

# A Brief Introduction To “Soft Systems” Thinking

Susan Gasson, College of Computing & Informatics, Drexel University

## The Need For A Systemic View

Typically, systems requirements analysis fails because of two reasons:

1. The analysis is not sufficiently systemic – it reduces a complex system down to a subset of activities and work-goals that can be understood.
2. The analysis is too focused on what can be computerized. It does not analyze what needs to change, but what the IT system should do to support an [implicit] set of changes.

*Why are systems changes counter-intuitive?*

The human mind is not adapted to understand the consequences of a complex mental model of how things work. There are internal contradictions between future structures and future consequences, that become difficult to balance when multiple mental models are involved. An example of counter-intuitive systems is shown in the “systemic” view of a limited social welfare system, given in Figure 1.

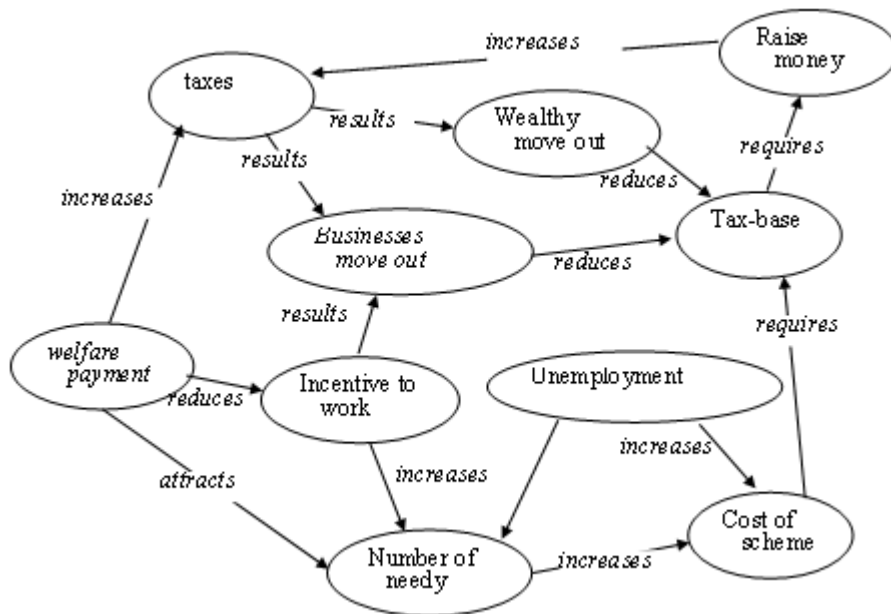


Figure 1. A Systemic View of A Limited Social Welfare System (Gharajedaghi, 1999, pg 49)

Unless a complete view of the whole system of interrelated processes is obtained, well-intentioned changes have unintended consequences. It is necessary to understand the system, rather than the processes, to understand these “vicious circles” of cause-and-effect.

## THE METHOD: SOFT SYSTEMS ANALYSIS

**Soft systems** are systems of human activity, that are organized to achieve some purpose between groups of people acting in concert. They may be supported by computer-based systems of information technology.

Soft systems analysis focuses on how the current system of purposeful human-activity needs to change and why, then breaks this complex analysis down into subsets of change that can be supported by IT systems redefinition. To achieve this, we employ a strategy of “divide and conquer”: we analyze the interrelated systems of human activity that are involved in the area that we wish to improve, then determine change priorities, then define sets of changes to the supporting IT systems. The real difference in perspective is that we need to deal with an increase in complexity to disentangle subsystems that we can improve. The method is shown in Figure 2.

