Identifying the problem-The Computational Approach

When faced with the task given to me by my client, I believe that the computational approach will be the most effective method of completing this task efficiently and professionally. This approach has been picked after fleshing out all the possible methods to face this task with, each one has been judged and weighted with myself, the task and my client's needs in mind.

Although I have already stated that the computational approach will be my method of solving the problem, there are other types of approach, each one different from the other. I believe that all the approaches can be split into three main groups:

- Computational
- Theoretical
- Practical/ Physical

The computational approach is an approach that is based on the idea of applying the Principals of Computing to any task:

-Communication- Reliably moving information between locations.

-Computation-What can and cannot be computed.

-Recollection-Representing, storing, and retrieving information from media.

-Coordination- Effectively using many autonomous computing agents

-Evaluation- Measuring whether systems produce intended computations

-Design- Structuring software systems for reliability and dependability

[List paraphrased from "Great Principles of Computing" by Denning, Peter J. Martell, Craig H.]

By applying these principles to any task for problem, not only would you be able to develop an algorithm for the task, but you can validate it, allowing that solution to be applied to that task over and over again. Another major benefit is that it then becomes possible to program a computer to solve the task.

The theoretical approach is less firmly structured than the computational approach. While logical thinking is a main trait they both share (one can make many links with computational and logical thinking, it is safe to see computational as a special kind of logical thinking).

Unlike computational thinking, theoretical thinking has an adaptable freedom that allows it near infinite abstraction and the ability make assumptions that allow this approach to tackle problems that are impossible, not only by the standards of the other two approaches, but by the limits of human abilities as well (e.g. science, wealth, time, etc.)

The method of the approach is simply to break down the task before attempting to complete it. After a sufficient amount of abstraction has been applied, the thinker will then proceed to try to separate the problem into logical steps. This method becomes more useful with an increase in

people participating, as more thinkers allow a larger scope. This can lead not only to faster ideas, but can also give a selection of valid solutions that can be analysed to get the best solution for the task. This can then be followed through to complete the task.

Finally, there is the **practical approach**. This approach may have the smallest scope of all, as it can only deal with tangible problems. While often seen as the simplest approach, the practical approach may require the most skill and knowledge to be successful, while bearing the greatest results.

The approach itself is hands on, rather than step away from the task and break it down, the first step is to start the task right away. This should not be confused with mindless fiddling, as each action should be broken down in steps.

It is these steps that define the success of the practical approach. After the completion of every step- regardless if it actually help to further you your goal or not- you are faced with another unique task. It is due to the step-by-step nature of the practical approach that gives it the highest skill requirement. To be affective, continuous steps are taken until a situation that is familiar is approached; this will then allow you dictate all the following steps into a path that is familiar to you.

At this point in reading, it should have become clear that the computational approach shares many key features with the other approaches. One could go far enough to state that it could be a hybrid of both. There is that aspect of abstraction that can be traced back to the theoretical thinking. Rather than breaking down and putting a form that is easy for humans to understand, one would have to break the task down and put it in the predetermined form that can be understood by computers.

This leads to the common ground between the algorithms of a computer and the step-by-step process of someone taking a practical approach. A computer can’t be handed a task and then do it. It has to be given the instruction broken down step-by-step.