WBF-WG “Flow Analysis”
ISA-88 Modeling using Flow Analysis

Draft 3 – September 2003

This document is a draft that represents work being done by a WBF Working Group

WBF grants permission to anyone to reproduce and distribute copies of this WBF draft, in whole or in part, but only for the following purposes and only as long as the recipient is not charged any fee for the copy (nor may the copy be included as part of a package with other materials or presentations for which a fee is charged):

1. Review of and comment on the draft standard;
2. Provide to others for review and comment;
3. Promotion of the WG work; or
4. Informing and educating others about the WG.

In addition, all copies must reproduce a copyright notice as follows:

© Copyright 2001 WBF. All rights reserved. Reproduced and distributed with permission of WBF.

WBF reserves all other rights to the draft. Any other reproduction or distribution without the prior written consent of WBF is prohibited.

The reader is cautioned that this document has not been approved and cannot be presumed to reflect the position of WBF or any other committee, society, or group. Although every effort has been made to ensure accuracy, neither WBF, members of WG, nor their employers shall be held liable for errors or limitations.

Yellow highlighting: under construction

Blue highlighting: question marks, editor’s notes

-

WBF-WG_FA Draft on Flow Analysis

Copyright © 2001 by WBF. All rights reserved. Not for resale. Printed in the United States of America

World Batch Forum
107 S. Southgate Drive
Chandler, Arizona 85226
Phone: 480-893-8803   Fax: 480.893.7775
E-mail: deann@kc-a.com

Preface

WBF working groups provide an unbiased means for the investigation and development of work items of interest to the members. People of like interests meet (in person or electronically) in order to delve into specific topics. Although the topics addressed will be determined by each working group possible topics are:

- XML based batch data exchange formats
- *Exception handling*
- Continuation or publication of EBF working group results
- Process plant analysis based on ISA-88
- Procedure Function Chart examples
- Good practices
- Benchmarks

While the WBF will own the copyright to all WBF Working Group Reports, all working group members are given royalty free use of the WBF Working Group Reports for personal and professional purposes. This includes use by the employers of working group members, to use work products of the working group, including any work published by the working group.

This draft technical report has been issued by the World Batch Forum (WBF), Working Group “Flow Analysis”

The following served as members of this group:

Email	Name	Company	Country
a-smith@its-ltd.co.uk	Anthony Smith	INDUSTRIAL TECHNOLOGY LTD	UK
alain.bosatelli@eu.rhodia.com	Alain Bosatelli	RHODIA	FR
alfonso.sancho@honeywell.com	Alfonso Sancho	HONEYWELL	
bernard.cubizolles@isa-france.org	Bernard Cubizolles	ASOPHIA	FR
bertrand.ricque@euraltech.fr	Bertrand Ricque	EURALTECH	FR
dbrick@aegiscorp.com	D. Brick	AEGIS ANALYTICAL	US
dclermon@lu.danone.com	Dominique Clermont	DANONE	FR
djoannes@danonefr.danone.com	Didier Joannes	DANONE	FR
dror@specsoft-pfs.com	Dror Eyal	SPEC SOFT	IS
Ed.Lynch@pfizer.com	Ed Lynch	PFIZER	US
emerson@dallas.yca.com	Dave Emerson	YOKOGAWA	US
emol@starren.nl	Elbert Mol	STARREN	NL
eric_vitte@mail.schneider.fr	Eric Vitté	SCHNEIDER ELECTRIC	FR
etsmrb@aol.com	Rémi Métras	RMB	FR
fl@controldraw.co.uk	Francis Lovering	CONTROL DRAW	UK
j.relland@alpha-cim.com	José Relland	ALPHA-CIM	FR
Jacky.Tachon@aventis.com	Jacky Tachon	AVENTIS	FR
Jacques.cothenet@eu.rhodia.com	Jacques Cothenet	RHODIA	FR
jacques.grossin@adp.fr	Jacques Grossin	AEROPORTS DE PARIS	FR
jean-francois.cardot@isa-france.org	Jean-Francois Cardot	RHODIA	FR
jean-pierre.bovee@aventis.com	Jean-Pierre Bovée	AVENTIS PHARMA	DE
jean.vieille@isa-france.org	Jean Vieille* ***	CONSULTANT	FR
jmrayon@jmrconseils.fr	Jean-Michel Rayon	CONSULTANT	FR
Josef.Meixner@erls.siemens.de	Joseph Meixner	SIEMENS	DE
jwindle@Foxboro.com	John Windle	FOXBORO	US
khaled.kassem@cran.uhp-nancy.fr	Khaled Kassem	CRAN University of Nancy	FR
lionelcerisier@yahoo.fr	Lionel Cerisier		FR
mcelenk@hotmail.com	Metin ÇELENK	YILDIZ Holding	TU
omer.akdeniz@frenchbatchforum.org	Omer Akdeniz**	SINFOR AUTOMATION	FR
p.levrard@arcinfo.com	Philippe Levrard	ARC INFORMATIQUE	FR
Patrick.Richard@aventis.com	Patrick Richard	AVENTIS	FR
paul.jones@eu.rhodia.com	Paul Jones	RHODIA	UK
pallot@ordinal.fr	Philippe Allot	ORDINAL TECHNOLOGIES	FR
quem@tpg.com.au	Spiro Georgakopoulos	QUEMM ASSOCIATES PTY LTD	AU
rdespins@foxboro.com	Robert Despins	FOXBORO	FR

rfp@optusnet.com.au	Robert Price	CSL Ltd.	AU
smmcleod_uk@yahoo.co.uk	Steve MacLeod	JACOBS ENGINEERING	UK
Tony@simmons.demon.co.uk	Tony Simmons	CONSULTANT	UK
viruega@ensgi.inpg.fr	Jean-Louis Viruega	ENSGI INPG	FR
willie.lotz@sabreweries.com	Willie Lotz	SOUTH AFRICAN BREWERIES	SA
yong.miao@aspentech.com	Yong Miao	ASPEN TECH	
yusaksp@yahoo.com	Yusak Pandiana	Technical University of Delft	NL