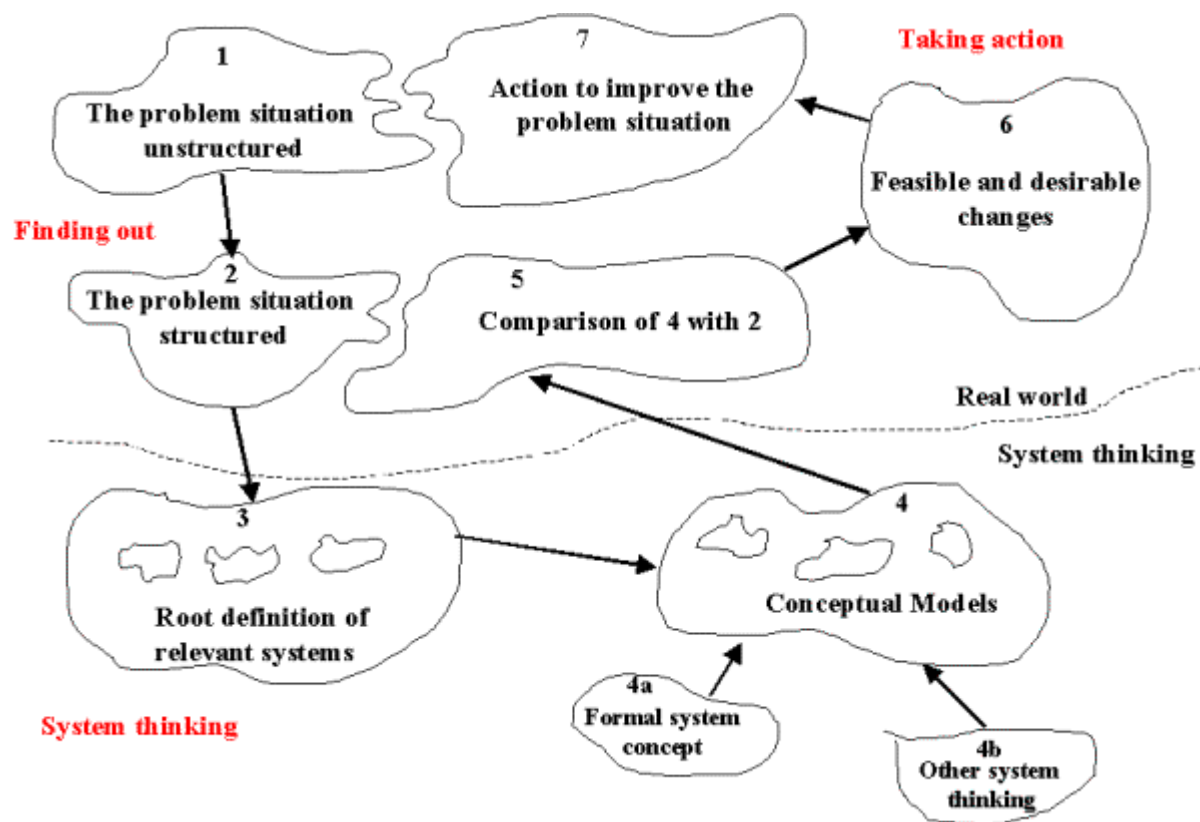


Figure 2. The Steps of Soft Systems Analysis (Checkland, 1981)

### Step 1: Understand the Problem Situation

The most critical parts of this step are (i) understanding the relationships, elements, and main activities involved in a problem situation and (ii) deciding a boundary within which the situation can be improved. A rich picture is often drawn, to understand the situation, as shown in Figure 3. The point of this diagram is to understand the situation, without trying to structure or analyze it.

