ISSN: 2167-1435
v30 n2545

RAPID, A Short-Life-Cycle Methodology for Development of Information Systems

Herbert E. Longenecker, Jr.
longeneckerb@gmail.com
School of Computing
University of South Alabama, Mobile, AL 36608

Jeanne Baugh
baugh@rmu.edu
Computer and Information Systems Department
Robert Morris University, Pittsburgh, PA

David L. Feinstein
dfeinstein@southalabama.edu

Shweta Purawat
spurawat@gmail.com

School of Computing
University of South Alabama
Mobile, AL 36608

Abstract

The rationale and the components of a short-life-cycle methodology are presented. Steps in the process have been examined and tailored to 1) be understandable to nontechnical participants, 2) simplification of traditional process to aid in understandability, and 3) produce functional results quickly. The approach has been successful for both undergraduate and graduate student classes and in professional settings. Some of the simplifications (TDLCRM and Function Flow, see below) may be controversial with traditional practices.

Key Words: Database, Methodology, Conceptual Data Modeling, Work Flow, System Analysis and Design

1 INTRODUCTION

The Nature of the Issues

To develop an information system requires the intersection of people and technological skills. People skills extend to team mates and to clients. Relationships are necessary for the implementation of successful systems (Barki and Hartwick, 1994; Hunton and Beeler, 1997). It is the belief of the authors that participation in the life cycle is important so that developed systems are acceptable to clients. It is also our belief that if these people relationships are not formed, and if clients are not involved, resistance will develop and systems will not be accepted (Markus, 1983).

Meetings
A good way to have people participate is to have a set of well run meetings. Bostrom and Clawson (1991) identified that successful meetings have a definite goal, and consist of an agenda of tasks wherein each task has a known structure. Then, once a desired result has been proposed, the meeting designer can select from these known tasks. Here, a team that is familiar with a particular methodology will be at an advantage, and will be able to guide other participants.

Types of Problems to Solve
A usual problem to solve in computer science is solving algorithms, that is, given an input and an output, the goal is to determine the process: Find a finite sequence of steps that when executed produce the desired results and then stops. The "finding" process is known as solving a process of analysis. Programming classes spend much if not all their time solving such analytic problems.

However, in developing information systems the input (current situations and goals) must be known or determined. Yet, the result of development of a new system does not exist. Therefore, to produce a new system an analytic approach will not work. What is required is a precise knowledge of the process to generate a new system. A process consisting of a series of steps must be known. These steps are called a methodology. This is a synthetic process and is similar to building a cake given ingredients and a recipe. The "solving process" for cake building and for building a new information system requires executing the steps of the methodology.

Development Team and Client Learning
In a professional setting it is assumed that the development team members are very skilled in their methodology for use in the project to develop an information system. However, in a student team led project students are not advanced in their knowledge and in fact may be struggling with the concepts, and at best may have little to no experience utilizing a methodology. This can be compounded by the fact that most clients have no experience with the methodology, yet have considerable experience in their own responsibilities.

Reinicke and Janicki (2010) have observed that one semester is not enough time to become effective in developing a new information system, whereas a two semester approach yields a 65% success rate in project completion for "real" clients. Their approach involves both a learning curve for the systems analysis fundamentals and actually carrying out the process including full implementation for their client. In their research, Seyed-Abbassi, King and Wiseman (2007) also feel that that one course is not enough for the student to adequately understand the necessary information system topics. But, if the instructor is willing to invest a more than usual amount of time with the students in their learning process, certainly the payoff is rewarding to all involved, the instructor, the student and the client.

Baugh (2011) has observed that in single semester courses project oriented learning is motivating for students to place value on learning goals and time in completing even challenging tasks. These studies involved freshmen through doctoral level courses. One of the salient points of these methods is evidenced by the fact that 74% of the students involved in the project oriented approach felt challenged as opposed to 38% involved in traditional approaches. One of the potential negatives of the approach is that considerable time is required of faculty—this must be recognized by reduced load and limiting class size. The course instructor must be able to be responsive to student communications and evaluation in a very timely manner.

In the process of developing a new information system, client learning must be emphasized If our students are having challenges at the end of a single semester, and probably still at the end of two semesters, imagine the feeling of clients. Unless the methodology being used is very simple to understand, it is easy to see that clients may feel not only overwhelmed, but antagonistic. Such feelings can only distract from the real work that must occur in transforming their business process commensurate with development of the inextricably interwoven information system Therefore, it is essential that all involved understand what the client needs are. (Deperlioglu, Sarpkaya & Ergun, 2011) The designer is modeling the client's view of his universe and sometimes that view takes a little time to elicit from a client. Helping the client to understand the design process will lead to less frustration on the part of the client and an actual partnership in the process between the designer and the client.

When a student is doing work for a real client, it is important that the system designs are

correct (Choobineh, & Lo, 2004). The students benefit from the discussions of real encounters with real clients and real issues/problems. "Real" projects have always been a great way to get the student involved in what he is suppose to be learning, especially a project that the he has selected for himself (Foltz , O'Hara, Wise, 2004)

Goals of the Paper

The purpose of this paper is to present a methodology that is easily learned by both students and clients. The approach has be utilized with student teams with undergraduate and graduate teams and with a few professional projects. We call this short-life-cycle methodology RAPID. To give students experience in the "real world", a bridge must be established between business and academia (Courte, and Bishop-Clark, 2005). This methodology creates such a bridge.

2 CHARACTERISTICS OF THE METHODOLOGY

The RAPID methodology of the paper should achieve the following characteristics.

a—Easily learnable by students and clients
b—Provides end-user satisfaction
c—Be well specified
d—Have stepwise transaction closure

a. Learnability
   Each step in the RAPID methodology can be completed within several hours by a development team. Written directions and an example are furnished to the development team and clients. This material is used as one or more agenda steps of a meeting. Description of the steps is written in simple prose. All steps except code production are attended by all members development team and clients. Code production is completed in small modules and then brought to the next meeting for review.

b. End-User Satisfaction
   Since clients are not only participants but approvers, any rough edges that exist can rapidly be removed and improved. Changes can be made at any stage with recycling of thinking. Keeping the client involved in the design process will not only create a good system design, but it will ensure that the client is aware of any unforeseen issues that may arise. It has been the experience of the authors that often the client will not know what they want until they see sample designs, prototypes or rough layout of screens. Our approach is similar to Lean UX (Cryillo, 2011)

c. Well Specified
   Each step in the methodology must be clear, well researched, clearly written, and productive of desired results that advance the step of the life-cycle. The methodology may evolve provided changes of each step produces needed work products for the next step.

d. Transaction Closure
   The property of transaction closure implies that techniques enabling progression from one to the next step are reversible. That is for the process to convert A→B meaning there is a process that enables B→A.

3 STEPS OF THE METHODOLOGY

Project management is important to the success of this or any other methodology. We utilized the methods of Kathy Schwalbe (2010) although the details will not be discussed here.