* Editor

** FBF WG4 chair

*** WBF “FA” WG chair

The following companies have sponsored this work:

SCHNEIDER ELECTRIC (Draft 1)

ROCKWELL AUTOMATION (Draft 2)

SIEMENS Automation & Drives (Draft 3)

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

INTRODUCTION

Although ANSI/ISA-88.01-1995 standard helps a lot in clarifying batch control, it can only be successfully implemented by experienced batch specialists. First implementations are proven to be always difficult and time consuming before taking all the benefits of S88 concepts.

Among difficulties, *exception handling* takes care of any situation that is not expected in normal operation when their probability of occurrence is above an acceptable level. It is often allocated as much as 80% of the design effort, leading to a huge number of discrete interlocks to prevent wrong operations.

The standard does not provide much information for the specification of *Control* and *Equipment modules* boundaries. Many S88 implementation reports raise the issue of *Process Cell* breakdown into its lower level components (i.e. *Units*, *Equipment Modules* and *Control modules*) and *exception handling*. From the "Top-Down" to the "Bottom-Up" through the mixed approaches, each project involves user preferences through specific, often inconsistent guidelines.

According to Spiro Georgakopoulos and Robert Price (see 2.2 Other documentation)

"**S88.01 – LIMITATIONS AND SOLUTIONS**

The S88.01 standard explicitly excludes offering any specific guidance in the area of low level decomposition. It expressly draws a line above the protective layer - *"total explanation of process segmentation is beyond the scope of this standard*". At the same time, the standard recognizes the difficulty and importance of decomposition to the whole modeling exercise -"*Effective subdivision of the process cell into well defined equipment entities is a complex activity, highly dependent on the specific environment in which the batch process exists*."

A poor decomposition can be recognized by a high number of dangling control modules that the batch server must monitor and manage. In addition, methodologies which treat the protective layer separately from the process result in a large number of interlocks that need to be enabled or defeated with each phase. Either way, the modularity aspect of S88 which promises so much is compromised in a tangle of "special cases".

There is a clear need of more practical guidelines beyond the general S88 concepts to build applications.

Flow analysis outcome is an efficient modularization that simplifies modules interactions, improves operation safety and dramatically reduces *exception handling* overhead. It may apply to any control models, however this report considers its application within S88 models.

The purpose of the WBF Flow Analysis Working Group is to develop guidelines based on current practices to complement S88 models to enforce physical and equipment *procedural model* design.

Editor's note:

Not sure it corresponds to the current scope

This work is intended to help people from process design, operation and process control communities to well understand these concepts and to communicate together, producing robust and safe applications on a consistent basis.

It should also help vendors to develop supporting tools based on consensual concepts.

BATCH CONTROL

ISA-88 MODELING USING FLOW ANALYSIS

1 SCOPE

This report focuses on design requirements for inherent actuators interlocks based on *flow analysis*. It addresses :

- *Process cell* equipment breakdown (typically at the *Control Module* level)
- *Exception handling* at the process actuator level

Other subsequent reports may develop other aspects of *flow analysis* in S88 control design.

The expected results will be an S88 compliant guideline to address specific issues encountered in actual implementations of the standard. The expected benefits are:

- Control Modules breakdown consistency leading to a high level of reusability
- Improvement of operation safety
- Easier *Exception Handling*

2 REFERENCES

2.1 Normative references

The following normative documents contain provisions, which through reference in this text, constitute provisions of this part of this report. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this part of this standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid normative documents.

IEC60902:1987, *Industrial-process measurement and control : Terms and definitions*

IEC61512-1:1997, ANSI/ISA-88.01-1995, *Batch control- Part 1: Models and terminology*

IEC61512-2:2001, ANSI/ISA-88.00.02-2001, *Batch control - Part 2: Data Structures and Guidelines for Languages.*

Note :

In this report, the ANSI/ISA-88 standard will be referred to as S88.

2.2 Other documentation

The following documents are related to existing flow-based methodologies. Although the present report uses fundamental elements from this documentation, it may not comply with the terminology and features used or expressed in them. However, the reader may find helpful information about the genesis of the methodology.

- "Methodological guide for the analysis of multipurpose batch process plants" RHONE-POULENC N° 501/93.875 - GET 5/EM/519/97/1119 - GT ASTRID
- "DELTA NODES Automated flows methodology" JMR Conseils N° JMR86/R01-2001_v3

- "A Flow Stream Approach for Process Cell Modularization" - By François Lebourgeois (Rhône Poulenc), Jean Michel Rayon (JMR Conseil) and Jean Vieille (Consultant) - Presented at the World Batch Forum Conference in Atlantic City (NJ, USA) on 23/04/2000
- "The Missing Link – A generic process control model" - By Spiro Georgakopoulos and Robert Price (Quemm Associates Pty Ltd) – Presented at the 6th World Congress of Chemical Engineering in Melbourne, Australia on 23-27 September 2001
- BatchML XML schemas developed by the World Batch Forum.

3 DEFINITIONS

For the purposes of this part of this report, the following definitions apply. All terms defined here are italicized in the text when their use corresponds to the definition. Abbreviations are included in the definitions. They are capitalized in the text.

3.1 Complex node

is made of several *nodes* of both directing and confining types.