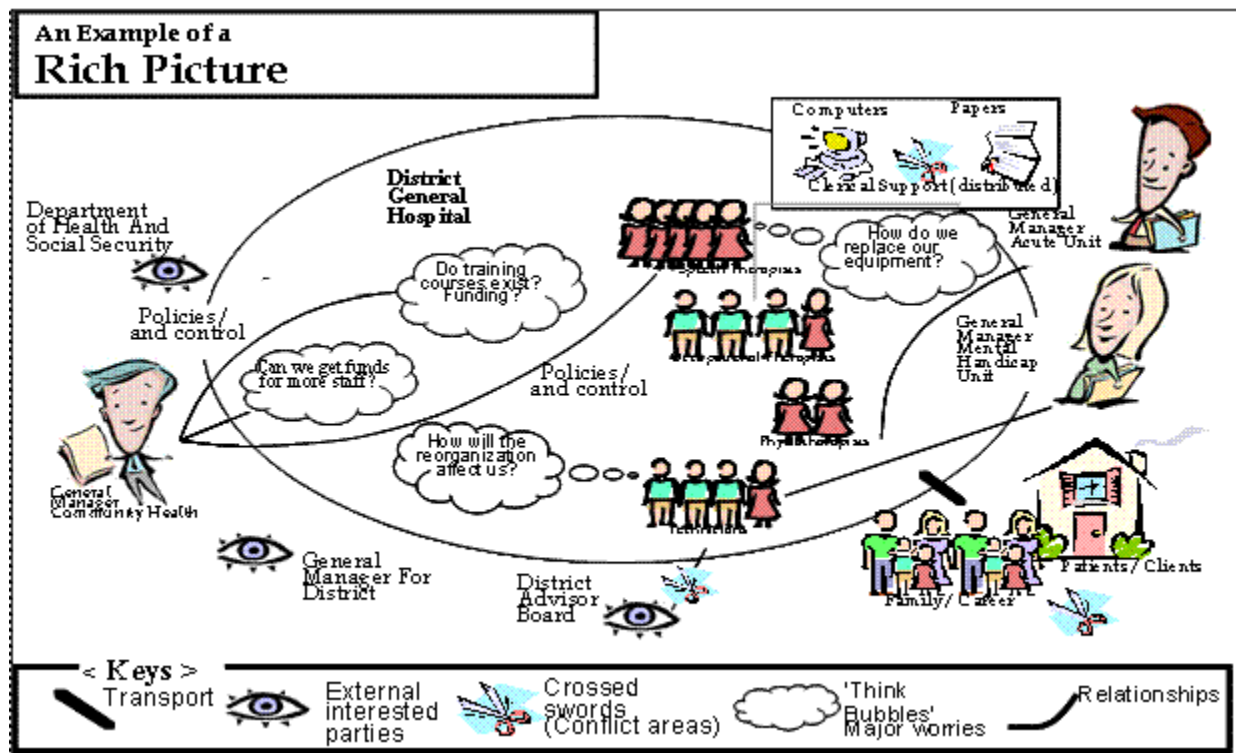
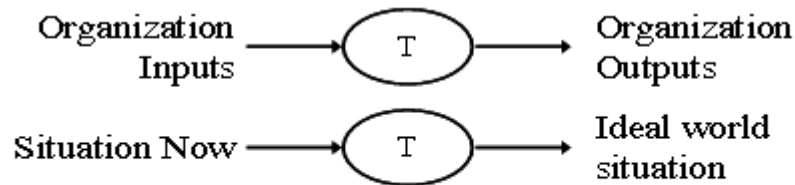


Figure 3. An Example of a Rich Picture

### Step 2: Structure Problem-Situation

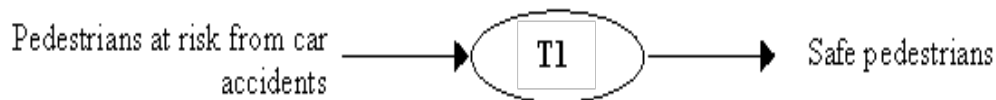
The "system" of purposeful human activity is split up into [related] sub-systems of human-activity that reflect separate purposes (or goals). This is done by defining sets of transformations that explain how the system works (input/output) or how it needs to change:



**Success measure** = what a successful transformation will achieve (*define a quantifiable measure*)

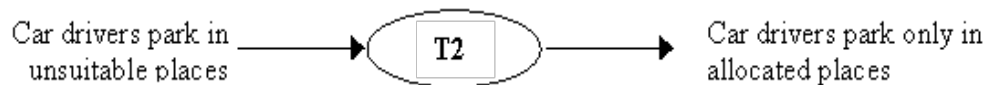
**Worldview** = how the world works and why this improvement is worth achieving

Examples (for a parking control system – ***you would expect to have at least 20 transformations***):



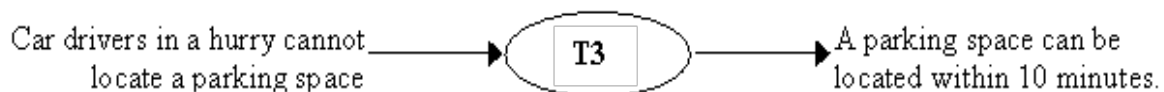
**Success measure** = pedestrian deaths and other injuries reduced by 10% of total, year on year.