Table 1 shows the steps of the RAPID methodology and contains references to illustrations to the methodology in subsequent figures.

Developing a scope statement is a well established territory and no additional time will be spent here other than to say it is important because it frames the work to be accomplished. In the Authors' professional design consulting experience, if the scope is not well defined, the client will always want more.

SWOT Analysis

In a SWOT meeting a problem or situation must be identified before the meeting. A SWOT analysis is completed knowing there is a business problem or situation that needs researched, analyzed and documented. Participants should be invited to represent all significant viewpoints within the organization. Strengths of the organizational process should be collected first. People prefer to talk about what is being done well. A list can be obtained with a PC, Word, and a projector showing the data collection. Then a list of issues or negatives can be collected. If management is present this may difficult to elicit since people may be at risk for speaking. Either the group needs to be limited, or some significant protection needs to be offered to members.

ISSN: 2167-1435
v30 n2545

But why collect the negatives? Figure 1 illustrates the value of collecting negatives.

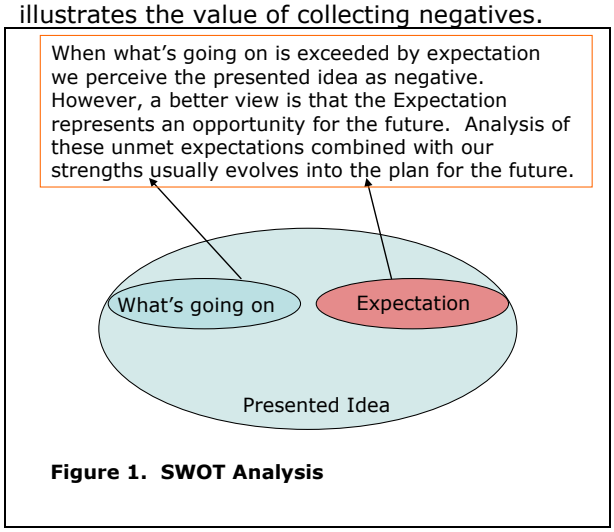


Figure 1. SWOT Analysis

A negative idea can only exist in reference to an unmet-expectation. The weaknesses within a current system need to be defined and corrected in the new system. What is not working and how should things work? The expectations are and represent strengths worth cultivating. As sufficient data is collected, a pattern will emerge, and will develop into a plan for a new process. During the final stages of the SWOT meeting, it is the task of the participants to draft a summary statement describing the new business process which explains the nature of **who** will **do** what in the new business process. This short statement we will call the "story". Figure 2 contains the results of a sample SWOT.

Requirements

The goal of requirements determination begins with identification of stakeholders of the business process. It identifies those who have a regular responsibility to the system. Stakeholders are not identified by name, rather, by the function they carry out. Input from these stakeholders is essential. Often a person's functionality within an organization is not widely known by others within that organization. Their duties, tasks or job responsibilities may not be recorded in any organized manner. This may lead to problems down the road that can be avoided in the very beginning. Communication skills of the designer are extremely important. (Baugh, Davis, Kovacs, Scarpino, Wood, 2009) Students are taught to "listen" to the client. Students not only need to hone their technical skills, but they need to develop a communication expertise that will allow them the best possible chance at drawing out of the client what their data needs may be.

In Figure 3 stakeholders have been listed in cells of the table in column 1. Based on the "story" activities derived from a review of the "story" are listed for each stakeholder. The facilitator of the meeting asks the question, "What is it that each person should "do" on a regular basis. An attempt is made to list the tasks in the usual order in which they are performed.

After this task is done a sequential order is developed for all of the tasks. It will be useful at the next stage.

It is possible that tasks not identified in the initial "story" will be found. It should be considered as part of a discovery process.

As this drawing is completed, it should be maintained to serve as a checklist to ensure that all of the functions presented are implemented.

Work Flow

A table is developed which is the number of stakeholders wide + 1. Stakeholder names are entered from left to right according to those closest to the customer (column 1) to least involved (column n). The last column is reserved for developing procedure.

Figure 4 presents the results of the transferring of the requirements to the workflow form. The requirements from each stakeholder cell are transferred to the columns of the workflow. Each requirement is aligned according to the sequence number determined during construction of the requirements.

The procedure column combines creation of short sentences that are in order according to the sequence number. Each sentence consists of a description of what the stakeholder does at each step.

A final story is created by copying the procedure and removing returns. This version is called the "final story" and is utilized in determining the database design. If new tasks have been developed during this process, they should be echoed back to the requirements.

Database—TDLCRM

A TDLCRM or Top Down Levelized Conceptual Relational Model is a conceptual relational model. It is a simpler model than its cousin the ERD in that it uses only 1:N relationship, uses no boxes to encase entity names, and

uses none of the symbols on relationship lines. It requires that the 1 side of the relationship is always drawn above the N side which is drawn lower on the page and to the right of the one side. (See the derived example in Figure 5.) Other characteristics of the model are illustrated in Table 2.

People (designers and clients) who have never seen an ERD find the model very simple to learn. Students learning the ERD seem to have problems defining what the entities are and the relationships among them. But with this model, defining "who" the entities are and "what" they do is very intuitive. Many clients as the look at the model say, "This is exactly what we do!" This is in fact a reasonable since the point of this methodology is that from functional requirements, that is what "who" "do", the database can be derived. The process is illustrated in figure 5.

The "story" derived from the work flow is marked to illustrate entities (nouns) and verb phrases to spot relationships. Status of entity words are identified and marked. Roles are also identified as are attributes. Each is marked with a different color.

Entities are drawn and arranged following the rules of Table 2. If a status was detected for an entity, that entity is underlined. Relationship lines are drawn differentiating controlled attributes with dashed lines. We find that PowerPoint is a convenient tool to make these drawings.

Screen Flow
It is the responsibility of the information system, given that the information and physical flows have been captured in the database as represented in the TDLCRM, then in a sort of back-handed manner, it is the job of the end-user to edit the database contents specified by the TDLCRM. The interface between the end-user and database are the screens that enable editing of the database.

Layout
In Figure 7 the location of a main menu is determined by consideration of the Level 1 entities specified by the TDLCRM. Each of these entities becomes a main menu item. Access to functions subsequent to the main menu may be controlled by setting Booleans in the person table, and then by utilizing these Booleans to control visibility to the leaves of the menu.

Within the login process the user must be authenticated by a user-name and pass word. Once this has occurred, the system will capture a PersonID from the database and enter it into a table called SVtable, for session variable table. Each entity that will be altered within the application is entered in this table. The row is passed as a session variable to the next page. This method maintains "state" for this web application. CreateSVtable, GetSVtable, and PutSVtable are managed by a dll called JSIM.dll which is made available to any academic programming group at no charge.

We utilize a pseudo-language to express the screen design. It is based on the CRUD operations, and gives the designer a simple mechanism to layout screen flow. The essence of the language is illustrated in Figures 8 and 9. In figure 8,

List, Add, and Edit

functions are shown as well as their translation to a screen layout. In Figure 9 a complete function design is shown for the application depicted in preceding figures. Figure 10 depicts a partial translation of the function flow into two pages of the application layout. Figure 11 is a special form of the select function.