3.2 Confining node

has the ability to contain the *flow*.

3.3 Connection

links *nodes* together through their *terminals*. It is not an equipment entity (as to flanges assembled together, each flange being owned by its corresponding *node*)

3.4 Directing node

has the ability to control the *flow* (direction or confinement).

Directing nodes support inherent interlocks that prevent erroneous automatic or manual handling according the safety needs.

3.5 Equipment procedural element (EPE)

Equipment procedural element is extensively addressed in S88 part 2. However, it is not formally defined in the current edition of S88.

Equipment procedural element is an element of the part of the S88 procedural model , which applies to equipment control. It can by either a *procedure*, a *unit procedure*, an *operation*, a *phase* or any user defined procedural entity. Its equivalent at the recipe side is the recipe procedural element.

3.6 Flow

the material or energy content in a *path*, i.e. the stream of product or energy that takes place within a set of equipment working together on it

3.7 Flow Analysis

a methodology based on an analytical review of processing equipment that aims at modeling equipment entities in a way that allows generic mechanisms to impose critical actuators to operate within specified constraints.

It is safety-focused, providing self-immune environment against automatic control and manual operation errors regarding material and energy handling.

3.8 Flow breaker

a device that has the ability to break the *flow*. It can be part of a *directing node*, part of a *complex node*, or an elementary *directing node*.

3.9 Node

a consistent equipment entity. It can correspond to any S88 equipment entity (Control module, equipment module, unit, process cell…), but would generally comply with Control Modules. It can be *confining*, *directing* or *complex* (mixed and embedding other sublevel *nodes*)

3.10 Path

the collection of *nodes* assembled to perform a functionality provided by an equipment procedural element. A *path* can support one or several *flows*.

3.11 Terminal

is the physical interface of a *node*

4 OVERVIEW OF FLOW ANALYSIS

4.1 A constraints based modeling

Process equipment is selected and installed with a clear expectation of the purpose it will serve in terms of physical and chemical processes it is to encounter. A process modeling exercise must recognize those constraints and incorporate them.

A critical constraint is the requirement that the material remains confined within a specified part of the process equipment. The operating context dynamically defines which part of the process equipment is allocated to each executing process oriented functions. A key performance measure of an operating process plant is how well material is contained within the process.

In selecting the equipment a number of physical and safety constraints are imposed, beyond which it cannot operate. In a plant as a whole, additional constraints derived from the interaction of equipment come in to play.

Finally, the product being processed will induce additional safety and operational constraints.

These constraints can be categorized as follow:

1. **Equipment** constraints may be physical things like capacity, maximum operating temperature, pressure, volume or flow. Or they may be a chemical (material) constraint – a vessel must never process material with pH less than 5. Or they may relate to hygienic status, etc.

   This information is attached to the equipment
2. **Product** constraints include things such as maximum temperature, blending compatibility.

   This information is attached to the product
3. **Relational** constraints between equipment, between products or between equipment and products. For example, possible connections between equipment determine assembling constraints; 2 products may be incompatible and shall never be in contact; a product may not be processed by certain equipment…

4.2 Intrinsically safe operational vs conventional interlocking

The following simple example below illustrates how *flow analysis* impacts on control system implementation and operational safety considering the confinement control aspect.

This example shows the difficulties that have to be faced of when dealing with flexible *process cells* to avoid cross contamination or pollution by opening a valve which does not contribute to the intended functionality (here, transferring product from T2 to T8).

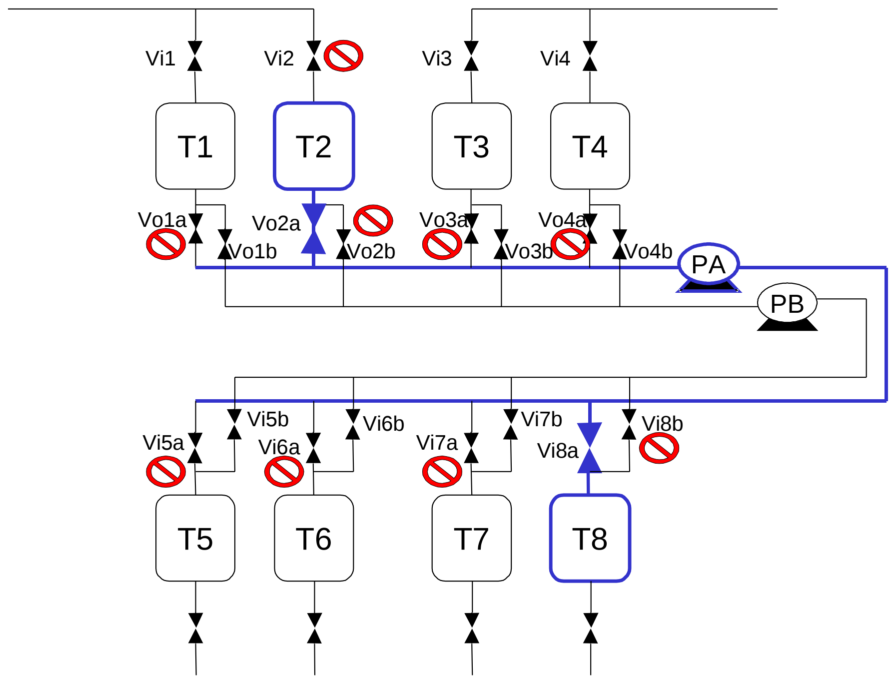

Figure 1: Flow integrity concern with complex networked process cells

Such process layout imposes to implement a protective layer, which must be effective when the operator manually controls the plant as well as in automated operation to avoid errors when controlling the different circuits.