**Worldview** = A lack of oversight on where people park results in unsafe parking that directly contributes to pedestrian injuries and fatalities.



**Success measure** = number of cars parked in unallocated parking spaces reduced by 10% a year.

**Worldview** = People park in unsafe or unallocated spaces because they cannot find a legal parking space within a short amount of time, because they do not want to walk more than a few yards to their destination, because they are unaware of the danger of parking in certain locations, and because there are few sanctions (they are unlikely to be ticketed or fined for illegal parking).



**Success measure** = number of cars parked illegally reduced by 10% a year.

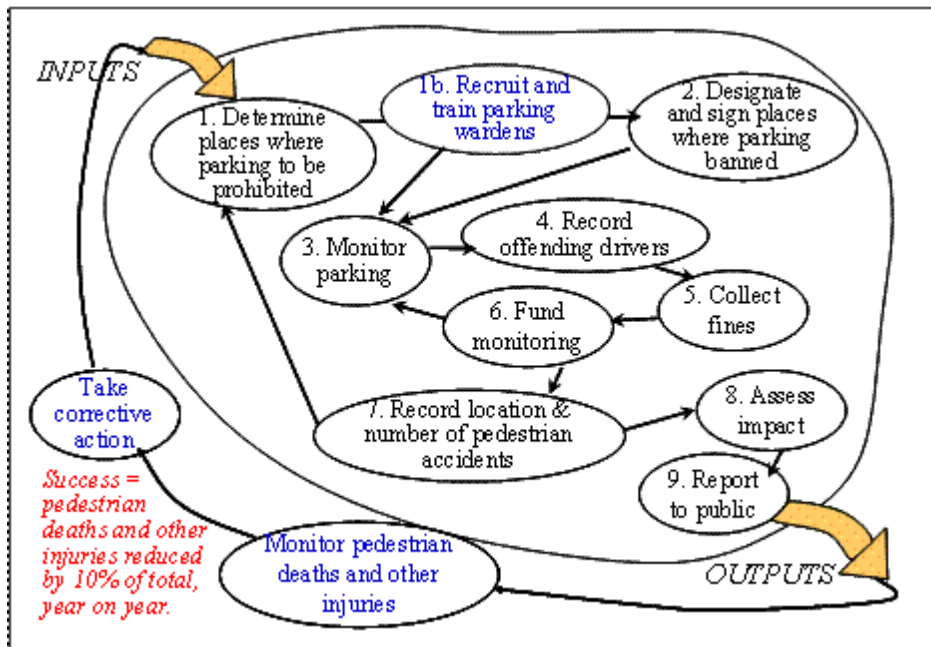
**Worldview** = To provide more parking spaces, we need to raise money to pay for this. The obvious ways are raising parking fees, raising local taxes, or raising car taxes. But many drivers do not insure or tax their vehicle and would park illegally rather than paying a lot for parking.

### Steps 3-4: Model Activities For Each Perspective

To reduce the complexity of the method, I will skip over step 3 and move onto modeling the activities required for the system. First, the activities are defined, then a model is produced that shows how these activities feed into each other. For example, the activities required for T1 are:

1. Determine places where parking causes danger to pedestrians.
2. Designate and sign those places where parking is not permitted.
3. Monitor parking in those places.
4. Record drivers parking in those places.
5. Collect fines from punishment.
6. Fund the additional monitoring with collected fines.
7. Measure the number and locations of pedestrians involved in accidents.
8. Assess the impact upon pedestrian safety
9. Report to the public on the results.