Implementation
In this paper, details of application coding are not presented. Usually VB.net and SQLServer have been used. An n-tier architecture has been employed consisting of a presentation, business and data access layers. Stored procedures are developed to minimize the risk of SQL injection attacks.

A database with no access to the outside world (that is, no IIS) maintains the data and stored procedures. As a new version of the database is posted to the system, each member of the development team receives a copy to ensure they are working with current information. Changes are batched during the day and updates to the database is made during the evening and posted as rapidly as possible. Database updates are distributed as "script". As this transaction is implemented end-user-programmer's database is dropped, and the new one is installed.

To enable ease of integration, a main menu is written first and installed. Subsequently, functionality is added stepwise, and is immediately reviewed by the end-user and

technical community for consistency and satisfaction. This continues until done. Changes are made and replaced as soon as possible and are scheduled for the next review session. Major errors in the function flow are also corrected during these review sessions, and documentation is updated.

In this paper there will not be a detailed discussion of implementation methodology. SQL Server 2012 and Visual Studio 2012 are the primary tools used in these systems. Conversion of the TDLCRM is very straight forward: For each entity script is developed. The table nameID serves as the primary key. Foreign keys can be determined immediately by inspection of relationship lines moving upward from the named entity. Attributes, are determined as soon as possible in development process, certainly by the layout stage.

It is very important to spend time generating realistic sample data. It is very important to resist the urge to label fields with values such as x or xxx. Such sloppy behavior will lead to disaster later in the process.

Development of the application should proceed from top down and from left to right (see Figure 9). However, it is very important to invest time in developing the main menu with permission systems. Once this form is implemented, the rest of the application can be integrated with minimal disruptions to the development process. Failure to heed this direction can cause enormous problems with integration.

4 Discussion

In Table 1 the steps of the RAPID methodology have been presented as s short-life-cycle process that has been used in over a 100 successful attempts to implement an information system. Both the TDLCRM and Function Flow processes are new. However, both of these processes have been successful in implementing successful information systems.

5 Conclusion

A simple to use short-life-cycle methodology that can be utilized to involve participants at all steps except the implementation stages has been presented. A student evaluation (see Figure 11) suggests the methodology can be utilized and put into practice quickly and produce successful outputs.

Real world projects allow the students to "learn better through a particular domain of their interest" and "see the practical value of what they learned" (Robbert, Wang, Guimaraes, Myers, and Martha, 2000). It is very important to bring such experiences into the classroom. The course described here is a step in the right direction (Kim, Hsu, and Stern, 2006). However, there are a few recommendations:

- ✓ Be careful to limit the scope of each project. The student/student team may not realize that the project they wish to do for a client may be too much for someone new to information system work. Also, the project must be completed in a limited amount of time (one semester or two). Perhaps deleting some of the functionality may be necessary.
- ✓ Make sure students interact with the client often. Any issues concerning the client must be communicated to the Instructor as soon as possible.
- ✓ A detailed evaluation of the information system design is necessary in order for the student to complete the project successfully. This will require a significant amount of time on the instructor's part.
- ✓ Find as many ways as possible of bringing real information systems experiences into the classroom. If the instructor is not doing actual consulting work, bringing in an actual designer to talk to the class would be advised. Any way that the students can understand what really happens in the "real world" is advantageous.
- ✓ Make sure the students understand the grading rubric for the project. This could include an evaluation from the actual client. But it is essential that students know how their grade is being calculated. This protects both the student and the faculty member.

6 References

Barki, H and Hartwick, J (1994). User Participation, User Involvement, and User Attitude, *MISQ Vol. 18, No. 1 (March, 1994),* pp.55-82.

Baugh, J M (2011). Make it Relevant and They Just May Learn it, *ISEDJ 9(7), December 2011.*

Baugh J Davis G., Kovacs, P., Scarpino,J, Wood,D. (2009). Employers And Educators

Want Information Systems Graduates To Be Able To Communicate, *Issues in Information Systems, Vol X No 1*. pp 198-207, 2009

Bostrom R. and Clawson V. (1991). *Advanced Information Studies 1991*, University of Baltimore, Maryland.

Choobineh, J., & Lo, A. W. (2004). CABSYDD: Case-Based System for Database Design. *Journal of Management Information Systems, 21(3*), 281-314

Courte, J. and Bishop-Clark, C. (2005). "Bringing Industry and Educators Together," Proceedings of the 6th Conference on Information Technology Education, October 20-22, pp. 175-178.

Cyrillo, M. (2011). Lean UX: Rethink Development, *Information Week, Nov. 14, 2011*.

Deperlioglu, O., Sarpkaya, Y., & Ergun, E. (2011). Development of a Relational Database for Learning Management Systems. *Turkish Online Journal Of Educational Technology - TOJET, 10(4),* 107-120.

Foltz , C., Bryan, O'Hara, Margaret T., Wise, Harold, (2004). Standardizing the MIS course: benefits and pitfalls, *Campus-Wide Information Systems, Vol. 21 Iss: 4*, pp.163 – 169

Hunton, J E and Beeler JD (1997). Effects of User Participation in Systems Development: A Longitudinal Field Experiment, *MISQ Vol. 21, No. 4 (Dec 1997)*, pp. 359-388.

Kim, Youngbeom, Hsu, J., and Stern, M. (2006). An Update on the IS/TT Skills Gap," *Journal of Information Systems Education*, *Vol. 17*, *No. 4*, *Winter 2006*, pp. 395-402.

McNurlin, B. C. & Sprague, R. H. Jr. (2006). Information Systems Management in Practice, Seventh Edition, Pearson Prentice Hall, Upper Saddle River, NJ.

Reinicke, B. A. and Janicki, T. N (2010). Increasing Active Learning and End-Client Interactions in the Systems Analysis and Design and Capstone Course, *ISEDJ 8(40) June 30, 2010.*

Robbert, M. A., Wang, M. Guimaraes, M., Myers, M. (2000). The Database Course: What Must Be Taught, SIGCSE *Bulletin Proceedings of 31st SIGCSE Technical Symposium on Computer Science Education*, *March*, pp. 403-404

Schwalbe, K (2010). Information Technology Project Management, Course Technology, Cengage, United States.

Seyed-Abbassi, B., King, R., & Wiseman, E. (2007). The Development of a Teaching Strategy for Implementing a Real-World Business Project into Database Courses. *Journal Of Information Systems Education*, *18(3)*, 337-343.

