Diagnostic Methods

Model Based Diagnostic Methods
Part 2

Outline

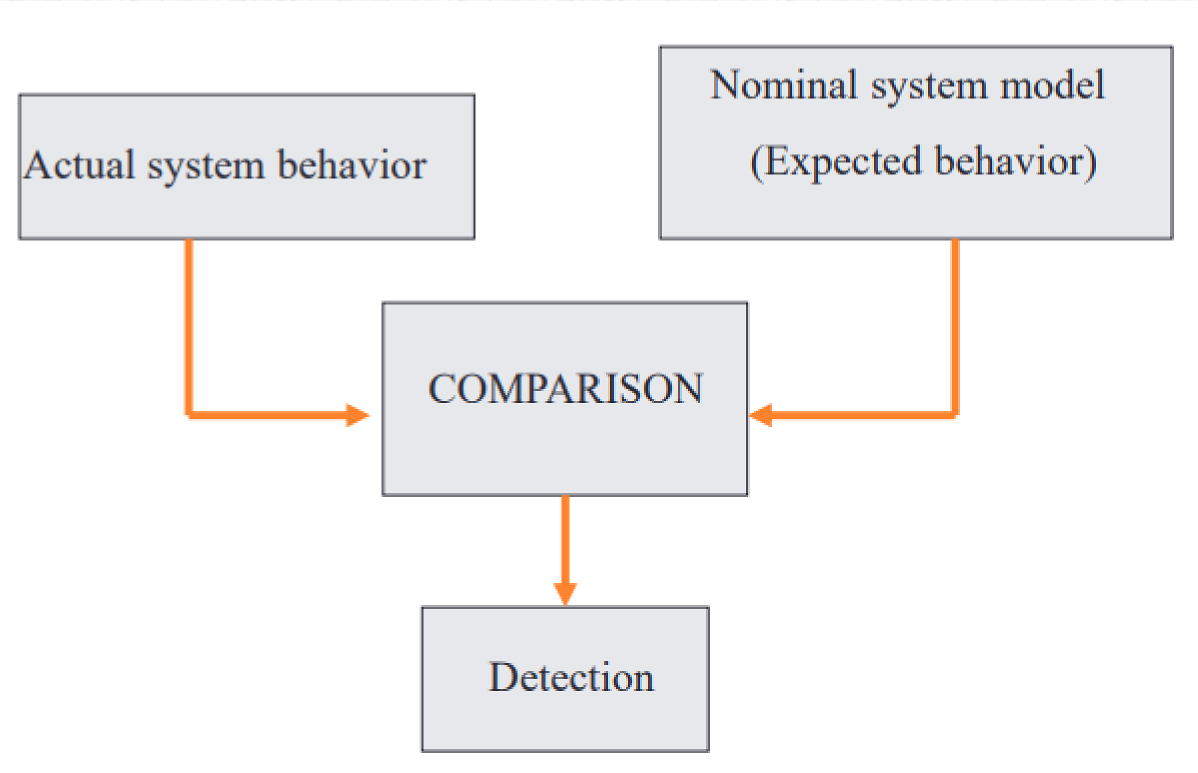
- Introduction
- Fault Concepts
- Fault Types
- Structured Analysis and Design Technique (SADT):
  - Construction
  - Limitations/Strengths
  - Roles
  - Case study

Introduction

- Fault detection plays an important role in high cost and safety-critical processes.
- Early detection of process faults can help avoid abnormal event progression.
- Fault detection, isolation and fault identification are together referred as fault diagnosis.

Introduction

- In model based approaches, we compare actual system with a nominal model system:
  - SADT



Fault Concepts

- It is of importance to define terminology of fault diagnosis:
  - Faults: A fault is an unpermitted deviation of at least one characteristics property (feature) of the system from the acceptable, usual, standard condition.
    - Causes: design errors, implementation errors, human errors, use, damages…
    - Consequences: worse performances, energy waste, waste of raw materials, economic losses, lower quality, lower production, environmental damages, human damages…

Fault Concepts

- Failure: A failure is a permanent interruption of a system’s ability to perform a require function under specified operating conditions.
- Malfunction: A malfunction is an intermittent irregularity in the fulfillment of a system’s desired function.