In order to protect the circuit “A” transferring product from T2 to T8, valves Vo1a, Vo2b, Vo3a, Vo4a, Vi5a, Vi6a, Vi7a, Vi8b are to be forced to their closed position.

A conventional approach begins by a very detailed risk analysis and ends by

1) a huge list of interlocks such as :

IF PHASE “TRANSFER T2->T8” IS ACTIVE THEN

Vo1a FORCED CLOSE

Vo2b FORCED CLOSE

…

2) an equivalent set of starting conditions for *phases*:

INITIALIZING PHASE “TRANSFER T2->T8”

CHECK Vo1a CLOSED

CHECK Vo2b CLOSED

…

3) coordination control to suspend the *phase* if one of these valves fail

This extensive analysis, specification and coding must be performed for each combination.

Conventional approaches propose to simplify this design by interlocking *phases* instead of valves. However, this method constrains the operation flexibility, and does not prevent a wrong manual operation. Initial conditions / actuators failures are to be controlled anyway.

As an example, the following values were noted in a typical multipurpose chemical *process cell*:

- Number of *units* / shared *equipment modules*: 34
- Number of phases: 180

- Number of instruments: 600
- Number of interlocks: 1700
- Number of interlocks combinations: 13000

Flow analysis provides intrinsic, extensive interlocking for providing a better level of circuit integrity control than these numerous interlocks without any additional coding.

Conventional interlocking seldom reaches a high level of safety because of its exponential complexity. These interlocks may be dramatically affected when adding new equipment or defining new *phases*. (The number of combinations corresponds to the number of cases that are to be considered when adding/modifying equipment or *phases*.)

Smart designers implement matrices to define these interlocks. Coding becomes better, however other issues remain.

Finally, the validation effort is directly affected by the number of checks to ensure the system will behave as designed. Replacing discrete interlocks by a single validated FA mechanism dramatically reduces this effort.

4.3 Production Plant from Flow Analysis point of view

Flow analysis is directed at looking at physical (people, material, energy) *flows*. It does not address financial, or information flows.

- A production plant uses, moves, acts on 2 basic *flows*:
    - Material *flow*
    - Energy *flow*
- … using suitable equipment
    - Liquid material: Pumps, valves, pipes, vessels
    - Solid or Packed material: conveyors, lifts
    - Powdery material: roots, pipes, conveyors
    - Electrical energy: cables, bus bars, transformers, breakers
    - Thermal Energy (i.e. steam): pipes, exchangers…
    - Mechanical energy: agitators
    - …
- … performing specific or more complex process oriented functionality
    - Charging, Emptying, Filling, Distillation, Filtering
    - Heating, stirring,
    - …

  These actions
    - may have an implicit (i.e. charge a vessel, which end when the requested quantity is obtained) or an explicit ending (heating, stirring… which run forever until they are requested to stop).
    - may act on a single *flow* (charge, stirring), on several *flows* (in-line mixing)
    - may use one *flow* to act on another one (heating)

Any production system (i.e. a *process cell*) can be considered as a set of "closed sections" regarding the different material or energy *flows* that can occur into it (e.g. reactors, filters, pump sets). These closed sections are bounded by *flow breakers* (e.g. valves), which control their linkage with other sections.

Note :

Some parts of the production system cannot be considered as closed locally, but rather as infinite capacities that supply or consume product or energy. They represent Process Cell connections to raw or finished material, ancillary fluid or utilities (Sources and Sinks on the diagram).

Whatever is the part of the plant to be considered, any production system can be viewed as follow:

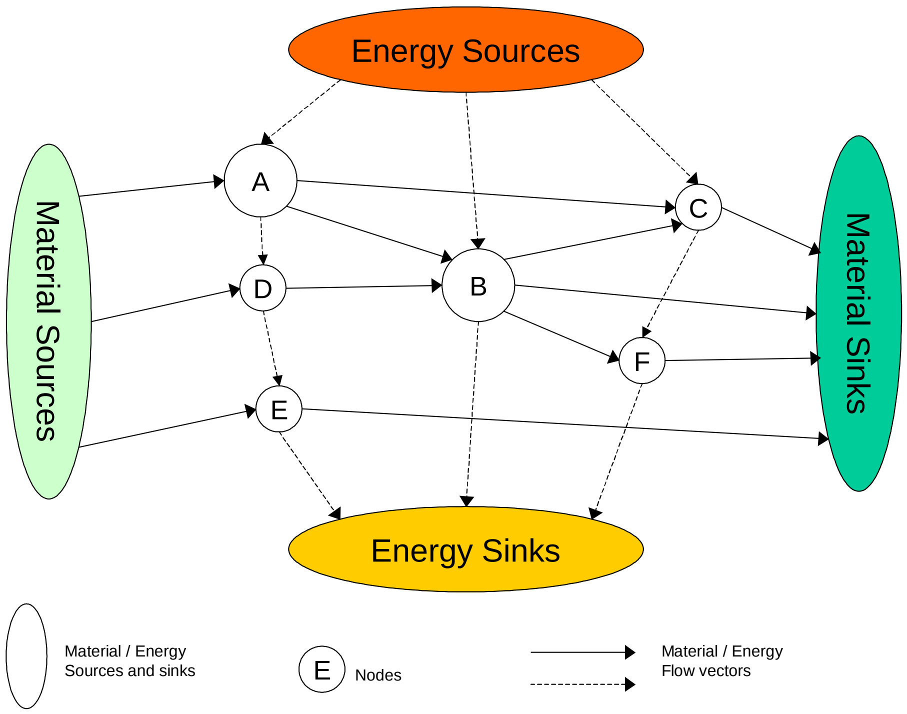

Figure 2: Flow Analysis conceptual view

The considered part of the plant (the studied production system) appears as a set of material and energy networks:

