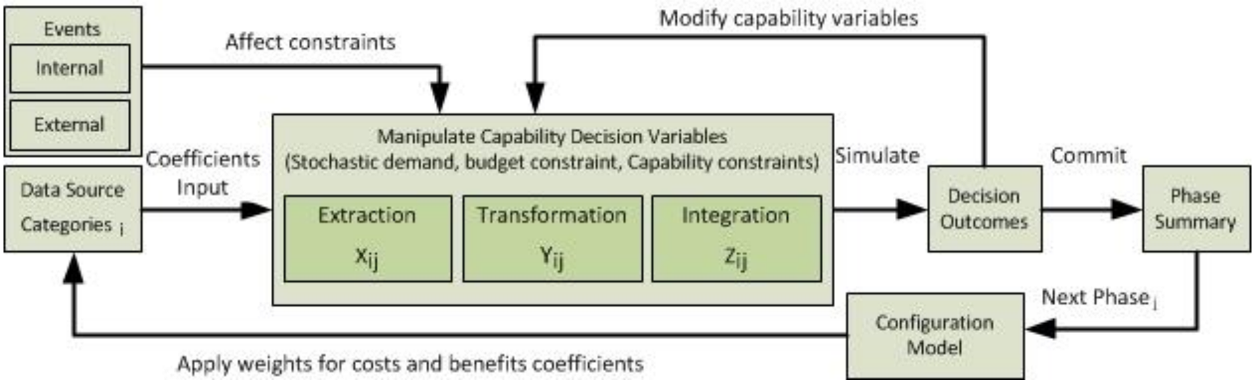

Figure 1: Game Flow

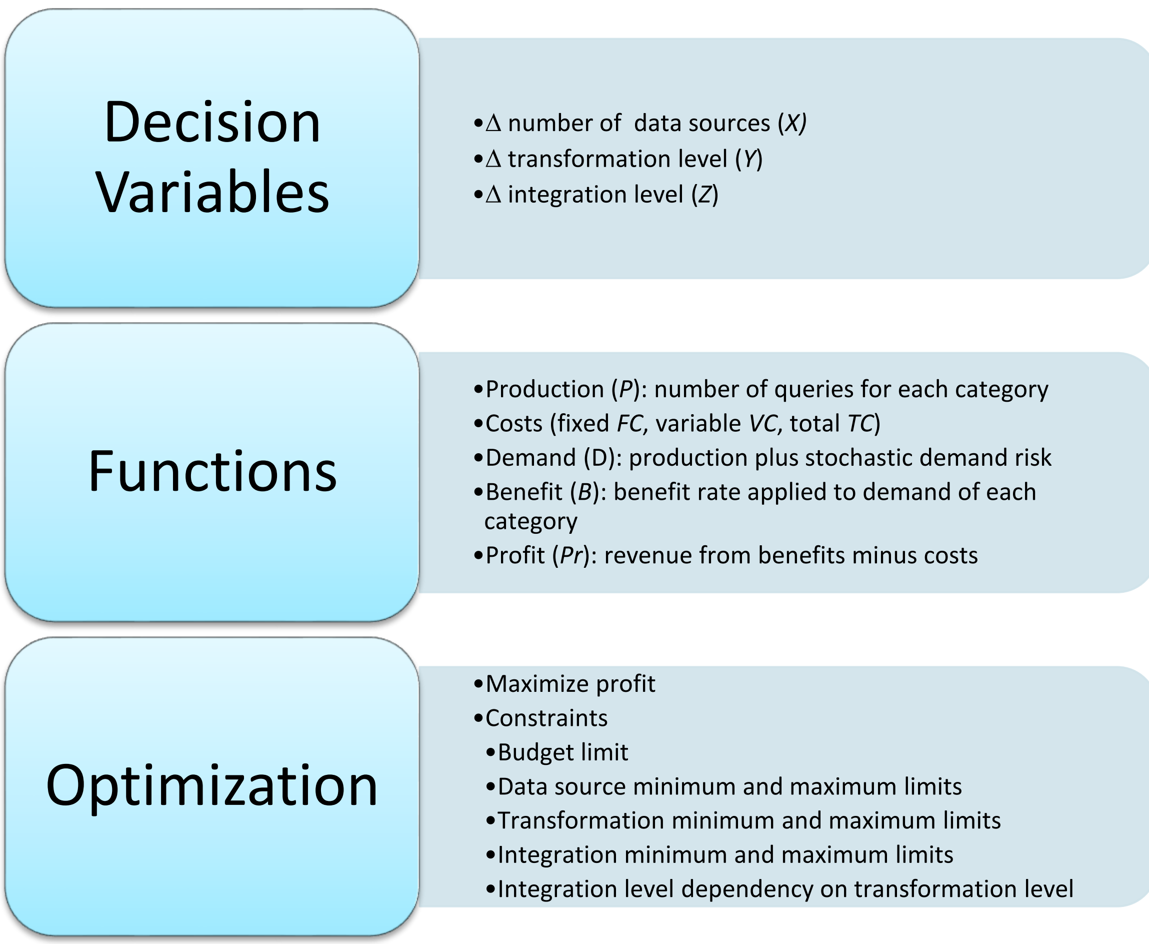

Figure 2: Elements of the Capability Assessment Model (CAM)