Fault Types

- Depending on the *localization* of the fault:
  - External fault: interactions between system and environment are not compatible with goals,
    - Inflow is too small
    - Input valve totally open
  - Internal fault: Depending on the faulty component,
    - Tank leakage
    - clogged pipe

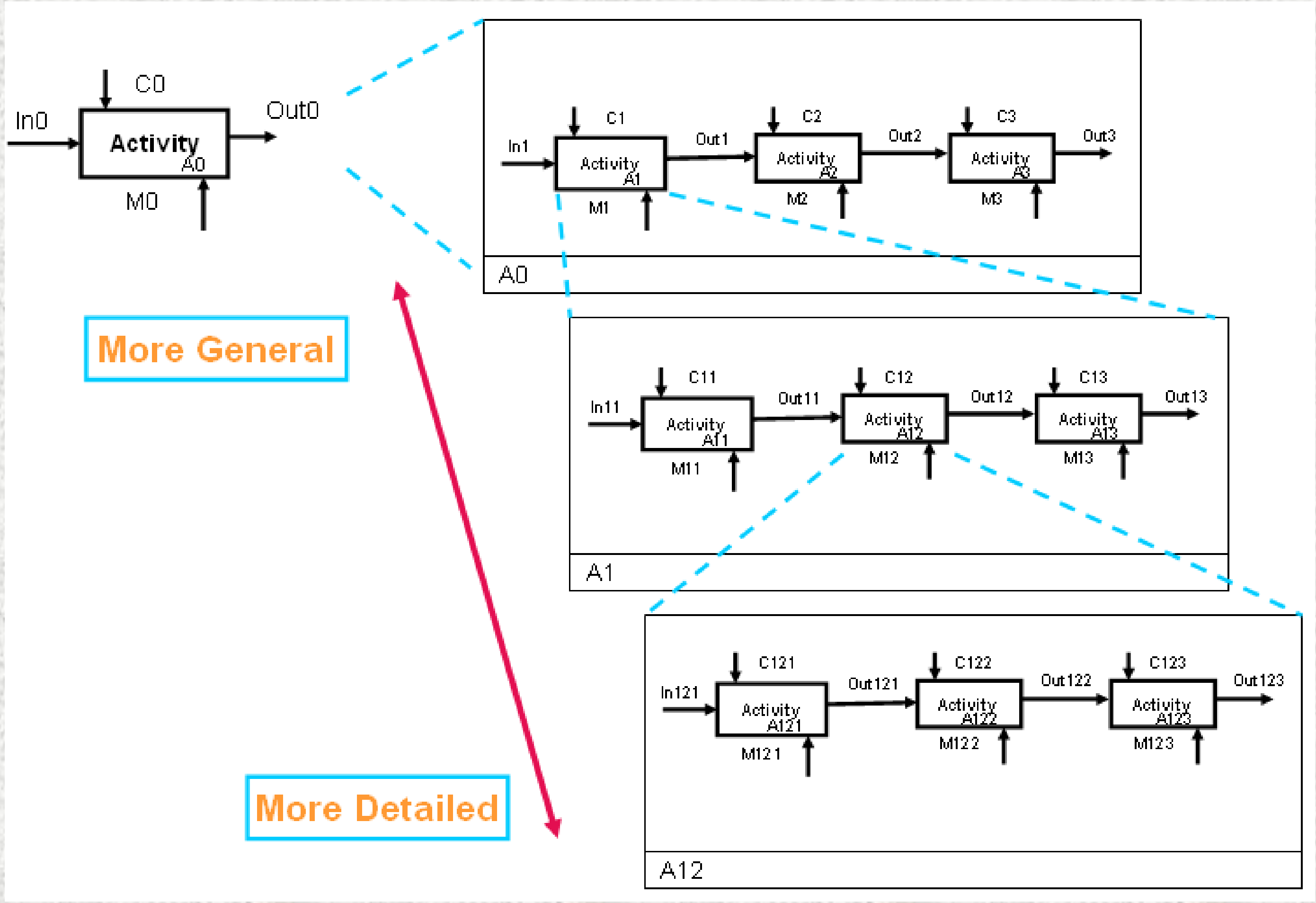
Fault Types

- Depending on the *magnitude* of the fault:
  - Acceptable departure from the usual state.
  - Failure: Catastrophic.
- Depending on the *temporal* aspects (time):
  - Abrupt fault: sudden and considerable.
  - Incipient or evolutive fault: affects slowly.

Structured Analysis and Design Technique (SADT)

- SADT is a structured analysis modelling language, which uses two types of diagrams: *activity* models and *data* models. It was developed in the late 1960s by Douglas T. Ross.
- SADT is a diagrammatic notation designed specifically to help people describe and understand systems.
- SADT is a *manual* graphical system for system analysis and design.
- It offers building blocks to represent data and activities, and a variety of arrows to relate boxes. These boxes and arrows have an associated semantics.