ISSN: 2167-1435
v30 n2545

Appendix

Life Cycle Step	Activities	Work Product	reference
Scope	Define mission, goals, deliverables, constraints, responsibilities, expectations and costs	Scope Statement	
Requirements Elicitation	Exploration of problem; elicitation of one or multiple issues to address; SWOT	SWOT and issue specification	Figure 2
Functional Requirements	Identification of Process Stakeholders; for each determine what the "who" "do"	Stakeholder List; requirements expressed as a table of what "who" "do"	Figure 3
Work Flow	A table, 1 column per stakeholder transferred from the who-do table of actions; the new table is arranged so that actions appear chronologically from left to right; a final column is added to develop Procedure utilizing each action that "who" "do" in a short sentence.	The Work Flow; The procedure is used to form a story describing the business process.	Figure 4
TDLCRM	A Top Down Levelized Conceptual Relational Model; each higher entity is in 1:N with lower entities. States of an entity are shown with an underscore; controlled attributes are shown using a - - - dashed line from the upper control values to the entity. The TDLCRM is derived from the "story" output from the work flow drawing.	The TDLCRM showed top down and left to right.	Table 2 Figure 5 Figure 6
Screen Function Flow	This is a drawing derived from the TDLCRM and work-flow depicting the CRUD operations needed to build the application	The Function Flow drawing for the application	Figure 7 Figure 8 Figure 9 Figure 11
Layout	An application sketch revealing the HCI and function placement on screens.	Screen Layout	Figure 10 Figure 11
Database	A script version of the database that may be utilized to build the database	DB Script	
SPROCS	Scripts for stored procedures	DB Script	
Sample Data	Scripts for sample data for each table	DB Script	
Coded Application	The application is coded in an N-tier fashion	A Coded Project	
Testing and Acceptance	Final System Testing is performed and each path is validated with the end-user community	Testing and Acceptance	Figure 11

Table 1. Steps of the RAPID Life-Cycle-Methodology.

Figure 2. Results of the SWOT Analysis

Results from the SWOT Analysis Strengths, Weaknesses, Opportunities, and Threats were captured in a meeting. The participants all agreed that implementation of a City Warehouse System (CWS) was the item most needed
Participants Gas, water, electric, public works, inventory and purchasing managers, and the warehouse supervisor
Final Statement – a description of how the new process works The City purchases, distributes and maintains gas, water, power and materials to maintain the various city services controlled by a services manager. Each department has a manager and staff. In order to take on a job, small trucks are available to each city worker and are filled with required inventory that is replaced by a warehouseman from central warehouse stores as needed. A purchasing agent is responsible to see that inventory is maintained by purchasing needed material from vendors. Each activity of the workers is referred to as a job which tracks material and worker time.

Figure 3. Requirements Analysis

Stakeholders are those "who" are actually involved in the business process. They "do" the work called activities.

Stakeholder "who"	Activity Performed "do"	Sequence
Service Manager	Maintains person accounts Requests inventory analysis report	1 10
Purchasing	Purchases material for the warehouse Assemble vouchers for AP Closes PO	3 7 8
Warehouseman	Stocks shelf from delivery Indicates Quantity Received Supplies material to workers	5 6 8
Vendor	Supplies materials and invoice	4
Department Head	Maintains Department accounts	2
Worker	Stocks truck from warehouse Initiates and works jobs Tracks hours worked on a job Tracks inventory used on a job	6 7 8 9

Figure 4. Work Flow

Service Manager	Purchasing	Warehouse man	Vendor	Department Head	Worker	Procedure
Maintains person accounts						The Service Manager adds people to accounts and set all privileges.
				Maintains Department accounts		The department head assigns privileges to workers.
	Purchases material for the warehouse		Supplies materials and invoice			Purchasing orders material for the warehouse from a vendor who supplies the material.
		Stocks shelf from delivery Indicates Quantity Received				The warehouseman stocks the shelf from vendor delivery, receives invoices and indicates quantity received.
	Assemble vouchers for AP Closes PO	Supplies material to workers				Purchasing validates the received order quantity and closes the PO.
					Stocks truck from warehouse	The warehouseman supplies stock to the workers who add the material to their truck.
					Initiates and works jobs	The department head dispatches a new job to a worker.
					Tracks hours worked on a job Tracks inventory used on a job	The worker completes the job and records time and material spent. The worker may write a comment about a job.
Requests inventory analysis report						The service manager may run a report comparing expected inventory in the warehouse and on trucks.

The final story

The Service Manager adds people to accounts and set all privileges. The department head assigns privileges to workers. Purchasing orders material for the warehouse from a vendor who supplies the material. The warehouseman stocks the shelf fro vendor delivery, receives invoices and indicates quantity received. Purchasing validates the received order quantity and closes the PO. The warehouseman supplies stock to the workers who add the material to their truck. The department head dispatches a new job to a worker. The worker completes the job and records time and material spent. The worker may write a comment about a job. The service manager may run a report comparing expected inventory in the warehouse and on trucks.

ISSN: 2167-1435
v30 n2545

TDLCRM Guidelines:

1. Since the "mission of information systems is to help people to achieve their goals through the application of information technology" (McNurlin and Sprague 2006), the People entity is always drawn in the upper left corner of the drawing.

2. Entities that occur earlier in time are drawn to the left of those that occur later. This gives the drawing a top-down-left-to-right feel. The entities appear to flow. Since the entities were derived from the story which is a description of the system, this is a reasonable observation.

3. The relationship lines must appear to create a flow. The flow is top-down, and left to right. Therefore, relationship lines should be drawn from the upper left to the lower right. This may mean that some of the entities have to be moved around to enable this to happen.

4. If a table has a status variable, the **entity** is underlined. This implies that a status table will be created, as well as a status change history table (date-time, person, old status, new status, reason for change. State machine processing may be involved with the changing of status values.

5. If an entity can exist in multiple versions, then the **| entity** is underlined and a preceding vertical bar is used. Older version records still exist as records, but they are inactive. Only the current version maintains an "active" status.

6. If an entity is associated with a financial transaction, then the **entity**, is dashed-underlined. A transaction history table is created (date-time, person, transaction amount for each financial variable—the sum of the financial variables is equal to the amount shown in the entity).

7. Controlled attributes may be shown (or not). A controlled attribute is always in 1:n with the entity. In an editor, the controlled is selected from a drop-down list. Controlled attributes are pictured with dashed line relationships in the TDLCRM. The list of allowed values is the 1 side and the entity is the N side

8. Any intersecting entities must be explored for a potential composite key to establish uniqueness. An " * " should be placed in front of the entity with a composite key.

Table 2. Rules for Construction of a TDLCRM (Top Down Levelized Conceptual Relational Model)

ISSN: 2167-1435
v30 n2545

Figure 5. Deriving the Database from the Story, The TDLCRM

Color – Meaning	Description
Nouns	Nouns usually describe entities in the database. They may also describe a report name or an attribute.
Report Name	Report name is something that will be printed out or displayed on a screen.
Attributes	Attributes are clearly nouns that relate to the component of an entity. A careful inspection of the table has to be done to detect attributes.
Verb phrases	Verb phrases may contain a verb or prepositional phrase such as "sale has items".
States	State describes status. For example, in progress or complete.
Roles	Roles describe the functional name of a person carrying out a task, e.g. manager, cashier, customer. They appear to be nouns but are differentiated from entities because of their responsibilities in the system.

