Critical Thinking and Unit Analysis Examples

Critical thinking is thinking about thinking. CT is one of the top most skills requested by employers. There are many different versions of critical thinking available. I use the methods outlined in the works of Richard Paul and others. Please read Content is Thinking; Thinking is Content by Richard Paul. To be an IT professional, one needs to *think like an IT professional*. It is *extremely* easy in the IT field to focus on product, because that is what we use, and not on process. Process is thinking about why things work (education), not so much on how to make it work (training). Please review The Elements of Critical Thinking based on Richard Paul's work. Paul identifies eight elements in CT. For my classes, I focus on purpose, concepts, and implications. I feel these three are the most crucial elements of CT while learning about computer technology.

Unit Analysis as Critical Thinking:

The Unit Analysis will only be using three critical thinking elements from Paul's list. Often in project management, you will be asked things like, "What is the purpose or why do we need this?", "How is this different from what we are doing now?", and "What will happen if we do this and if we don't?"... CT can help you develop answers to these types of questions when one is presenting a resource intensive project like the installation of a new network.

Purpose...

- Take time to state your purpose clearly.
- Distinguish your purpose from related purposes.
- Check periodically to be sure you are still on target.
- Choose significant and realistic purposes.

Concepts...

- Identify key concepts and explain them clearly.
- Consider alternative concepts or alternative definitions to concepts.
- Make sure you are using concepts with care and precision.

What concepts are not: products or elements of products (like ChartWizard or Windows), task based events (like how to format a paragraph), judgments (like "Linux is the best operating system"), and alike. Concepts just focus on the core idea. They are ideas that everything else is based on. They are neutral.

Implications...

- Trace the implications and consequences that follow from your reasoning.
- Search for negative as well as positive implications.

- Consider all possible consequences.

Developing a Unit Analysis based on Critical Thinking:

The example below is about making vegetable soup. Imagine you are taking a cooking class and the current topic is soup. The instructor has presented the unit outcomes, objectives, and projects to assist you with *thinking* about soup. At the end of the unit, you are asked to demonstrate critical thinking. Here are two examples.

A Bad Example of Critical Thinking:

Purpose

To make vegetable soup (Product focused)

Concepts

SOUP: A food made primarily with water (A good start!)

PREPARING THE INGREDIENTS: Washing and cutting the vegetable (This is a task, something you do, ingredients is the concept)

CARROTS: A root food or vegetable (Shouldn't vegetable be the concept?)

SALT: Sodium chloride (What new information was given? They are the same thing!)

HISTORY OF SOUP: People around the world have been making soup for thousands of years (This is a statement, soup is the concept)

Implication

By following directions, you can make a good soup which can be salted to taste. (Would that not be true of most things?)

A Better Example of Critical Thinking:

Purpose

Learn the basics of making vegetable soup. How food can be transformed during cooking and complement tastes with various spices. The use of spices can make a subtle change or significant change to the taste experience depending on the amount and choice of spices used. (Process focused)

Concepts

SOUP: A liquid food prepared from meat, fish, or vegetable stock combined with various other ingredients and often containing solid pieces

VEGETABLE: The edible part of a plant, such as the root of the beet, the leaf of spinach, or the flower buds of broccoli or cauliflower

SPICE: Any of various pungent, aromatic plant substances, such as cinnamon or nutmeg, used to flavor foods or beverages

COOKING: To prepare or treat food by heating for eating

TASTE: The sense that distinguishes the sweet, sour, salty, and bitter qualities of dissolved substances in contact with the taste buds on the tongue

Implication

Almost any vegetable can be cooked and used in a soup. During the cooking event, the vegetable will often go through a transformation of texture and sometimes taste. The key is to understand how each vegetable is transformed during cooking. All the vegetables together create a complex mix of textures and tastes that can be very pleasing to the pallet. By using various spices with various vegetables, the taste of the soup can be enhances or altered. The most basic spice is salt but some soups may use as many as 40 spices to give the soup its special flavor and taste texture that makes the dish what it is, like a spicy gumbo.

Good soup is about balance between taste of the vegetables, overall texture created during cooking and spices selected to modify the taste. Making good vegetable soup is an art but easily mastered with experience.