Structured Analysis and Design Technique (SADT)



Construction

- It follows the top-down principle.
- It is represented by the rectangular transformation box with its four symmetrically arrows.
- If the box represents an activity, the left arrow describes the input data, the right arrow the output data, the upper arrow the control data and the lower arrow the means used to effect the transformation.

Construction

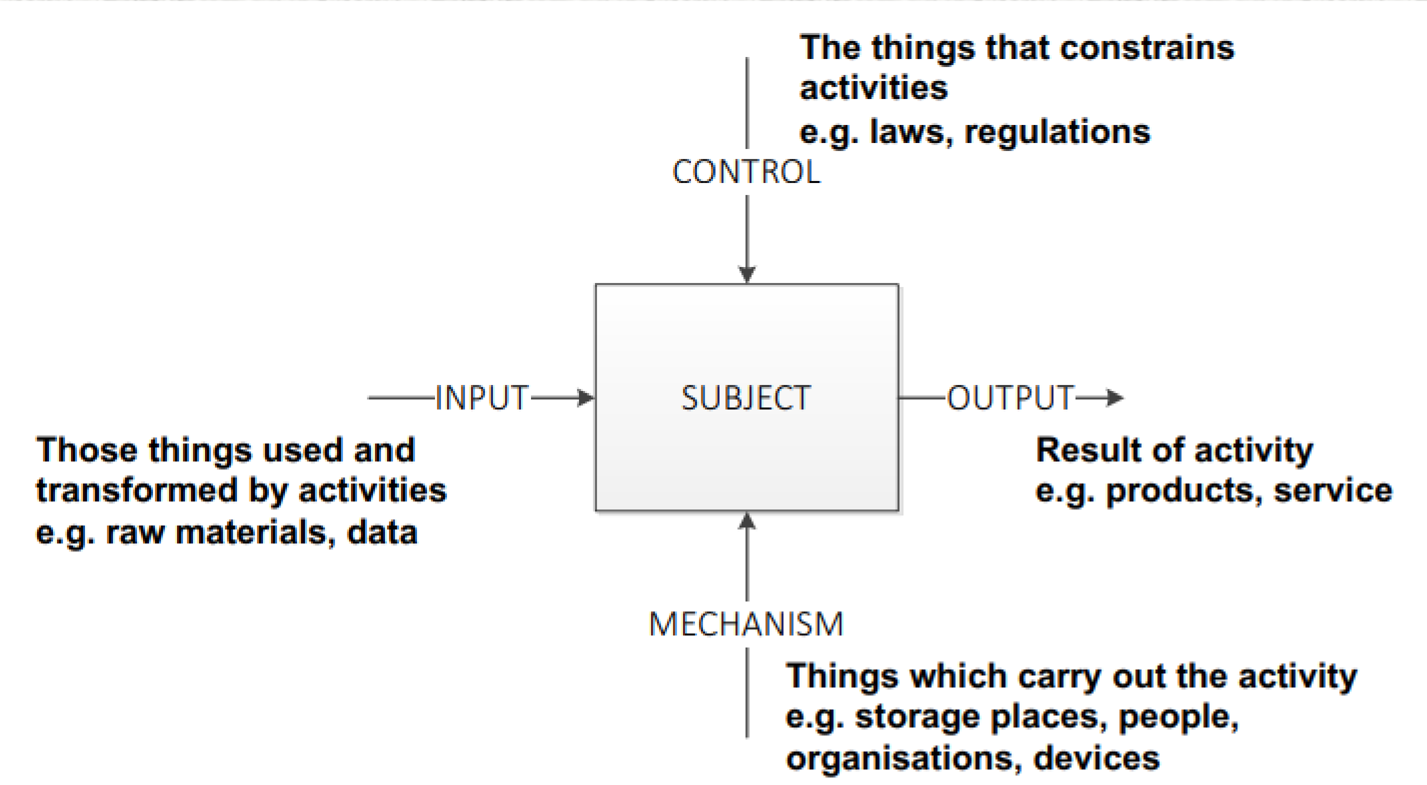
- SADT uses two types of diagrams: activity models and data models. The SADT's representation is the following:
  - A main box where the name of the process or the action is specified.
  - On the left-hand side of this box, incoming arrows: inputs of the action.
  - On the upper part, the incoming arrows: data necessary for the action.
  - On the bottom of the box, incoming arrows: means used for the action.
  - On the right-hand side of the box, outgoing arrows: outputs of the action.