The Final Story

The Service Manager adds people to accounts and set all privileges. The department head assigns privileges to workers. Purchasing orders material for the warehouse from a vendor who supplies the material. The warehouseman stocks the shelf fro vendor delivery, receives invoices and indicates quantity received. Purchasing validates the received order quantity and closes the PO. The warehouseman checks-out stock to the workers who add the material to their truck. The department head dispatches a new job to a worker. The worker completes the job and records time and material spent. The worker may write a comment about a job. The service manager may run a report comparing expected inventory in the warehouse and on trucks.

The TDLCRM

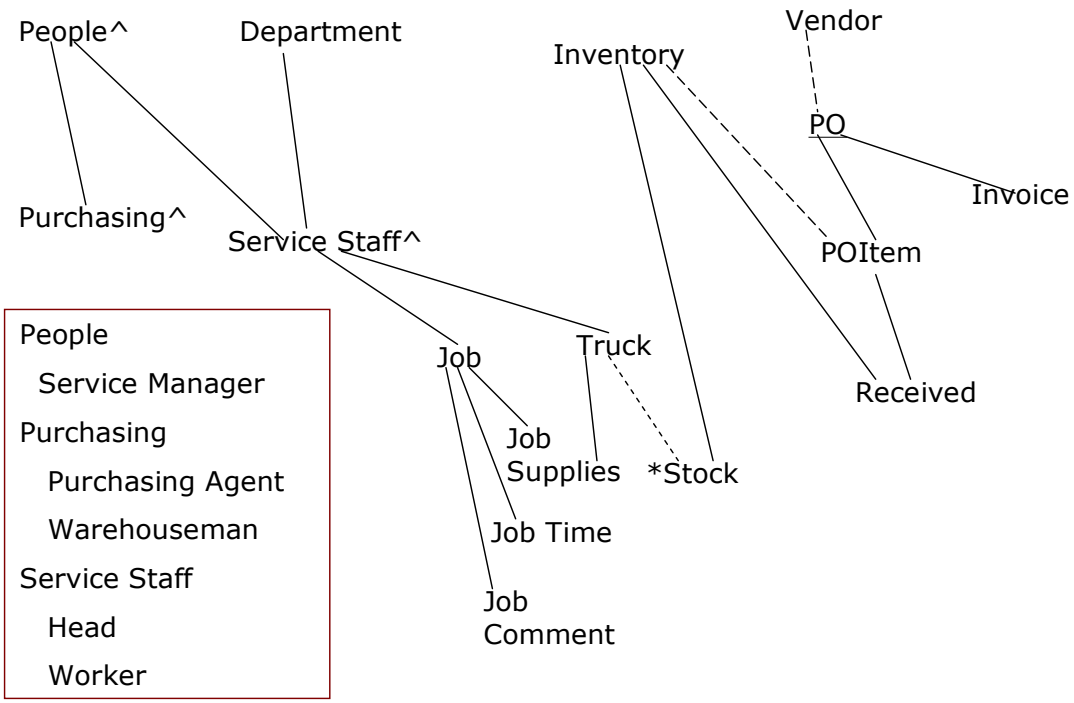


Figure 6. TDLCRM Conventions

Capability Security
If an end-user is to be granted access to various portions of the system, this can be indicated in the TDLCRM by placing an ^ symbol at the end of the entity name:

Person^

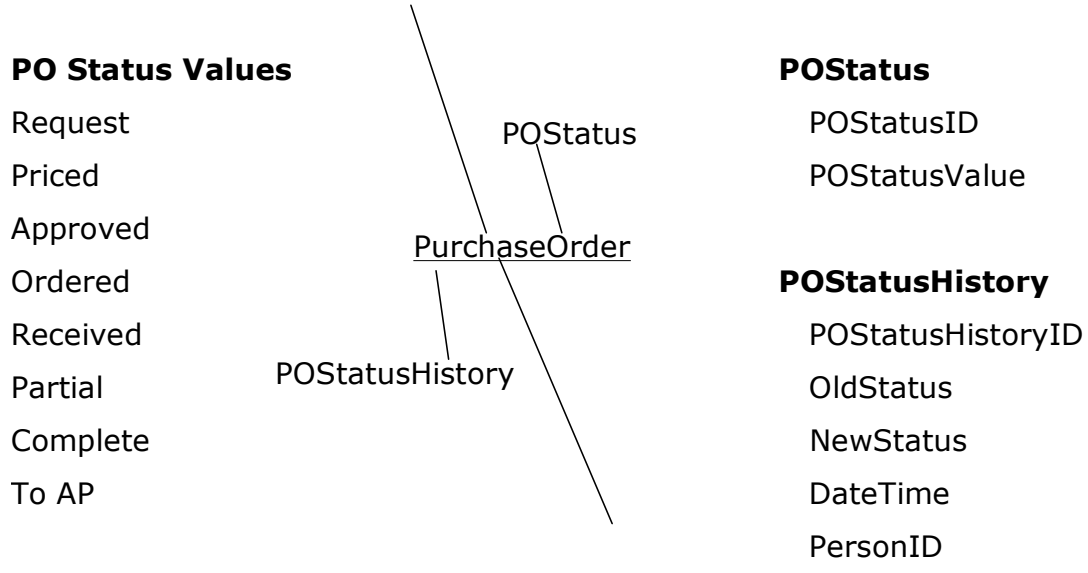
The implication in the database is that Booleans may be used to indicate the nature of the security. These Booleans may be utilized within the application to control button visibility:

Person Table
Person ID
Last
First
Other fields
isSA
isDepartmentHead
isPurchasingAgent

Entity Status
An entity may be found to have an associated status. This is depicted in the TDLCRM by underlining the given entity:

PurchaseOrder

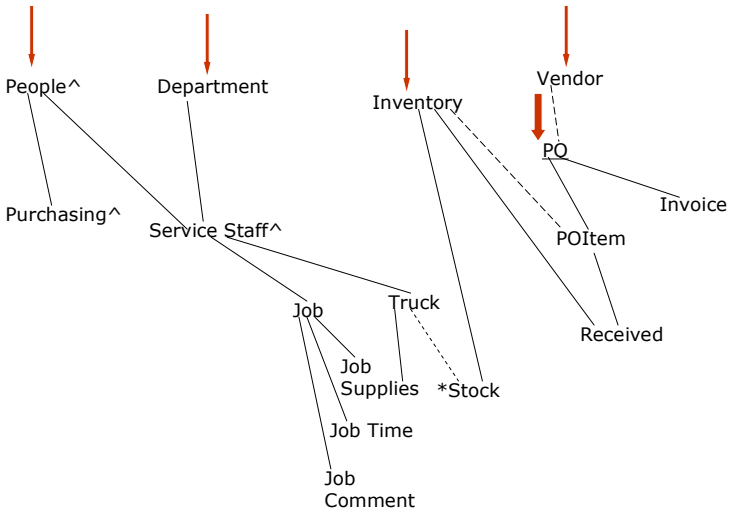
The meaning of the status is there is a status as well as a status history table associated with the named entity as shown in this TDLCRM. There is no need to show these additional tables in the main TDLCRM since they are assumed because of the underscore. Both OldStatus and NewStatus are not pointers to current status values; rather, they are the values in place at the date and time of the transaction.