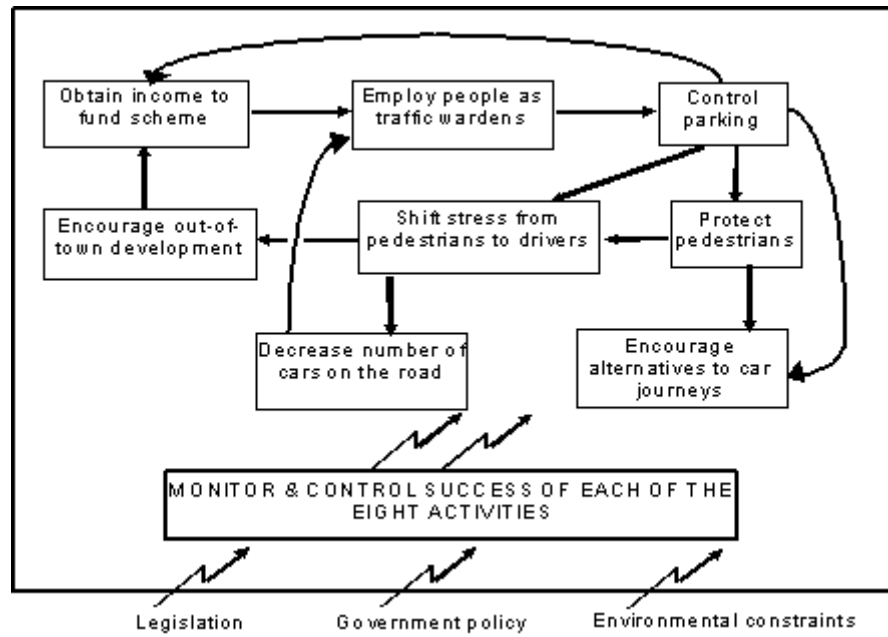
These activities build into the following conceptual model:



### Steps 5-7: Compare With Real World, Suggest Improvements, and Take Action

These steps depend on iterations of determining what is not done (or not done well), what is not communicated in the way required (i.e. information-flow between activities is missing), or what is not monitored appropriately. The success measure defined for each transformation should be monitored, as shown in the conceptual model above. The blue activities in the conceptual model represent activities that are not currently done. This is where feasibility is

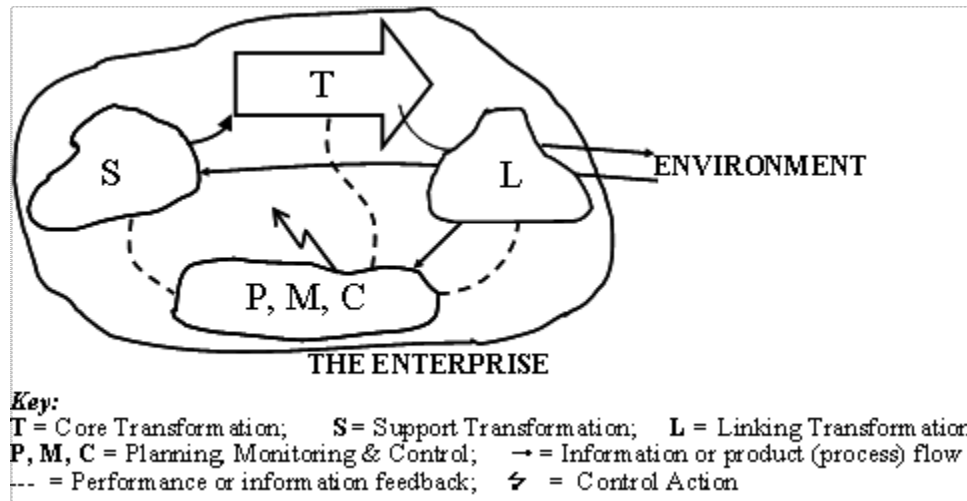
considered: how and can these activities be performed effectively? What should be fixed first? The “take corrective action” activity is complex and may suggest a new transformation that needs to be modeled as a whole, or it may relate to another transformation that was already modeled. To analyze priorities for change (and to analyze missed synergies between the subsets of activity), we can assemble a “bricolage” from various transformations.



Each process in model above represents the T from a single Transformation (purposeful sub-system), as modeled in step 2.