Construction

- The semantics of arrows for *activities*:
  - Inputs enter from the left and represent data or consumables that are needed by the activity.
  - Outputs exit to the right and represent data or products that are produced by the activity.
  - Controls enter from the top and represent commands or conditions which influence the execution of an activity but are *not consumed*.
  - Mechanisms identify the means, components or tools used to accomplish the activity. Represents allocation of activities.

Construction

- SADT basic elements: activity diagram



Construction

- The semantics of arrows for *data* (3 types):
  - Inputs are activities that produce the data.
  - Outputs consume the data.
  - Controls to represent constraints influencing data.

Construction

- the system may be represented equally by a connected set of data boxes.
- The activity boxes are connected by links representing data while the data boxes are connected by links representing activities.
- The methodology acknowledges that the activity and data representations are symmetric and equally valid.
- but since a system has to be implemented from its constituent activities, the activity-based approach is the dominant one.

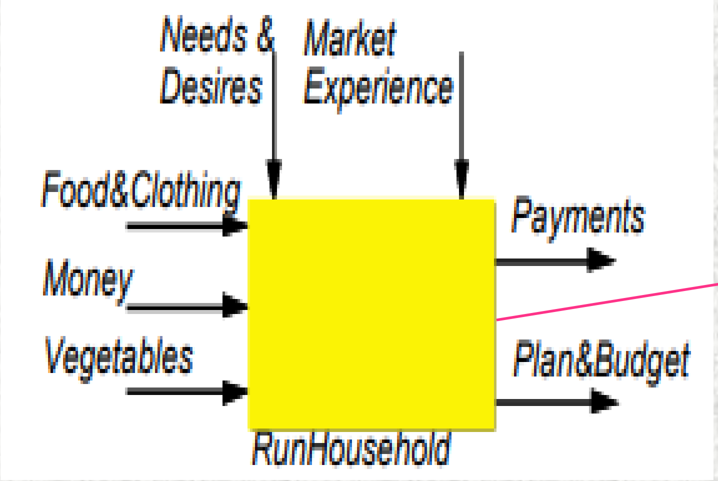
Limitations/Strengths

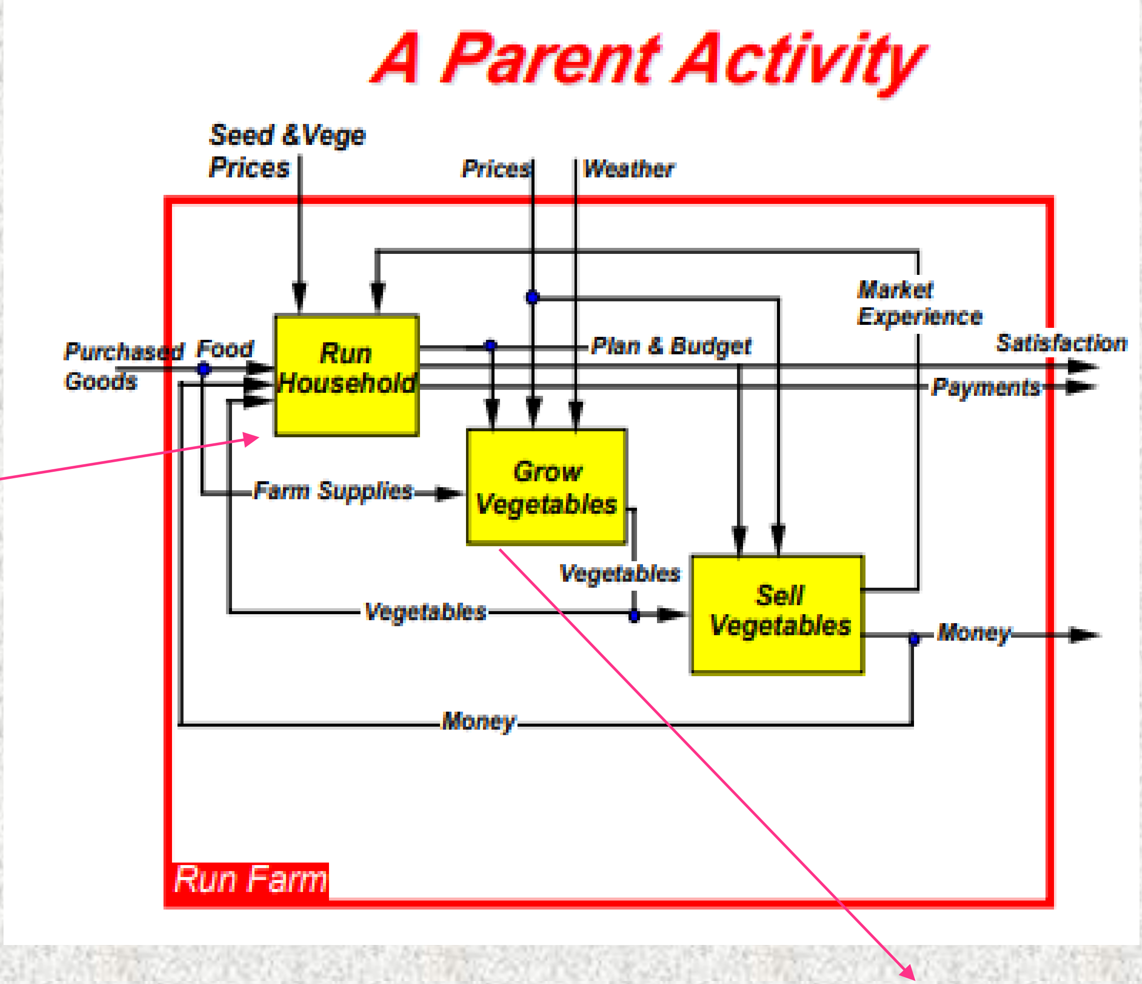
- Strengths of SADT:
  - Precise and concise scheme to represent graphically complex system requirements.
  - The notation is easy to understand, and the requirements can be discussed at whatever level is appropriate for the customer.
  - Clear top-down decomposition for I/O and control.
  - Distinguishes between control data and input data.
- Limitations:
  - One of the limitations of SADT approach is handling the systems with a large number of activities and resources where the combination of individual models can prove to be cumbersome.