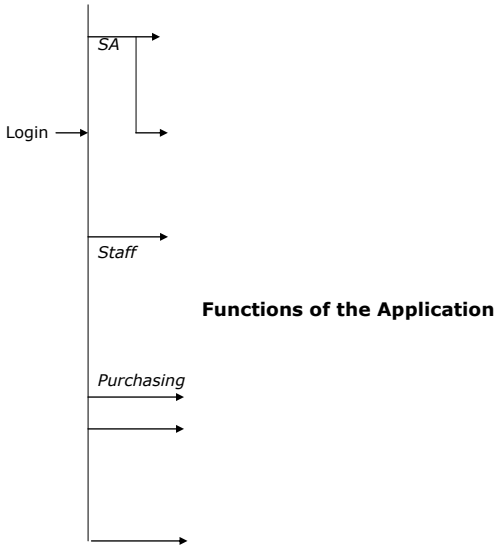
ISSN: 2167-1435
v30 n2545

Figure 7. Entry Points and The Main Menu

Determination of the Main Menu
Level 1 entities are a possible entrance point from the main menu. If a level 1 entity is a controlled attribute, then the next level 2 entity may be the true entry point. Red Arrows represent entry points.



The Main Menu
The main menu distributes control to parts of the application upon end-user desire or permissions. The login, based on data in the person table is responsible for access to the system. The ^ symbol indicates permissions are stored in the entity which can be used to control application flow by controlling visibility of buttons.



Functions of the Application

SA – system Administrator

Staff – services staff

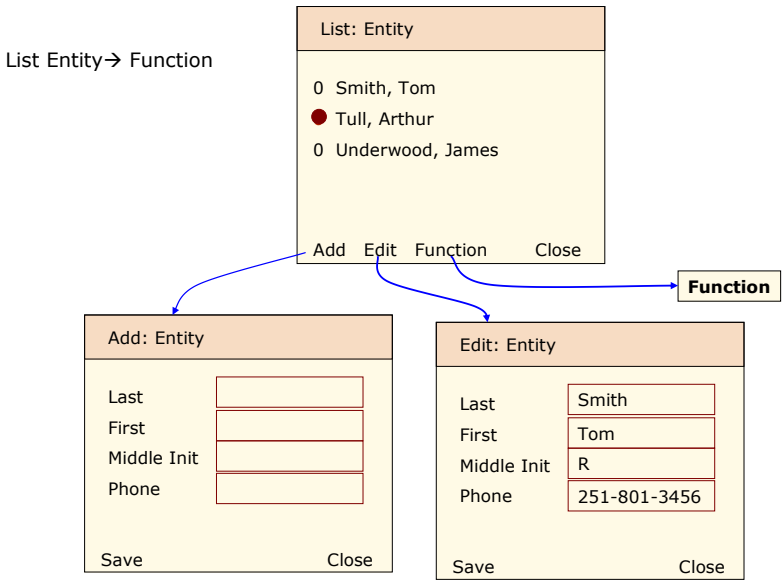
Purchasing – purchasing department members

Note: SVtable is initiated at Login, and is passed with each control flow

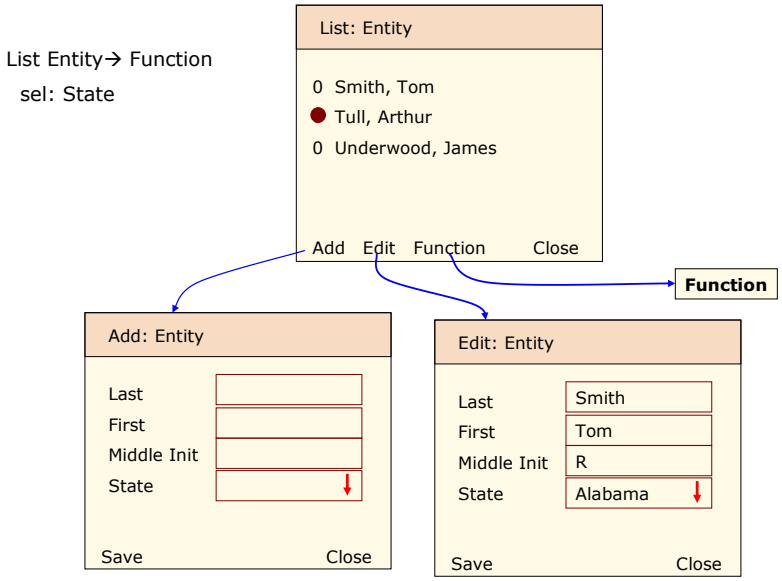
ISSN: 2167-1435
v30 n2545

Figure 8. Conventions for Screen Flow

The text **List Entity →** is the equivalent of the 3 pages shown below, the difference being that it takes far less time to write the phrase than it takes to draw out the entire design. Also, focusing on the attributes during the initial phases of design enables the designer to loose focus on the overall design.