Student Examples of Critical Thinking (Used by permission)>

Purpose

The purpose of Unit 1 is to provide the basic concepts of the relational database in the framework of the SQL. The parts of the database consist of tables, where the table rows of data records are organized under columns of field headings representing those attributes in cells that describe the record contents. The Lunches database is briefly described along with design decisions of table structure, field data types and primary keys.

Concepts

TABLES: database structures for holding rows of records as data within field names as columns.

CELLS: the smallest part of a table containing a single piece of data where the row and column meet.

PRIMARY KEY: one or more columns (fields) of a table that uniquely identify each row (record).

ATOMIC: the idea that the data in a cell is a 'single indivisible unit'.

NULL: an unknown value in a cell of a table that is not permitted in the primary key column.

Implications

With SQL widely used in an organization today in extracting information from the data stored in the database tables, the simplicity of this language makes it relatively easy for even the average employee to gather the information needed for their provision of services to internal and external customers. The IT personnel should also benefit from the ease of using SQL in the organization since they should be able to spend more time on non SQL issues, not having to assist as much with SQL training or maintenance issues.

Purpose

For this first unit, we are studying the basics of operating systems. We are taking a look at the tasks of an operating system, the history, and the basic family of operating systems. We are also taking a look at UNIX.

Concepts

PROCESSOR MANAGEMENT: ensures that each process and application receives enough of the processor's time to function properly. Processor management also uses as many processor cycles for real work as possible.

MEMORY AND STORAGE: management makes sure that each process has enough memory in which to execute, and it can neither run into the memory space of another process or be run into by another process. The first task of memory and storage management requires the OS (operating system) to set up memory boundaries for types of software and for individual applications.

DEVICE MANAGEMENT: uses drivers to translate and organize bits of information so it can be sent to specific locations on storage devices.

THE APPLICATION INTERFACE: lets application programmers use functions of the computer and OS without having to directly keep track of all the details in the central processing unit's operation.

THE USER INTERFACE: basically brings structure to the interaction between a user and the computer.

Implications

An operating system provides users the necessary tools to more effectively use the computer. However, if the user does not have proper knowledge of the operating system, then problems can occur. That's why "help" features are an important part of OS. However, it is not always the user who contributes to the errors. Computers placed in a "bad" location can result in weak signals to and from other computers, resulting in a loss or incomplete data transfer. With the rise of users becoming efficient computer operating systems, it is probable that OSs will continue to improve and get better.

Purpose

We are studying this material in order to learn about SQL and how it gets information from relational databases, how to handle that information as well as learn what relational databases actually are.

Concepts

CELL: Is the smallest part of data and can't be broken down in to smaller parts.

COLUMNS: This is a collection of cells of the same type of information.

ROWS: A collection of cells that are related together in that they represent information about the same object or event.

PRIMARY KEYS: They identify a row something like a name. All rows must have a unique primary key. They consist of one of more columns, and if it contains more than one column, then it's called a composite primary key. The primary key makes it easier and faster to retrieve a row from the table.

TABLE: Is a collection of rows. They usually contain many rows which makes them tall and thin. In other words, they can contain thousands or rows, but the rows may only contain a dozen columns. All the rows are named by primary key.

RELATIONAL INTEGRITY: Where a collection of tables are related to each other by a primary key of one table to another table. This means that in order for a related record from a given table to exist, it must know primary key from a table that is trying to relate to. Otherwise, its existence will not be allowed by the database.

DATABASE: A collection of related tables with linked by rules of relational integrity to ensure non-sense data can never reside within the collection of data. The database controls the rules of

integrity between the tables and enforces that related information must be valid before it can exist in a given table.

Implications

This technology can enhance the organization software products by creating a collection of SQL queries that follow the standard business of the organization. These collections can be presented to the end-users in the format similar FAQ (Frequently Asked Questions). The SQL objects collection can be given to the user in such a way that they could use common English and get powerful and complete sets of information in return. The price would be to collect the business model of FAQ's, create the collection in SQL and present it through a higher tier.

Purpose

The purpose of studying this material is to learn the fundamentals of canvas and window objects, as well as learning a variety of methods for displaying them. So far, the applications we have developed have contained only one window and one canvas. The window (WINDOW 1) was created by default when a new form was created, and the canvas was created by the Layout Wizard and then it was automatically assigned to WINDOW 1. As the forms we develop get more complex, we will want to add additional windows and canvases then display them in response to interface or internal processing events.