Roles

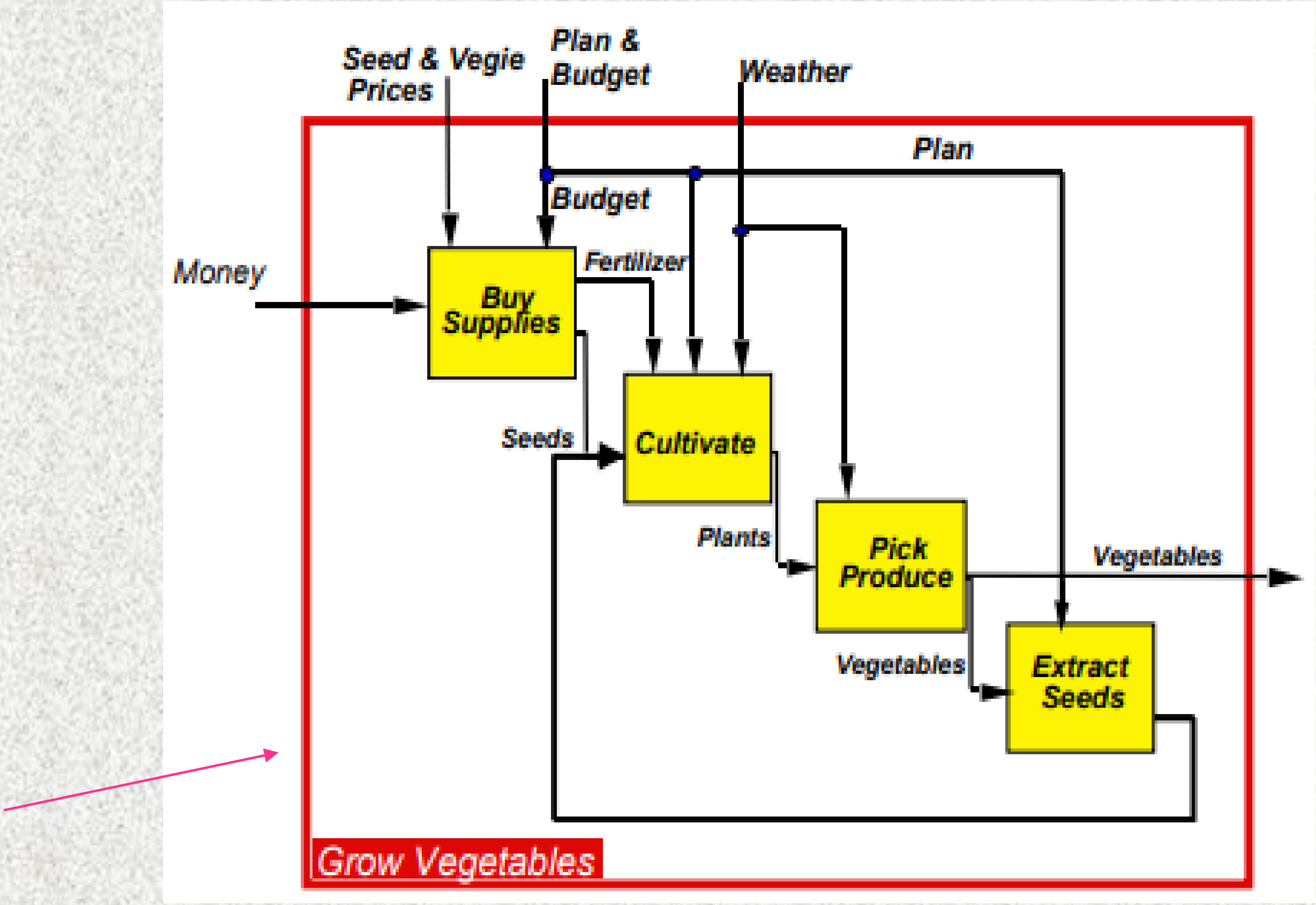
- In the SADT development process, multiple roles can or should be distinguished:
  - Author or developer of the SADT models
  - Commenters, who review the author's work
  - Readers or users of the SADT models
  - Experts, who can advise the authors
  - Technical committee or reviewers of the SADT models in detail
  - Project librarian, who govern the project documentation
  - Project manager, who governs the system analysis and design.
  - Monitor or chief analyst to assists SADT developers and users
  - Instructor to train SADT developers and users

