Archetypes

By using the capacity to see recurring patterns and behaviours systems analysists can more simply also understand systems and suggest remedies or ways out of patterns.

Scientists may see the process of identifying archetypes as similar to other scientific processes. Physicists learn about the smallest subatomic particles by studying the traces they leave and their patterns; psychologists and other scholars study archetypes by examining their recurring present patterns in art, literature, myth, and dream. Carl Jung recognised that the archetypical images that recurred in his patients' dreams also could be found in the myths, legends, and art of ancient peoples, as well as in contemporary literature, religion, and art. They are archetypical because they leave the same pattern over and over again.

And what does an archetype mean then? That's simple. Archetypes are controlling paradigms or metaphors, the invisible patterns or symbols, how we experience the world.

The essence of systems thinking is that structure influences behaviour, put another way, when placed in the same system, different people tend to produce similar results, similar patterns; archetypes. The reason for this is that there are a number of 'rules' which seem to be true of all complex systems. System archetypes can be used as a stepping-stone to understanding more complex situations. Presently, about a dozen archetypes have been identified. Once an archetype is identified it usually suggests areas of high and low leverage. The following section examines probably the most common archetypes -

1. Balancing Loops
2. Limits to Success
3. Shifting the burden
4. Escalation
5. Success to the successful
6. Tragedy of the Commons
7. Fixes that fail
8. Growth and underinvestment
9. Eroding Goals
10. Accidental adversaries

There is space under each of these archetypes for notes about a.) Examples, b.) Early warning signs, and c.) Management principles

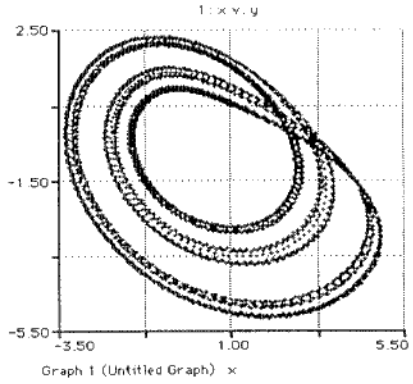
Chemical Engineering II
Lund University
Box 124
S-221 00 Lund, Sweden

Mats G E Svensson
email: mats.svensson@chemeng.lth.se

tel + 46 46 222 82 75
fax + 46 46 222 82 74

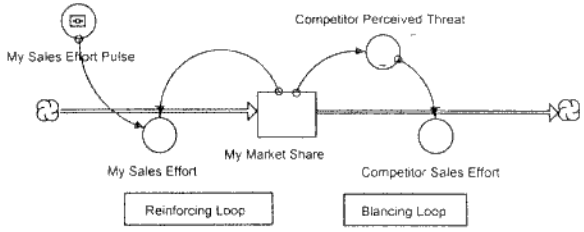
Archetypes in System Analysis

Mats Svensson



Systems Analysis Course
LUMES, Lund University
February 1999

Limits to Success



A Limits to Success structure is characterized by a reinforcing loop which is offset by a balancing loop. The reinforcing loop initially shows added performance for additional effort, which inturn feeds additional effort. This continues until some constraint produces a limiting action and additional effort does not produce additional results.

In this example My Sales Effort is balanced with Competitor Sales Effort so My Market Share remains constant. If I increase My Sales Effort, it will increase My Market Share which will promote additional sales effort. At some point the Competitor will perceive My Market Share to become a threat and response by increasing Sales Effort. This will limit My Market Share growth.

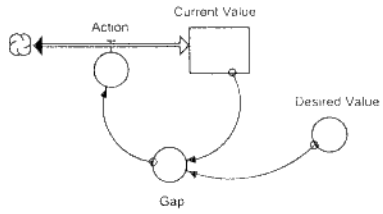
```
My_Market_Share(t) = My_Market_Share(t - dt) + (My_Sales_Effort - Competitor_Sales_Effort)
* dt
INIT My_Market_Share = 40

INFLOWS:
My_Sales_Effort =  1 * My_Market_Share + My_Sales_Effort_Pulse
OUTFLOWS:
Competitor_Sales_Effort = 4 + Competitor_Perceived_Threat
My_Sales_Effort_Pulse = 0
Competitor_Perceived_Threat = GRAPH(My_Market_Share)
(0.00, 0.00), (10.0, 0.00), (20.0, 0.00), (30.0, 0.00), (40.0, 0.00), (50.0, 0.00), (60.0, 0.00),
(70.0, 5.50), (80.0, 12.5), (90.0, 24.5), (100, 46.5)
```

Examples: .....