ISSN: 2473-3857
v3 n4388

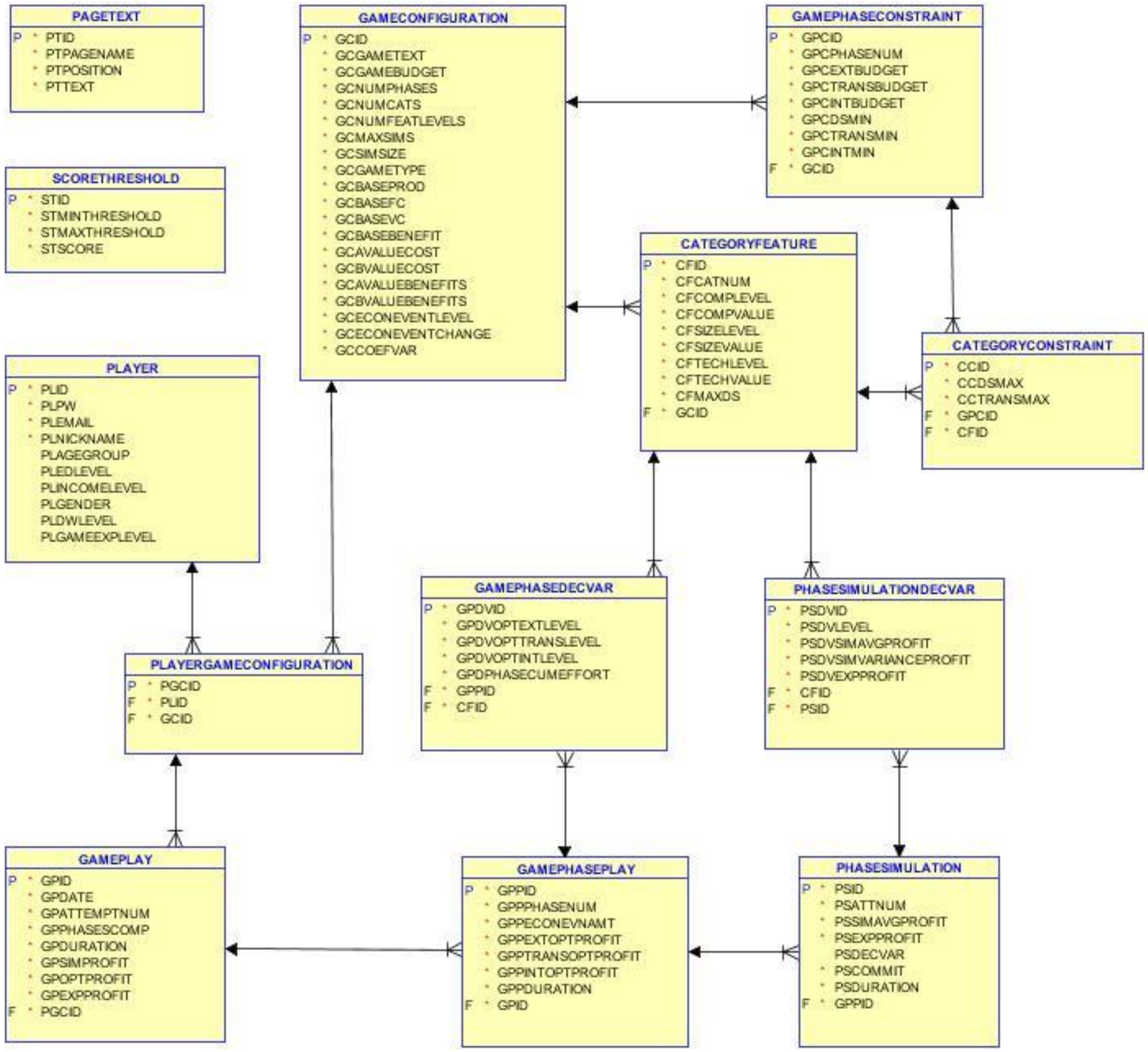

Figure 3: Data Model for the Game Controller

ISSN: 2473-3857
v3 n4388

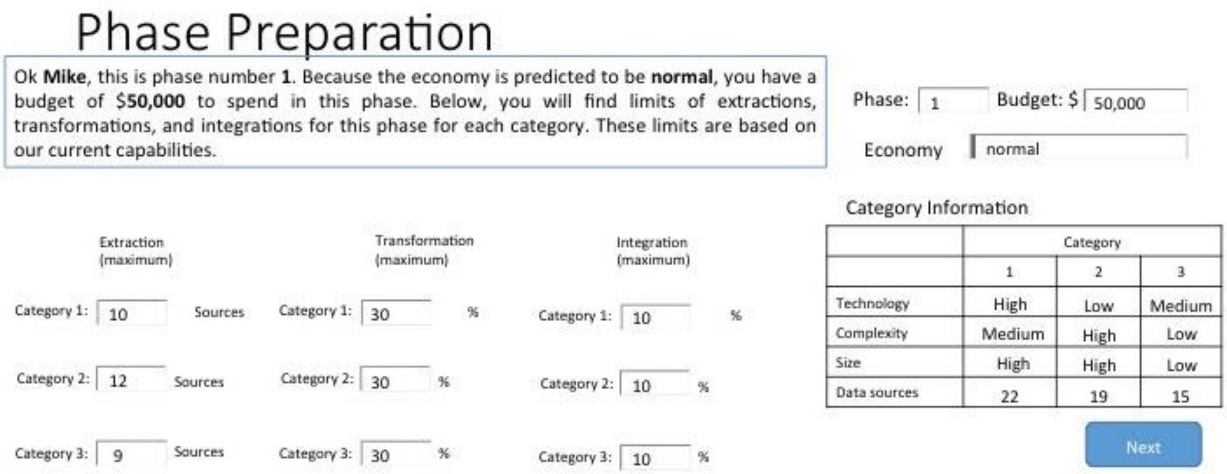