- Sources and sinks are the connection points that limit the scope of the analysis
- *Nodes* are the places were material or energy is transformed, stored, directed. They are actual equipment such as vessels, reactors, transformers, exchangers, pipes, valves, pumps, conveyors...
- *Flow* vectors indicates how the pieces of equipment are actually connected. The connections established by *flow* vectors define the actual plant layout. It may be fixed, but it may also be dynamically arranged for the purpose of the actual product processing requirements. The figure below shows free *nodes* that are not yet connected: equipment is available, but disconnected (imagine it lies disconnected on the workshop floor) and unable to fulfill any process oriented functionality (no material nor energy can be processed). It is not presumed to be assembled in a particular shape until it is used to make something. *Flow* vectors do not currently connect the equipment. They are waiting for any future usage.

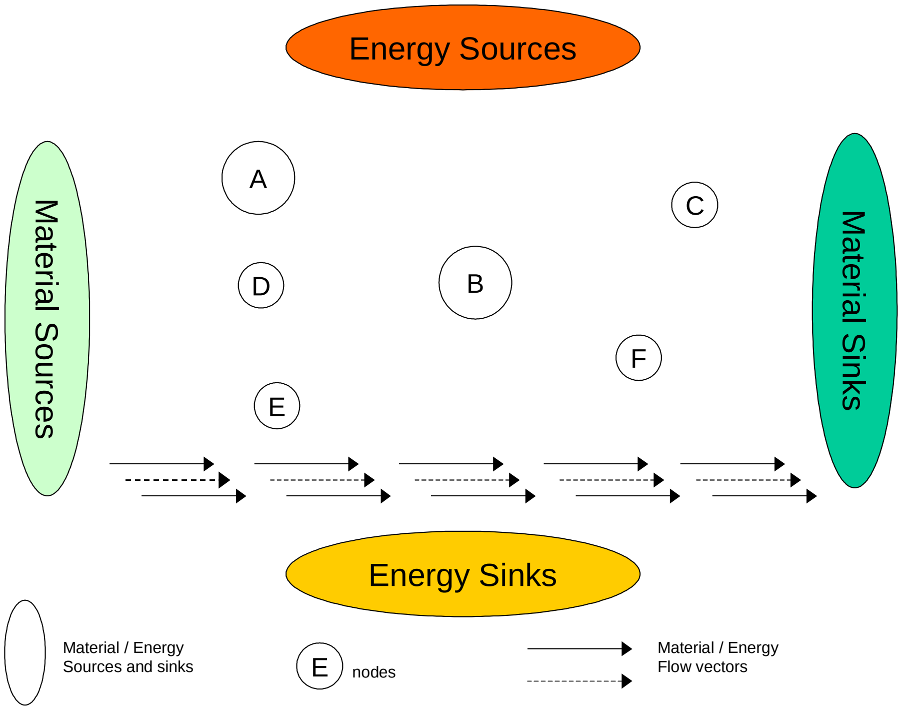

Figure 3: Free nodes unconnected by flow vectors

- A process functionality requires the support of a *path* by allocating several adjacent *nodes* (connected by *flow* vectors). The figure below shows the *path* P1 assembled to perform a process functionality using *nodes* A, B, F.

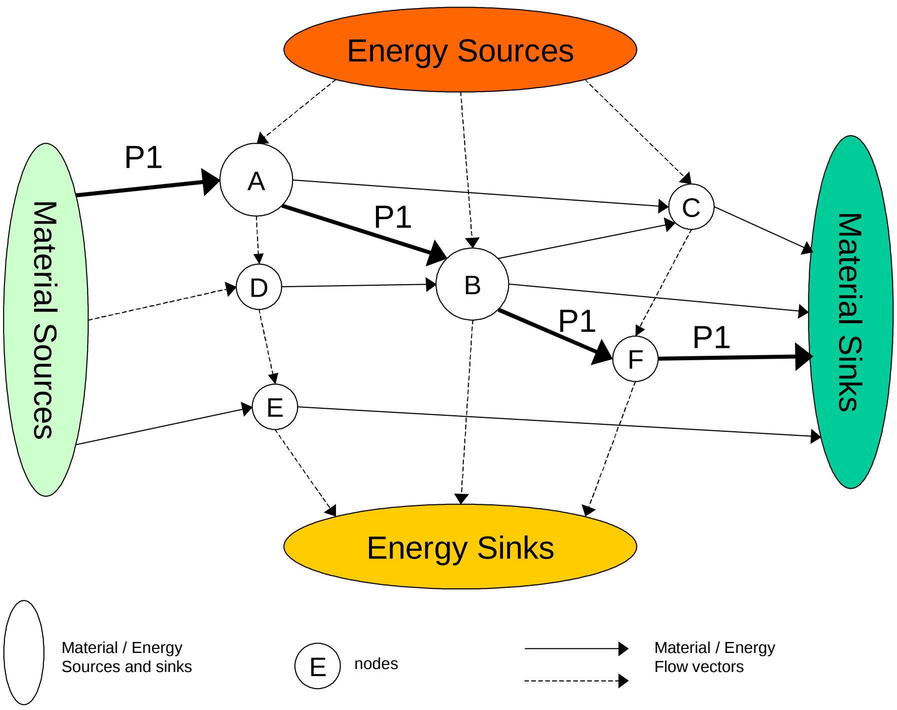

Figure 4: Nodes and vectors making a path

5 MODELING PRINCIPLES AND TERMINOLOGY

5.1 Basic principles

Analyzing *flow*s in a process cell may be done at different level:

- The connection analysis reveals how equipment is actually linked depending of *flow breakers* and *connections* statuses
- The static analysis concentrates on motionless *flows*, in order to check if static conditions allows the *flow* to occur (i.e. to open the corresponding *flow breaker*). It gives up once an actual *flow* is established.
- The dynamic analysis focused on operating *flows* to compare the actual performance to internal models in order to detect abnormal deviations and to predict expecting physical conditions.