We are expected to know the different styles of windows in Forms: document and dialog, as well as gaining an understanding of the behaviors and properties of (MDI) Multiple Document Interface, which is only available on Window's platforms. We are also expected to understand that window objects have properties that determine their size and position, specifically under the Functional property category; they have a series of window-specific properties that determine whether or not the window can be minimized, resized, or closed. There are also a variety of methods for displaying and hiding windows at runtime, we are expected to be familiar with each of these methods. Finally, by studying this material we are expected to have the ability to create multiple canvases: content, stacked and toolbar, as well as understanding their viewports and learning the rules for displaying and hiding them.

Concepts

WINDOWS: Like any other physical object in Forms, windows have properties that determine their size and position. Under the Functional property category, they have a series of window-specific properties that determine whether or not the window can be minimized, resized closed, and so on. Windows also have a modal property that helps determine how and when the user can leave the window. Navigation cannot leave a modal window until the user has completed whatever task the window calls for, they force the user to respond to a message or task before moving on. There are two styles of windows in Forms: document and dialog.

DOCUMENT WINDOWS: Document windows are commonly used for data entry and query screens; they are the main windows in an application. These windows are completely contained by their parent MDI window. If a user were to drag the document window as far to the left as possible, it would disappear under the boundary of the MDI window. Document windows are usually modeless (Modal property set to No). It is common to build forms that have multiple document windows. Theses windows can be configured to be open simultaneously so that users can toggle back and forth between them.

MULTIPLE DOCUMENT INTERFACE (MDI) WINDOW: On Microsoft Windows operating systems, there is a third style of window called the MDI. The MDI window serves as a parent window to all of the other windows in a form. The MDI window in not visible in the Object Navigator, nor are its properties visible in the Property Palette. However it does have properties and they can be adjusted programmatically using the SET_WINDOW_PROPERTY built-in.

MODAL PROPERTY: Modal property is a Functional property of Windows objects that allow values of Yes or No. Modeless windows (Setting property to No) allow users to work on two or more tasks simultaneously. Modal windows (Setting property to Yes) require some sort or response form the user before regular processing can continue.

CANVAS VIEWPORTS (VIEWS): The viewport refers to the area of the canvas that is visible to the user. The canvas object itself is not always entirely visible; only the area defined by its viewport is visible. It is common to place some items outside the viewport so that they are not initially visible. Then as the user scrolls or navigates the form, the viewport can move to expose different parts of the canvas. A canvas can be displayed either by using built-ins to open it explicitly or by navigation to an item on the canvas.

CONTENT CANVASES: Content canvases are the most common canvas type because every window must have one as its main canvas, they are used to display and position the items in your form. While content canvases have viewports, thy do not have properties to set the size of their viewports. The size of a content canvas' viewport is the same as the size of the window that it occupies. To set the size of a content canvas' viewport, you must either: 1) use the Height and Width properties of the window it is assigned to or 2) adjust it visually in the Layout Editor.

STACKED CANVASES: Stacked canvases are never the sole canvas in a window. They are always stacked on top of other canvases, and partially or completely obscure those canvases when displayed at run-time. To stack a stacked canvas, you must position it relative to the content canvas that it is stacked upon.

Implications

Forms that have only one window and one canvas can suffice for some applications, but sometimes they are simply just not adequate to display the complexities of the more intricate applications. When this is the case, having the ability of adding additional windows and canvases and displaying them in response to interface or internal processing events are necessary. With Oracle's Forms Builder, there are numerous combinations of property settings and programmatic or navigational methods that affect the opening and closing of windows and their canvases. The

application developers must understand the use of multiple windows and canvases that aid in presenting information to the user.

These Forms objects, windows and canvases, give the developer the ability to create forms that the end user is able to multitask with; this ultimately models the real world. By offering the user the opportunity to navigate between multiple windows allows them to view all pertinent information at once. The Key, to developing windows and canvases, is knowing which of the numerous combinations of property settings and programmatic or navigational methods to implement in achieving the desired results of opening and closing of windows and their canvases. Designing these objects is a balance of understanding when and where they are needed in the form, as well as knowing how to display and hide them at runtime. By implementing multiple document windows and canvases into forms applications, the windows can be configured to be opened simultaneously so that the users can toggle back and forth between them, just like in MS Word and other typical windowing applications.