Case Study 1


Needs & Desires
Market Experience
Food&Clothing
Money
Vegetables
Payments
Plan&Budget
RunHousehold


A Parent Activity
Seed &Vege Prices
Prices
Weather
Market Experience
Satisfaction
Purchased Goods
Food
Run Household
Plan & Budget
Payments
Farm Supplies
Grow Vegetables
Vegetables
Vegetables
Sell Vegetables
Money
Money
Run Farm

Case Study 1



Seed & Vegie Prices
Plan & Budget
Weather
Plan
Budget
Money
Buy Supplies
Fertilizer
Seeds
Cultivate
Plants
Pick Produce
Vegetables
Vegetables
Extract Seeds
Grow Vegetables

Case Study 2

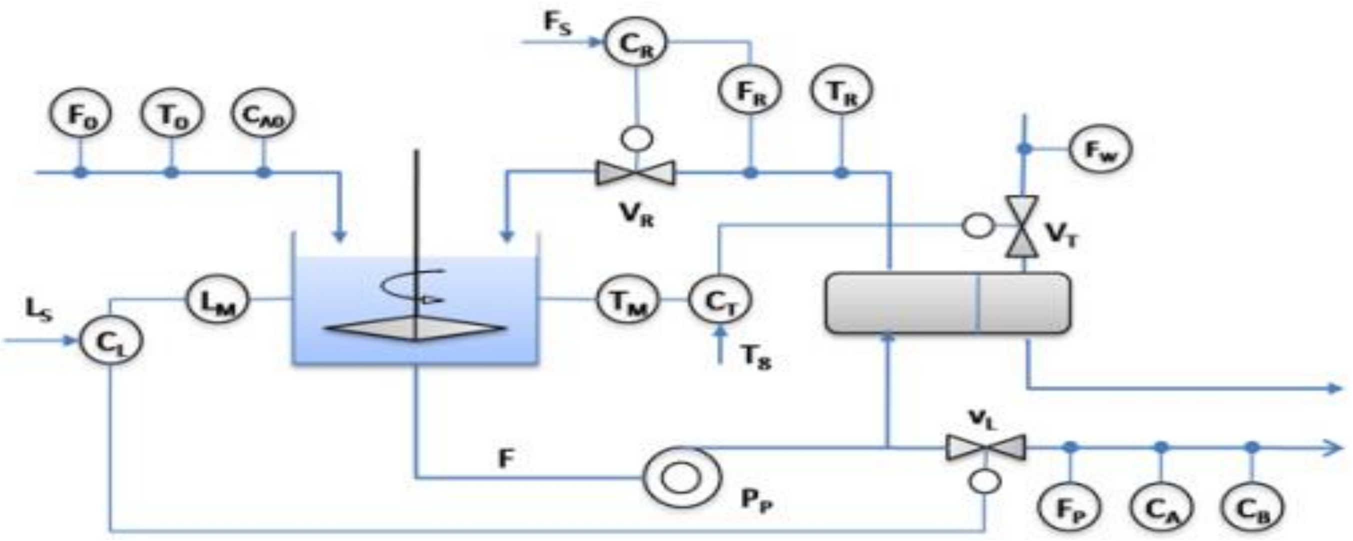
- This example is part of a Master thesis.
- It is related to a diagnostic system in a dairy.
- The studied process Is composed of the following elements:
  - Reactor.
  - Heat exchanger.
- The temperature in the reactor is set at a constant value through a preheated liquid in the heat exchanger where another cooling liquid lowers the temperature of the liquid leaving the reactor.