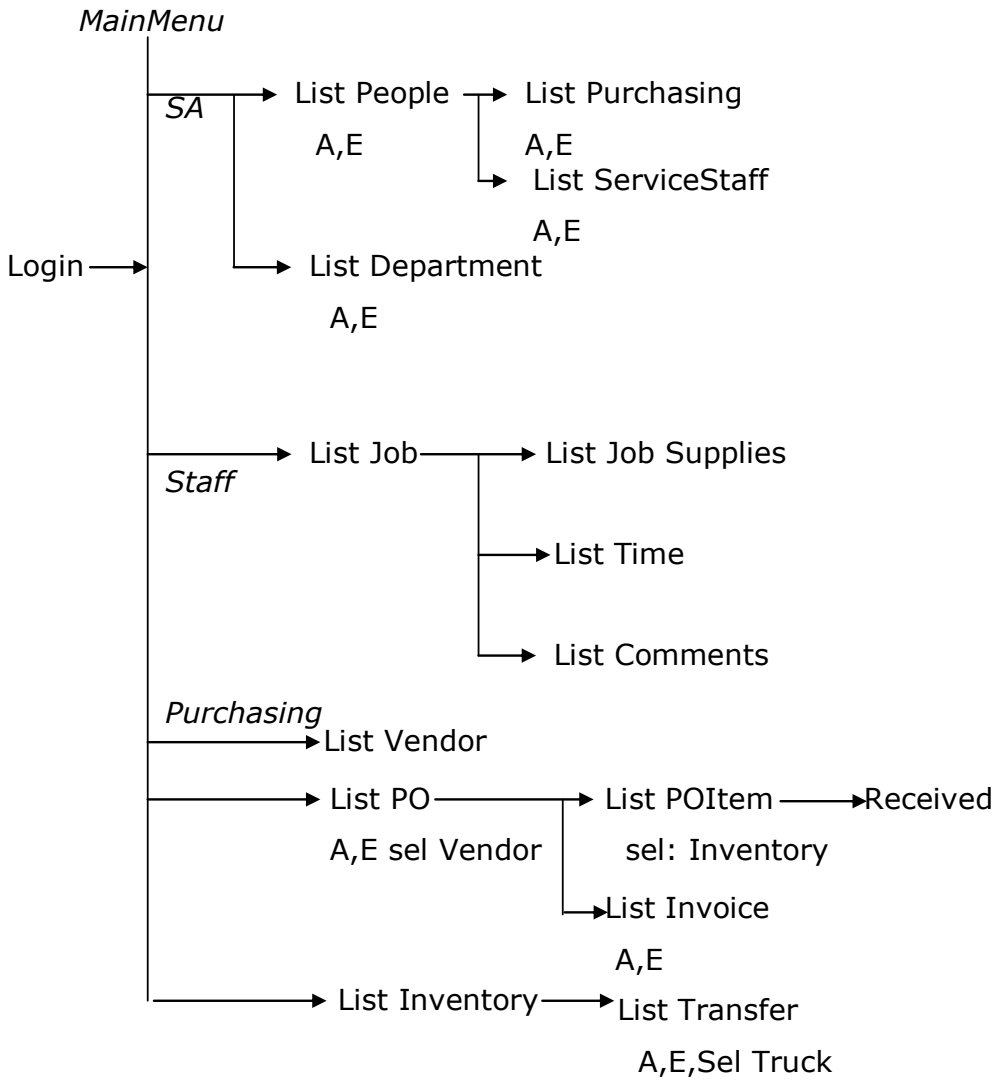


If the letters **sel: State** are used, this implies that State is a controlled attribute and should be used as a drop down list.



ISSN: 2167-1435
v30 n2545

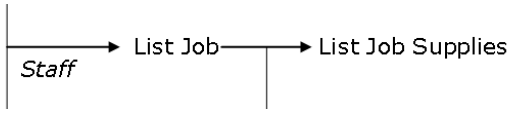
Figure 9. Function Flow



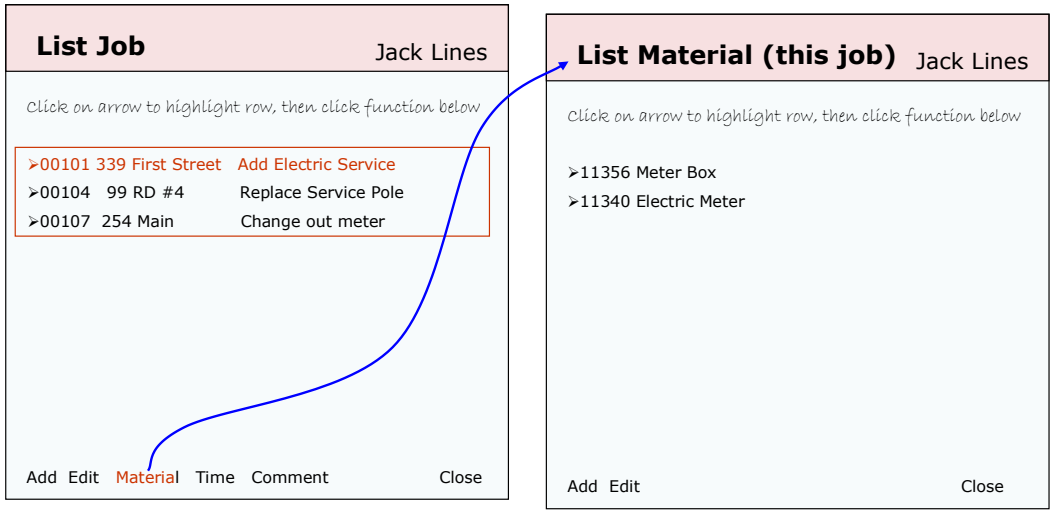
ISSN: 2167-1435
v30 n2545

Figure 10. Partial Layout of Function Flow

The portion of this function flow is used to draw the screens listed below:



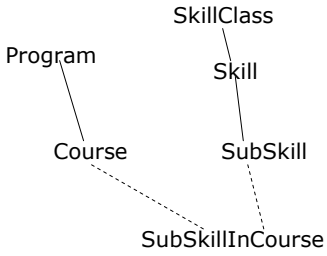
The resulting layout:



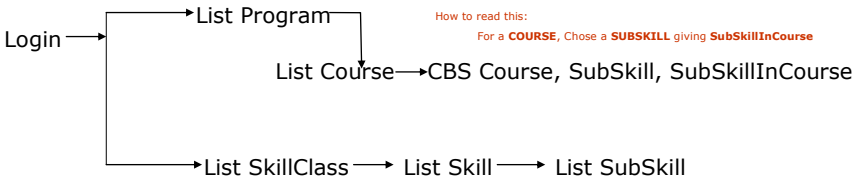
ISSN: 2167-1435
v30 n2545

Figure 11. Defining and Using the Check-Box-Select (CBS) Routine

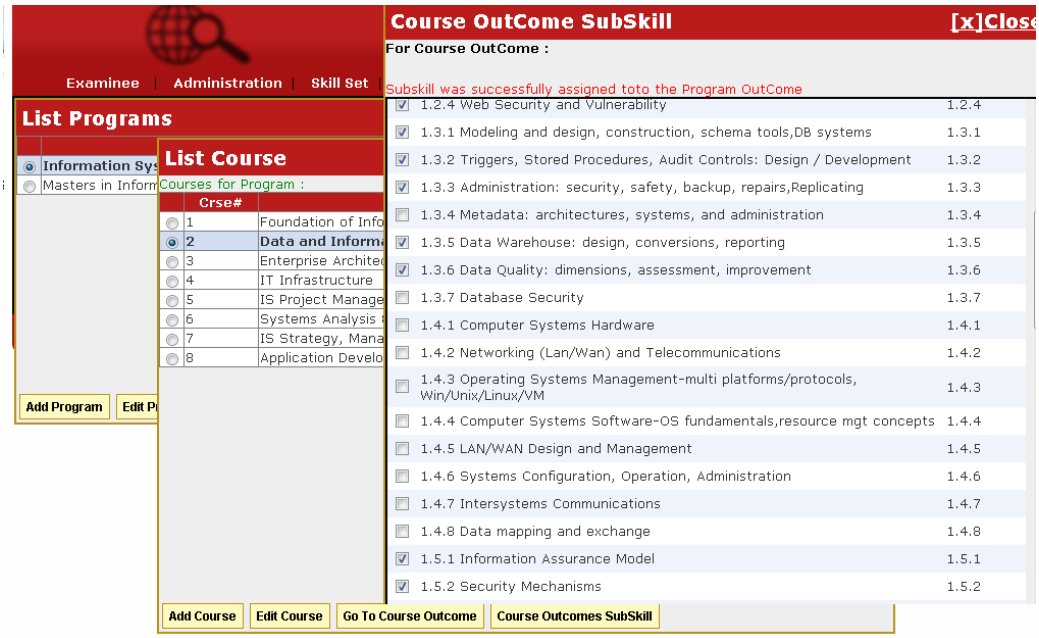
Occasionally two vertical streams of entities will meet in what is traditionally called an intersection table. Although this type of entity is also called an intersection in our experience the table contains attributes other than the required primary key and two foreign keys. For example, skill depth, hours spent on the skill in class, number of exam questions given on this skill and other necessary data.