This report focuses on operation safety, checking the conditions before the *flow* occurs to allow a *flow breaker* to open or a *directing node* to operate as requested. It is based on connection and static analysis.

5.2 Modeling a process cell

The following figure shows an example of a simple *process cell* from which basic concepts, terminology, and modeling rules can be presented.

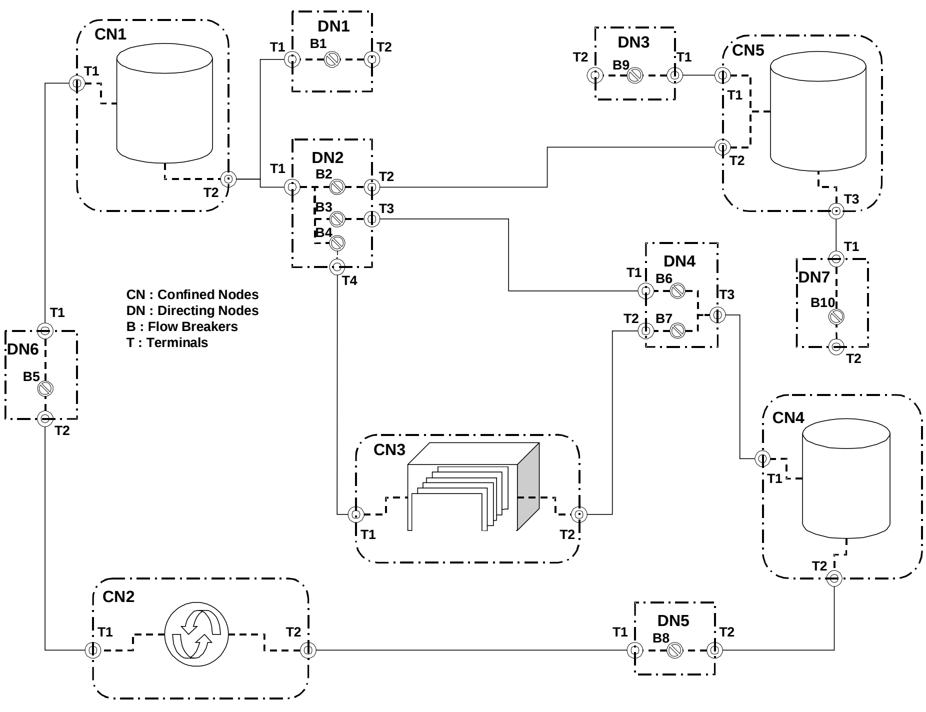

Figure 5: Process cell example 1

This *process cell* includes the following equipment:

- 3 main processing equipment (CN1, CN4, CN5) that could be qualified as *units* following S88 definitions.

- 2 in-line processing equipment (CN2, CN3) that could be qualified as *equipment modules* following S88 definitions.
- valves or group of valves (DN1-DN7) that could be qualified as *control modules* following S88 definitions. These valves are represented as no-way symbols but they can be removable pipes and/or blind flanges as well.

5.3 Nodes, Terminals and Connections

Figure 6 : Nodes, terminals, connections

All identified pieces of equipment are *nodes* which:

- Represent the recipient (the physical equipment) and its content (the product or energy inside)
- Have an informational interface provided by
    - properties attached to the recipient (equipment statuses and capability, product id…)
    - properties attached to the content (physical and chemical conditions)
- May have a behavioral interface provided by methods or functionalities that can be invoked by other entities
- Have a physical interface provided by *terminals*, which
    - hold specific properties and statuses,
    - can be linked to together by *connections.*

Node categories

Nodes can be categorized to address generic set of constraints and specific modeling rules. For example Astrid / Delta Nodes identifies the following categories:

- **Energy**

  Controls energy through the equipment (Reactor heating jacket, energy side of an exchanger, vacuum system, agitator)

- **Utility**

  Corresponds to common resources delivering or receiving fluids or energy to or from the *process cell*. It has no direct contact with the product being manufactured. It is generally shared by several *Units* or *Process Cells* (Compressed Air supply, N2 distribution, waste collection, venting system…)

- **Material**

  Participates to the product transformation or movement (Pipe, tank, reactor, transfer line, conveyor, product side of an exchanger…)

  Sub types of material can be defined. This can help to distinguish different *flows* through the same equipment:

    - Solid
    - Liquid
    - Gaseous

- Powdery
- …

For example, the liquid side (lower part) of a reactor is member of the “Material/liquid” *node* that can hold the product, while the gaseous upper part of the reactor is member of the “Material/gaseous” *node* controlling the pressurization. This allows dealing with 2-chemical phases equipment in order to allow parallel control using separate *equipment procedural elements*. Exclusive *node* allocation does not compromise functional flexibility.