Case Study 2

- The flow of this liquid is controlled using a controller and a valve in addition to another controller is used to stabilize the level of liquid in the reactor.

Case Study 2

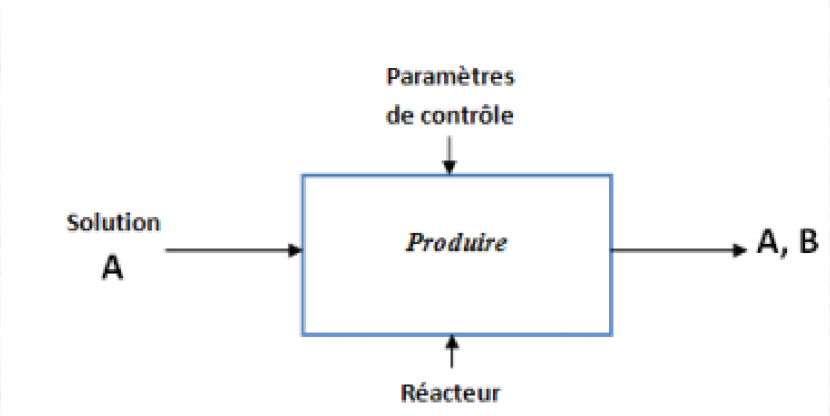
- Industrial process with reactor and heat exchanger

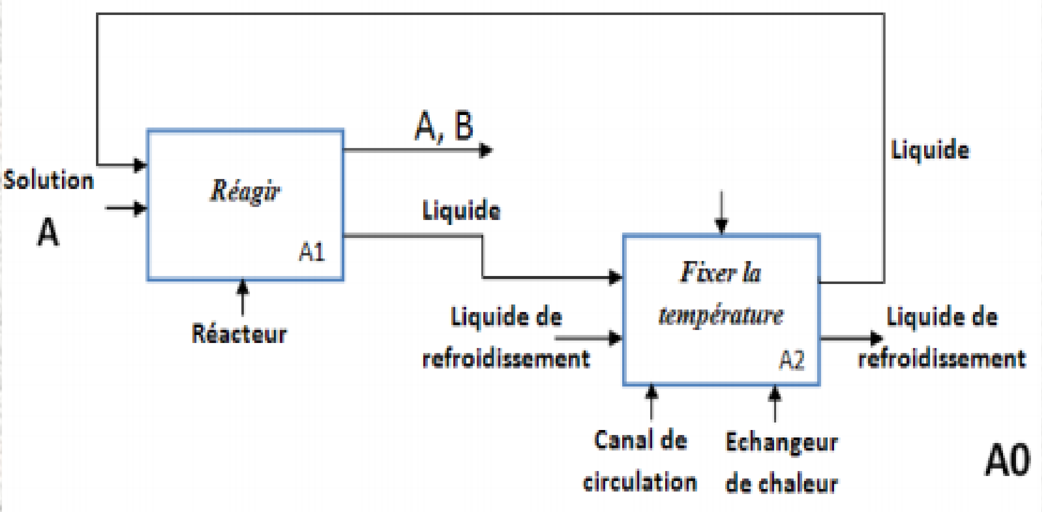


F_0 : Liquide entrant au réacteur.
T_0 : Température du liquide entrant au réacteur.
C_{A0} : Concentration de la substance A dans le liquide d'entrée.
L_M : Niveau du liquide dans le réacteur.
T_M : Température du réacteur.
F_P : Liquide sortant du processus.
C_A : Concentration de la substance A dans le liquide sortant.
C_B : Concentration de la substance B dans le liquide sortant.
F_W : Liquide refroidissant dans l'échangeur de chaleur
F_R : Liquide circulant.
T_R : Température du liquide circulant.
C_R : Sortie du contrôleur de circulation.
C_L : Sortie du contrôleur de niveau.
C_T : Sortie du contrôleur de température du réacteur