.....

Early warning symptoms: .....

Balancing Loop



Assumption: Current Value = Desired Value

Current Value and Desired Value interact to produce a Gap. The Gap promotes action to move the Current Value closer to the Desired Value.

As the Current Value approached the Desired Value the Gap decreased resulting in less action. When the Current Value reaches the Desired Value the Gap becomes zero and there is no further action.

A good example of this is when a car accelerates from 0 to 50 miles per hour. The initial Current Value is 0 and the Desired Value is 50. The Gap between the two results in pressure on the Accelerator. As the car gets closer and closer to 50 miles per hour the pressure on the Accelerator is reduced until when goal is reached a steady pressure is maintained.

This structure could just as well run in the reverse direction beginning with the Current Value being greater than the Desired Value. In this case the Gap and the resultant action would serve to decrease the Current Value until it reached the desired value.

```
Current_Value(t) = Current_Value(t - dt) + (Action) * dt
INIT Current_Value = 100

INFLOWS:
Action = Gap
Desired_Value = 100
Gap = Desired_Value -Current_Value
```

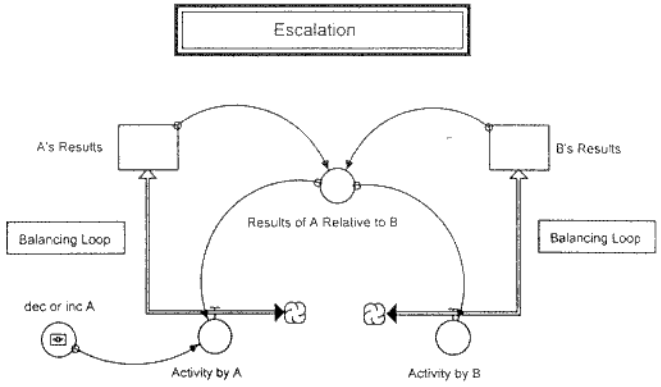
Examples: .....

.....

Early warning symptoms: .....

.....

Management principle: .....



An Escalation structure is characterized by two balancing loops which interact in a manner with results in a single reinforcing structure which may may be either virtuous or viscous.

This structure is stable until a fluctuation is introduced. When this happens A and B may both either rise or decline depending on the initial fluctuation.

```
A's_Results(t) = A's_Results(t - dt) + (Activity_by_A) * dt
INIT A's_Results = 100

INFLOWS
Activity_by_A = Results_of_A_Relative_to_B + dec_or_inc_A
B's_Results(t) = B's_Results(t - dt) + (Activity_by_B) * dt
INIT B's_Results = 100

INFLOWS:
Activity_by_B = Results_of_A_Relative_to_B
dec_or_inc_A = 0
Results_of_A_Relative_to_B = A's_Results - B's_Results
```

Examples: .....

.....

Early warning symptoms: .....

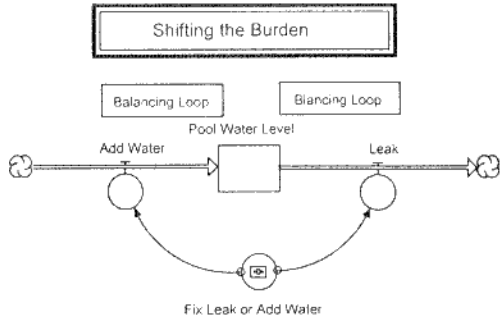
.....

Management principle: .....

168

.....

Management principle: .....



A Shifting the Burden structure is characterized two balancing loops, each of which moves something in the desired direction. One of these loops actually provides a better long term solution to the situation, yet has an associated time delay. The other balancing loop, the one most often chosen has a short term result, but does not actually resolve the fundamental difficulty, and the perceived problem returns.