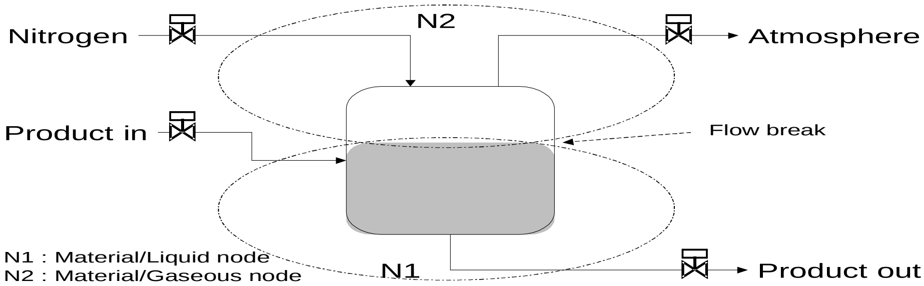

Figure 7: Liquid/Gaseous nodes interface

Node types

There are 3 types of *nodes*:

- *Confining nodes* which contain the *flow*
- *Directing nodes* which control the *flow* by their embedded *flow breakers*
- *Complex node*s which combines both *confining* and *directing nodes*

5.4 Directing nodes and flow breakers

Directing nodes (DN) have the ability to control the *flow* (breaking, directing). They generally cannot retain the *flow* inside themselves.

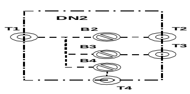

Figure 8: Directing nodes

They embed *flow breakers* that actually control the *flow*. *Flow breaker can be* the lowest possible breakdown of *directing nodes*. The following figure develops the previous one into elementary *directing nodes* encompassing single, simple *flow breakers*.

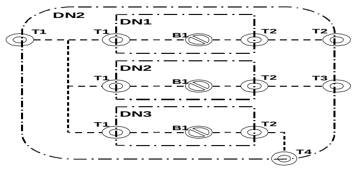

Figure 9: Ultimate directing node breakdown : Flow breakers

Depending on the user’s choice, *directing nodes* can be defined at the appropriate breakdown level (Figure 8 versus Figure 9).

Several *directing nodes* can be linked together to form a more complex *flow* control equipment.

Flow breaker can be controlled remotely or manually. They are devices such as

- one-way valves, multi-ways valves
- liquid / gas interface
- feeder or conveyor motor
- removable pipes, blind flanges or plugs
- doors or windows,
- traffic lights or mobile bridges
- …

5.5 Confining nodes (CN)

Confining nodes do not control the *flow*, but can contain it. They do not embed *flow breakers*, though they can influence the *flow* by embedded devices such as pumps or control valves.

All *terminals* of a *confining node* are “statically equipotent” i.e. if the *flow* doesn’t move and the *node* is not running any material or energy transformer, all physical conditions are the same for all *connections* (static analysis). This is generally not completely true: hydrostatic weight can influence the pressure depending of the altitude of the *terminal*. Also, the temperature can be different at different places and different times. The Figure 10 shows 2 different examples.

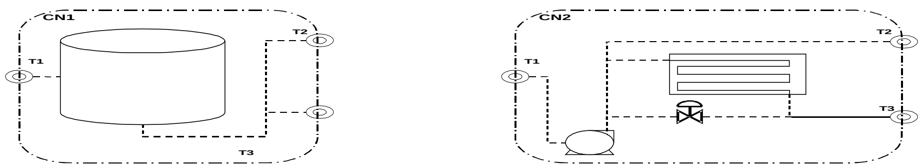

Figure 10: Confining nodes

Confining nodes can be combined into a bigger *confining node*.

5.6 Terminals

Terminals represent the physical interface of *nodes*. Nodes can be connected to each other by their *terminals* linked by *connections*.

Terminals can be:

- Inputs, if they can only receive an incoming *flow*
- Outputs, if they can only send outgoing *flow*
- Bi-directional if they can send or receive *flow* either.

Terminals can have other characteristics such as:

- type and size of *connection* mean (flange)
- Physical limits (max. pressure and temperature
- …

Directing nodes terminals have a particular “Flow” attribute that control the *flow* occurrence. This attribute has 2 values:

- Open
- Close

The “close” state means the *terminal* is a dead end. The *flow* does not go further. *Flow breakers* that are responsible for this state look at the *terminal* information to accept an opening order.

The “open” state means the *directing nodes* has established a way out for the *flow* toward another *node*, and there is at least one way toward a *confining node* (i.e. the *flow* can occur between 2 *confining nodes*).

Terminals are fundamental in referencing information relative to:

- the equipment they are attached to
- the product which *flows* through them: Product related information is acquired from the *terminals* to *terminals* relationships attached to the *node* for output *terminals*, or from other *terminals* through its *connection* to outside equipment for inputs.

Finally, *flow breakers* will operate under safe conditions that are assessed by looking at the information available at its *terminals*

The Figure 11 below shows information propagation through *terminals*. *Flow breaker* B1 is supposed to be closed. Because the ultimate goal is to protect operation from erroneously opening a *flow breaker*, there is no point to consider information propagation between communicating *nodes* (static analysis). We only need to propagate information at the *flow breaker terminals* from the corresponding *confining nodes*. Once the *flow breaker* has been opened, the resulting *flow* is no longer considered, and the *terminal* information is not requested anymore.