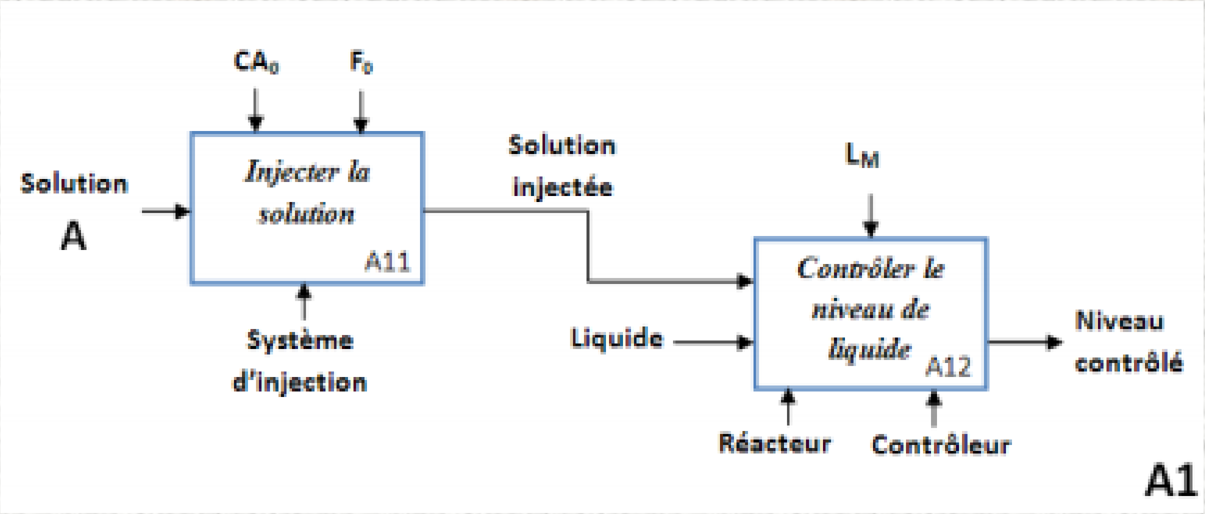
Case Study 2

- SADT diagram

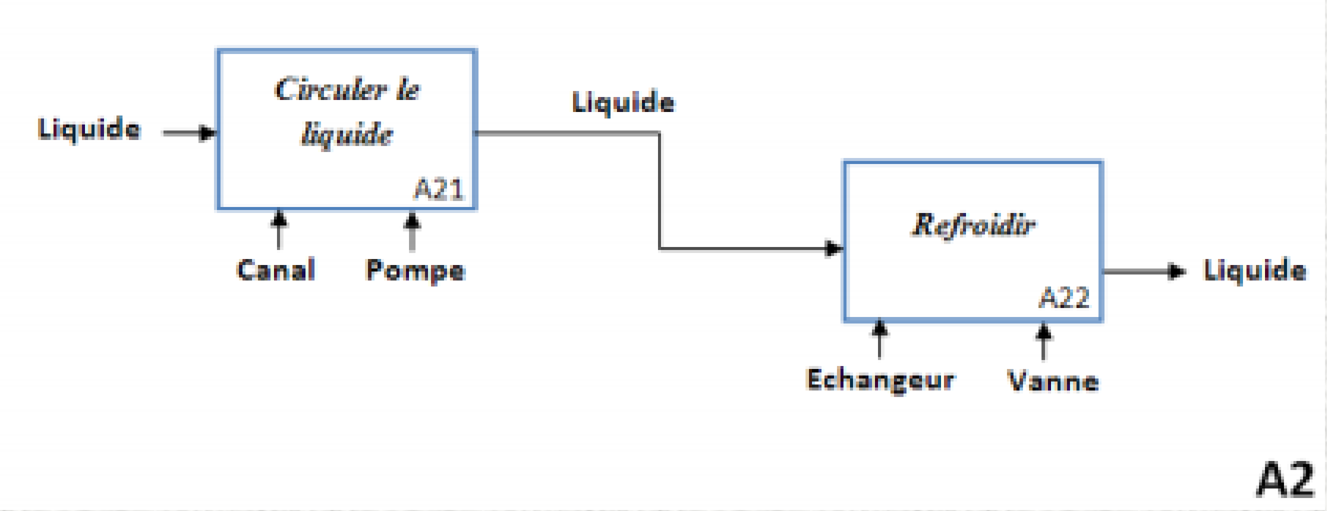




Case Study 2



CA0
F0
Solution A
Injecter la solution
A11
Système d'injection
Solution injectée
LM
Liquide
Contrôler le niveau de liquide
A12
Niveau contrôlé
Réacteur
Contrôleur
A1



Liquide
Circuler le liquide
A21
Canal
Pompe
Liquide
Refroidir
A22
Liquide
Echangeur
Vanne
A2