Figure 4: Emerge2Maturity Game Phase 1 Interface

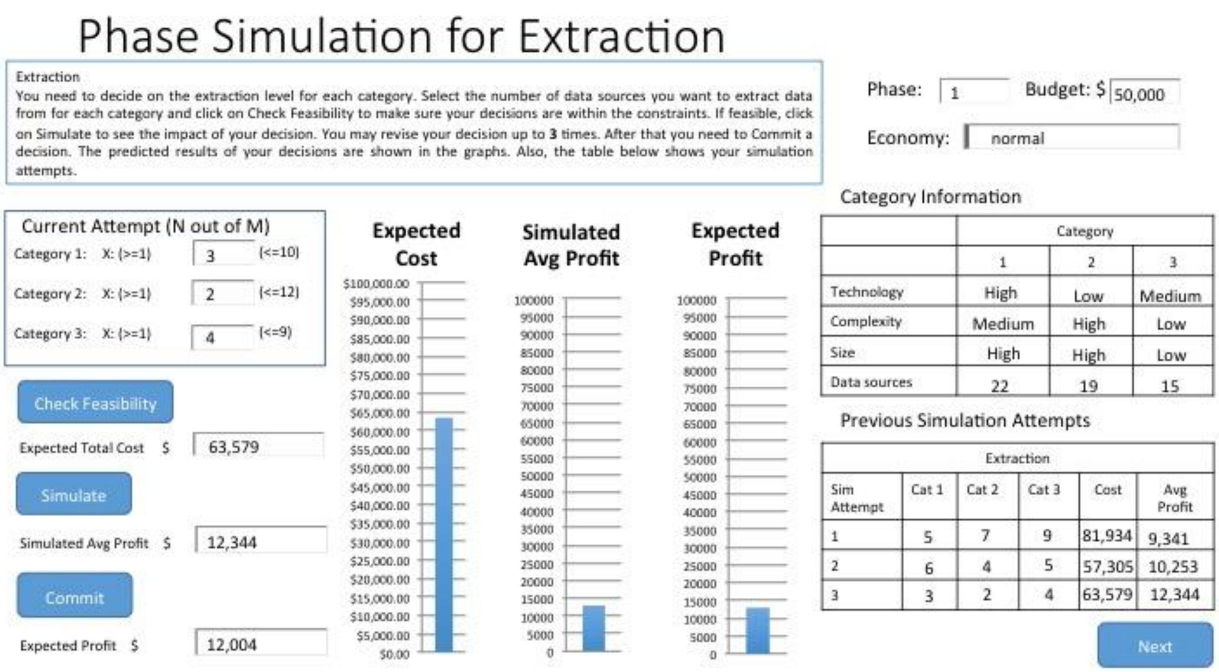

Figure 5: Phase Simulation for Extraction Decisions in Phase 1

ISSN: 2473-3857
v3 n4388

Figure 6: Game Score and Leaderboard Ranks