Lets face it, this is a real dumb example, yet it's what came to me when I sat down to do this model. I have a pool in which I want to have 72" of water. The pool just happens to have a leak which leaks at a rate of .25" hr. So, I can add water or I can fix the leak. Adding water is easier, and takes far less time, but will ensure I have to add water again in the near future.

```
Pool_Water_Level(t) = Pool_Water_Level(t - dt) + (Add_Water - Leak) * dt
INIT Pool_Water_Level = 72

INFLOWS
Add_Water = if Fix_Leak_or_Add_Water > 0 then Fix_Leak_or_Add_Water else 0
OUTFLOWS
Leak = (.25 + time/10.0) * (If Fix_Leak_or_Add_Water = -1 then 0 else 1)
Fix_Leak_or_Add_Water = 0
```

Examples: .....

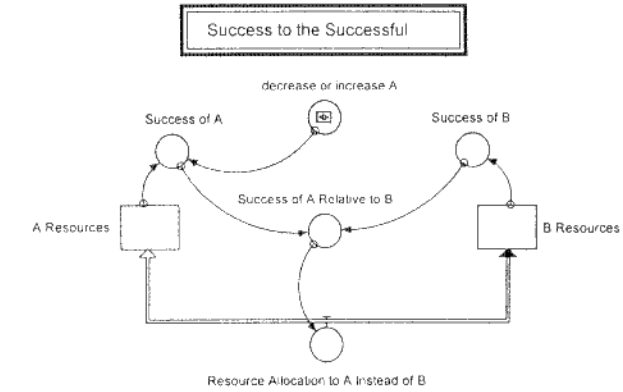
.....

Early warning symptoms: .....

.....

Management principle: .....

Management principle: .....



A Success to the Successful structure is characterized by two reinforcing loops which interact in a manner with results in a single reinforcing structure which may go in either direction.

Given two groups, A and B, each of whose success is dependent on the number of resources it has produces little change unless one group becomes even slightly more successful than the other.

If the success of one group increases or decreases it will set off a trend that will ensure its success continues in the same direction as the fluctuation.

```
A_Resources(t) = A_Resources(t - dt) + (Resource_Allocation_to_A_Instead_of_B) * dt
INIT A_Resources = 50

INFLOWS:
Resource_Allocation_to_A_Instead_of_B = .1 * Success_of_A_Relative_to_B
B_Resources(t) = B_Resources(t - dt) + (- Resource_Allocation_to_A_Instead_of_B) * dt
INIT B_Resources = 50

OUTFLOWS:
Resource_Allocation_to_A_Instead_of_B = .1 * Success_of_A_Relative_to_B
decrease_or_increase_A = 0
Success_of_A = A_Resources + decrease_or_increase_A
Success_of_A_Relative_to_B = Success_of_A - Success_of_B
Success_of_B = B_Resources
```

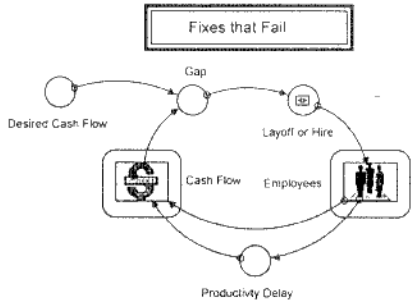
Examples: .....

.....

Early warning symptoms: .....

.....

Management principle: .....



A Fixes that Fail structure is characterized by a balancing structure offset by a reinforcing structure which acts after some delay.

If the business is experiencing cash flow problems, i.e. the Desired Cash Flow is greater than the current Cash Flow, the organization has two options. It can either cut costs, usually accomplished by a reduction in resources, or it can increase revenue.

Reducing expenses via a layoff has an immediate affect on expenses thus increasing cash flow. The inner circle is a balancing loop which seeks to achieve the Desired Cash Flow.

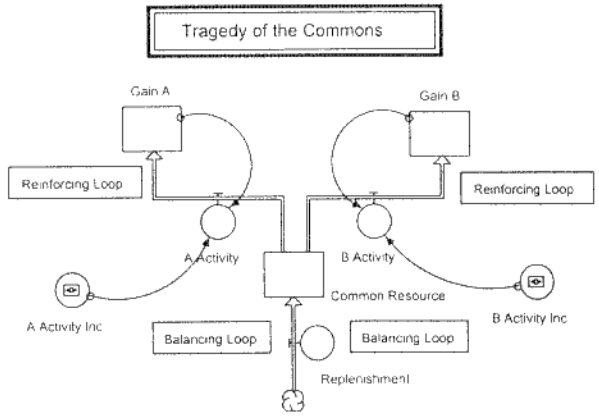
Reducing Employees will also produce a reduction in revenue, yet this will not be experienced for some period because of the delays built into the structure. The outer circle represents a viscous reinforcing loop which will, over time, overshadow the gains from the short term balancing loop.