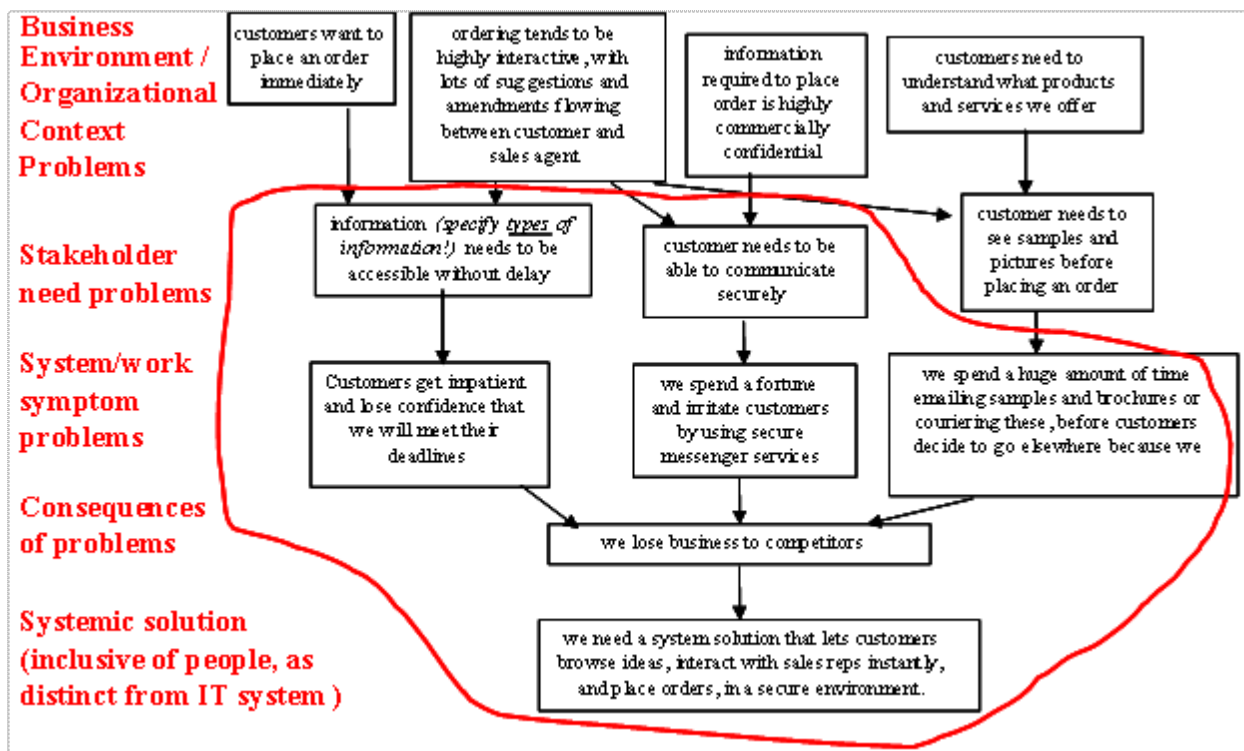
### *Enterprise system models*

This view of the enterprise lets you understand what is missing. Some of the transformations modeled will be core transformations, some will be support transformations, some of the linking transformations that are required will not have been included, and some planning, monitoring, and control systems will also have been forgotten.