The function flow for this application would appear as follows:



The layout involving the check-box-select is shown below. There is no save function since update to the database is shown immediately by using AJAX:



ISSN: 2167-1435
v30 n2545

Figure 12. A Student Review of the Methodology

567-8 Review…….

*Instructions: Please enter your answers in **bold text**. Thank you.*

Were you able to understand and use the steps in the life cycle as presented in the templates? Explain:

Yes, I understood all the steps in life cycle as presented in the template. The steps Scope, Story, Work-Flow, TDLCRM, Screens Sketch, Screens Layout, and Screen Implementation Planning were used in development of the system. This organized and stage wise development approach was helpful in proper understanding, design and implementation of the system. Also by utilizing this approach it was easy to implement work breakdown structure among different teams.

In the Scope template, explain what is meant by the customer focused mission statement:

The customer focused mission statement is aligned to customer's organizational beliefs and values. The products and services to be provided are tailored to customer's expectations and satisfaction.

I was database administrator and also took the responsibility of leading the design efforts for the sub-system that included examtype, ExK1, ExK2, ExK3, Key1, Key2 and Key3 screens. My customers for database included Dr. Longenecker, Greg, my team, Kelly's team and Ali's team; I was persistently in communication with all the teams about the database requirements and fulfilled their expectations. To maximize the satisfaction of the customers, I used to take feedbacks from the customer and enhance the database if required. Eg. My instructor wanted DOL 2011 data from the website in the tables in a particular format, I closely worked with through multiple meetings and feedbacks to fulfill customer expectations. Similarly for the Examtype system my design team and I worked closely with My instructor to achieve the customer's requirements.

Comment on your ability to use the requirements capturing tool ("who-do"):

The requirements capturing tool ("who-do") was helpful in developing work-breakdown structure that is dividing the requirements and keep the track of the progress.

After understanding the requirements of the database and Examtype system, we divided the total work by "who do" chart. I took major responsibility of the database creation, populating tables with valid data and Guru was the backup support for the database. For examtype system, I divided the requirements into simple tasks and designed all the screen sketches and layout, did screen implementation planning and developed few SPROCs. In "who do" list Siddartha was responsible for coding of examtype, key1, key2 and key3 screens which included checkbox select and edit screens. And Guru was responsible for coding the screens for Exk1, Exk2 and Exk3 and providing support to Siddartha. Dividing the work by "who do" list result in smooth execution and complete implementation of the requirements as per customer requirements and satisfaction.

Comment on the mechanism of conversion of the functional requirements into work-flow:

The different stages in conversion of the functional requirements into work flow includes System description like who are the stakeholders and what do they do, Story building, and translation of story into workflow what are the procedures that stakeholders perform to accomplish their tasks.

Comment on the concept of translating the work-flow and requirements into the TDLCRM:

ISSN: 2167-1435
v30 n2545

The workflow and requirements are translated into TDLCRM. In the conversion of story and workflow into TDLCRM the nouns are represented as entities. The characteristics of the entities are represented by attributes of the corresponding entities. A status may be associated with an entity, e.g. completed or in progress and the entities that have status is represented by underline. The relationships between the entities are determined by verb phrases. The entities that occur in time ahead are drawn left of the entities that occur later, thus the story gives the flow of the data from top to bottom-left to right.

Since you have now used the TDLCRM, state at least 3 positive and 3 negatives about the TDLCRM:

3 positives:
1) Easy to recognize entities and attributes.
2) Easy to identify which entities are associated with status and permits.
3) Easy to develop physical database.

3 negatives:
1) Seems complex for new developer to convert story to TDLCRM
2) Learning curve associated.
3) Unconventional to the database developers who are used to ERDs.

Explain the difficulty of conversion of the TDLCRM into a physical database:

A learning curve is associated with the TDLCRM to understand and adopt this convention. Once you understand TDLCRM, conversion from TDLCRM to physical database is easy after that.

Name the primary elements (screen sketch tool names) to express the screen function flow:

The primary elements or screen sketch tools that we used were paper, pencil to develop the screen function flow and scanner to share with team members.

Explain conversion of a TDLCRM into a screen flow:

TDLCRM is obtained from story and the workflow of the system. TDLCRM is converted into derived story, the entities represent nouns in the system and workflow determines the various functions to be implemented. Draw the sketch of screen functions following the login screen until each table in the database has been edited.

Explain how capability security may be presented in a screen flow:

The capability security may be presented in a screen flow by controlling function access to various roles and privileges associated with the roles through menu. The capability security represents who can access each function.

Explain conversion of the screen flow into a screen layout:

The screen flow is converted into a screen layout by creating a network of screens for each function in the screen flow.

Explain the utilization of in-line-frames in systems development:

An IFrame (Inline Frame) is an HTML document embedded inside another HTML document on a website. The IFrame HTML element is often used to insert content from another source into a Web page. An IFrame behaves like an inline image; it can be configured with its own scrollbar independent of the surrounding page's scrollbar.

What is CSS and how is it implemented in screen construction?

ISSN: 2167-1435
v30 n2545

CSS stands for Cascading Style sheets. Style sheets define how a HTML element is displayed. Styles are saved in external .css files and enables you change the appearance and layout of the HTML page. An external style sheet is best option when you want to apply style to many pages. The external style sheet is linked to a html page through link tag.

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

And external CSS page can be written in any text editor and is saved with .css extension.

Explain the need for SVTABLE, and the mechanism of its implementation:

The SVTABLE is needed as web-server does not know information from one webpage to another because the HTTP request does not maintain the state.

This problem is solved in ASP through creating cookies or session variables. The session variable stores information about user id, preferences and holds information for a user and pass it through all the pages. SVTABLE stands for Session Variable Table and in our project contains ids and privilege information for the users. This information is passed from one page to another for a user and depending on that we can give users security capability.

Discuss conversion of a layout drawing to a working physical screen system. Briefly discuss realistic mechanisms for developing a multi-tier architecture.

Once screen layouts and flow are finalized, the conversion to working physical screen system is extremely clear and involves majorly coding. In Screen Implementation planning one can decide what HTML elements to be used to implement various functions.

The application implemented in multi-tier architecture is high quality coding style as it is easy to develop, read, maintain, debug and test. Mainly four tiers are implemented data tier contains tables and stored procedure, data access tier contains function to access data tier, business logic tier implements behavior and activities of the organization and last tier is presentation tier that implement user interface functions.