As it turns out the best answer is counterintuitive. When Desired Cash Flow is less than Current Cash Flow you increase cost by hiring additional employees. Your Cash Flow then goes down, but only for a while.

```
Desired_Cash_Flow = 30000
Gap = Desired_Cash_Flow - cash_flow'
Layoff_or_Hire = Gap * 0 {This is zeroed out to provide for user inteaction as to whether to hire for layoff
employees}
Productivity_Delay = DELAY(employees',6,employees')
Cash Flow
cash'(t) = cash(t - dt) + (revenue' - expense') * dt
INIT cash' = 10000
INFLOWS:
revenue' = 2250 * Productivity_Delay
OUTFLOWS:
expense' = 2000 * employees'
cash_flow' = revenue' - expense'
Employees
employees'(t) = employees'(t - dt) + (emp_chg') * dt
INIT employees' = 100
INFLOWS:
emp_chg' = Layoff_or_Hire
```

Examples: .....

.....



A Tragedy of the Commons structure is characterized by two reinforcing loops which are offset by two balancing loops.

In this example A and B both rely on the same common resource for their gains. As each gains it increases its use of the common resource, until such time as the use of the common resource exceedes the replenishment of the common resouce. Whe this happens the Gain of both A and B is limited.

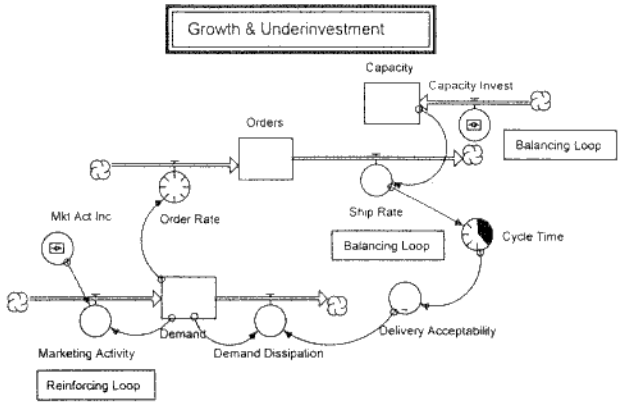
Nothing fails like success!

```
Common_Resource(t) = Common_Resource(t - dt) + (Replenishment - A_Activity - B_Activity) *
dt
INIT Common_Resource = 100

INFLOWS:
Replenishment = 10
OUTFLOWS:
A_Activity = 1 * Gain_A + A_Activity_Inc
B_Activity = 1 * Gain_B + B_Activity_Inc
Gain_A(t) = Gain_A(t - dt) + (A_Activity) * dt
INIT Gain_A = 1
```

Examples: .....

.....

Early warning symptoms: .....



A Growth & Unverinvestment structure is really an enhanced Limits to Growth structure. The structure begins with a reinforcing loop which continues to cause a change in the direction of action, yet at some point an offsetting balancing loop take over to limit the performance. The investment portion is represented by another balancing structure which can server add reduce the limiting action by the other balancing structure. I say can, because it seldom does because of associated delays. By the time it is decided to take action, the requirement to take action has passed.