## ADDITIONAL TECHNIQUES FOR SYSTEMIC ANALYSIS

*Problem cause-effect diagrams:*



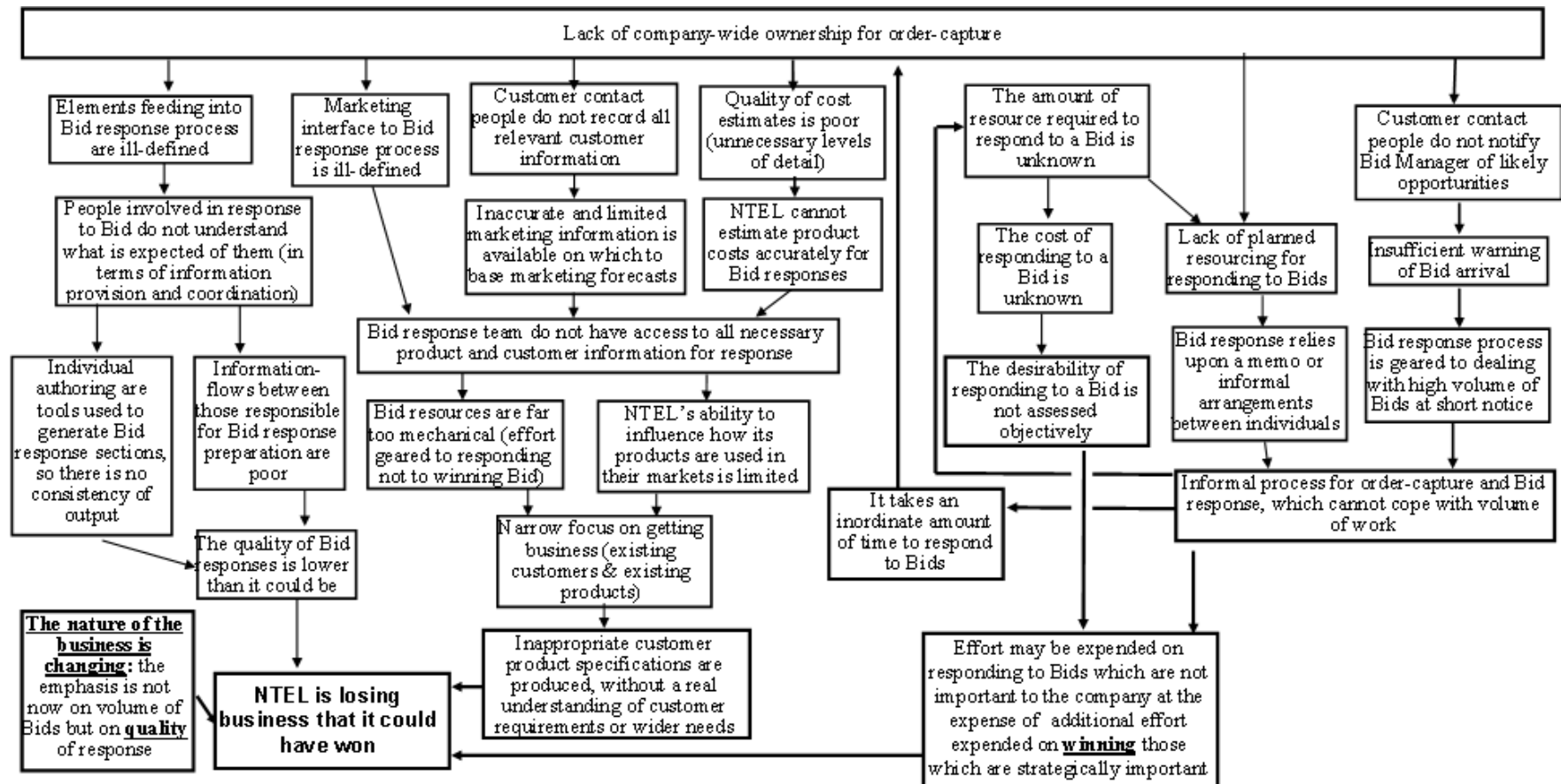
You can lose the levels of analysis once you become experienced at drawing problem cause-effect diagrams. They help to get started but get in the way of complex analysis. Just keep

working backwards. Elements inside the boundary-line are things you can affect or influence. Elements outside the line are constraints on what you can change.

#### Sample Problem Cause-Effect Diagram, Illustrating Vicious Circles of Causality

This example derives from a real-world company analysis (the names of the company, its functional groups, and business processes have been disguised). It demonstrates how a well-done problem analysis can reveal vicious cycles of causality, that need to be disrupted for any systems change to be effective. It also demonstrates how "root cause" problems constrain solutions to the problems experienced (symptom problems). Without this sort of analysis, any real-world change is doomed to failure, *or worse* -- a partial success that does not really change the culture of failure.





## References

Checkland, P. (1981) ***Systems Thinking, Systems Practice***, John Wiley & Sons.

Checkland, P. and Holwell, S. (1998) *Information, Systems, and Information Systems*, John Wiley & Sons.

[This is an excellent book and highly recommended].

Gharajedaghi, J. (1999) ***Systems Thinking: Managing Change & Complexity***, Butterworth Heinmann.

S. Gasson, Soft Systems Analysis web pages - *a student tutorial*,  
<http://cci.drexel.edu/faculty/sgasson/SSM/SSMhome.html>