Marketing activity creates Demand, which creates more Marketing Activity. Deman creates Orders. As the Order Rate outstrips Ship Rate Cycle Time increases and Delivery Acceptability Declines. As Delivery Acceptability declines is increases the Demand Dissipation rate. The trick is to add Capacity at the right times so the Ship Rate satisfies the growth in orders crated by the Marketing Activity.

```
Capacity(t) = Capacity(t - dt) + (Capacity_Invest) * dt
INIT Capacity = 10
INFLOWS
Capacity_Invest = 0
Demand(t) = Demand(t - dt) + (Marketing_Activity - Demand_Dissipation) * dt
INIT Demand = 10
INFLOWS
Marketing_Activity = .1 + ( .1 * (Demand - 10)) + Mkt_Act_Inc
OUTFLOWS
Demand_Dissipation = ( .01 + (-Delivery_Acceptability)) * Demand
Orders(t) = Orders(t - dt) + (Order_Rate - Ship_Rate) * dt
INIT Orders = 50
INFLOWS
Order_Rate = Demand
OUTFLOWS
Ship_Rate = Capacity
Cycle_Time = if time <= 6 then 5 else CYCLETIME(Ship_Rate)
Mkt_Act_Inc = 0
Delivery_Acceptability = GRAPH(Cycle_Time) (0.00, 0.00), (1.00, 0.00), (2.00, 0.00), (3.00, 0.00), (4.00, 0.00),
(5.00, 0.00), (5.00, 0.00), (7.00, 0.00), (8.00, -0.22), (9.00, -0.56), (10.0, -1.00)
```

Examples: .....

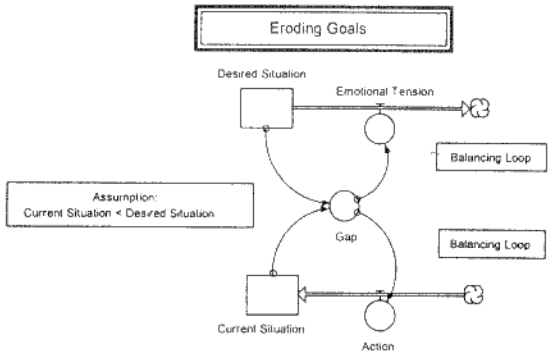
.....

Early warning symptoms: .....

Early warning symptoms: .....

.....

Management principle: .....



An Eroding Goals structure is characterized by two balancing loops which influence the same result, yet one of them has a time delay.

In this example the difference between the Desired Situation and the Current Situation produces a gap. This Gap has two influences. First, it produces emotional tension tends to reduce the Desired Situation. This is a short term interaction. Second, the Gap promotes action which will move the Current Situation in the direction of the Desired Siuation, thus reducing the gap.

The difficulty lies in that the time it takes for the Action to move the Current Situation in the direction of the Desired Situation is much longer than the time it takes for the Emotional Tension to reduce the Desired Situation. As such, the Gap will dissappear with the Current Situation at a level much less than was initiall desired.

```
Current_Situation(t) = Current_Situation(t - dt) + (Action) * dt
INIT Current_Situation = 10

INFLOWS:
Action = .02 * Gap
Desired_Situation(t) = Desired_Situation(t - dt) + (- Emotional_Tension) * dt
INIT Desired_Situation = 100

OUTFLOWS:
Emotional_Tension = .1 * Gap
Gap = Desired_Situation - Current_Situation
```

Examples: .....

.....

Early warning symptoms: .....

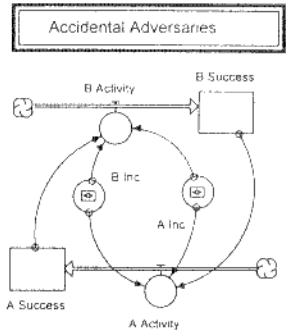
.....

Management principle: .....

.....

Management principle: .....

Management principle: .....



An Accidental Adversaries structure is characterized by a reinforcing structure which is offset by two internal balancing loops.

In this example A's Success along with B's Activity leads to B's Success and B's Success along with A's Activity leads to A's Success. This is single reinforcing structure.

Being that we are seldom willing to leave well enough alone, either A or B, or both, may undertake efforts to increase their own activity to increase their success. Well as it turns out, their increase in activity results in a decrease in the activity and success of the other, and ends up limiting their own success

This is definitely one of those situations where doing nothing is better than doing something. Sometimes it is best to leave well enough alone.

```
A_Success(t) = A_Success(t - dt) + (A_Activity) * dt
INIT A_Success = 1

INFLOWS:
A_Activity = 1 * B_Success + A_Inc - 5 * B_Inc
B_Success(t) = B_Success(t - dt) + (B_Activity) * dt
INIT B_Success = 1

INFLOWS:
B_Activity = 1 * A_Success + B_Inc - 5 * A_Inc
A_Inc = 0
B_Inc = 0
```

Examples: .....

.....

Early warning symptoms: .....

.....

Management principle: .....