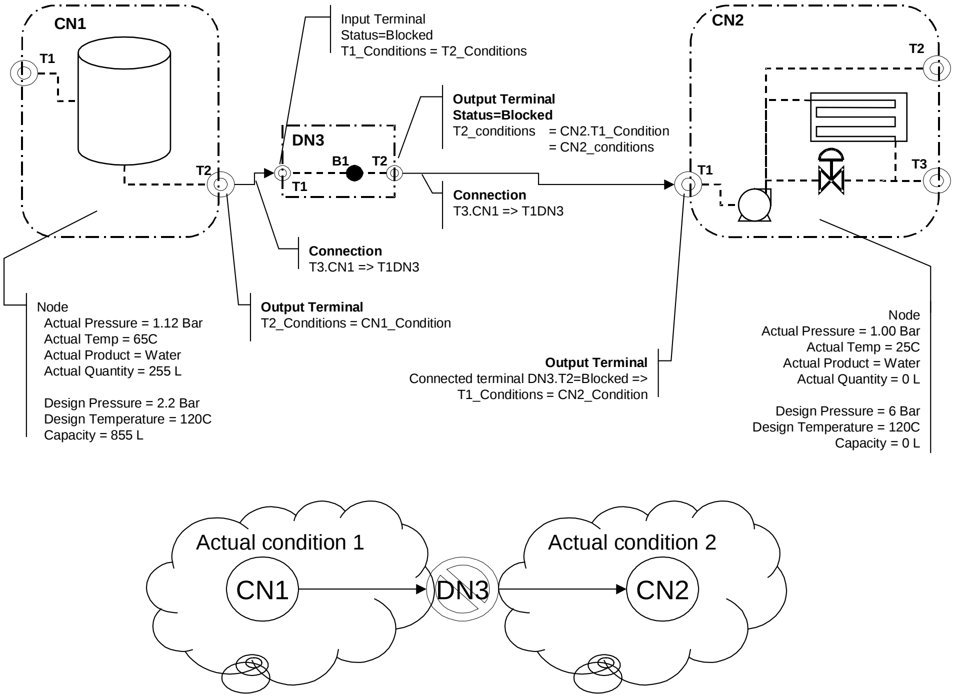

Figure 11: Information propagation

5.7 Connections

Nodes can be connected to each other using *connections* that link *terminals* to each other.

- In many cases, the process cell layout and its *connection*s are fixed.
- In other cases, *connections* are built at run time depending of the process to be executed.

Connections represent the fact that 2 physical entities (*terminals*) are purposely and adequately attached to each other. They aren't physical objects:

- A pipe which links 2 equipment is part of the *confining node* it is attached to (the side which is connected without valve).
- A removable pipe used to connect 2 pieces of equipment, is considered as a *directing node* with 2 *terminals*.

Connections shall fulfill the following requirements:

- All *confining nodes terminals* are to be connected to either another *directing node*. This allows the process to containing the *flow*, because it guarantees that it is bounded by *flow breakers*.
- *Connections* can only link compatible *terminals*:
  - *Terminals* can be input, output or bi-directional
  - An input *terminal* can be connected to only one output *terminals*
  - An output *terminal* can be connected to several input *terminals*
  - A bi-directional *terminal* has to be properly connected according its current use
  - Connected *terminals* shall have compatible characteristics

The main role of a *connection* is to link *terminals* in order to let the *nodes* communicate the actual process conditions downstream the *flow*.

5.8 Complex nodes

5.8.1 Grouping nodes together

In the previous example the process cell was broken down in elementary pieces were elementary *flows* can be revealed.

The concept of *nodes* is flexible and can describe any physical hierarchy. For example, the following figure represents the same process cell, but identifies “macro *nodes*”, which are consistent with the mechanical engineering (each *node* corresponds to separately engineered modules, or skids delivered by different subcontractors)

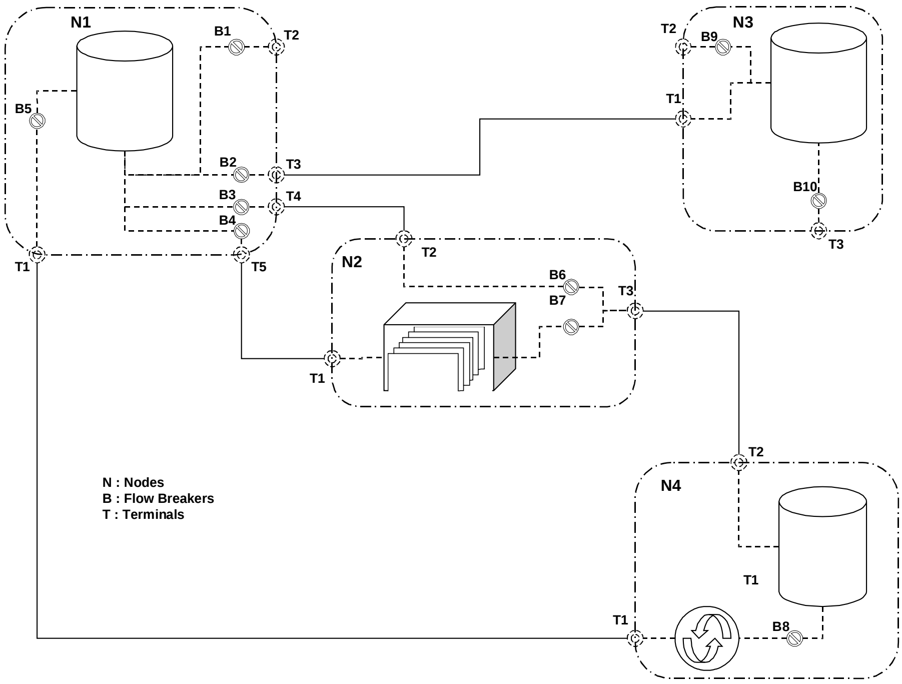

Figure 12: Higher level process cell modularization

Complex nodes do not allow the best flexible and accurate *flow* management (for example to deal with bi-directional *connections*), because the relationships between *terminals* are too complex to be efficiently modeled. However, if bounding *flow breakers* are identified as *nodes*, it becomes possible to control incoming and outgoing *flows*. This may be useful when integrating packages.

Complex nodes are obtained by increasing the modularization granularity (see 5.8.2). It dramatically simplifies the control system implementation by reducing the number of modules.

At the highest level, a plant may be analyzed the same way :

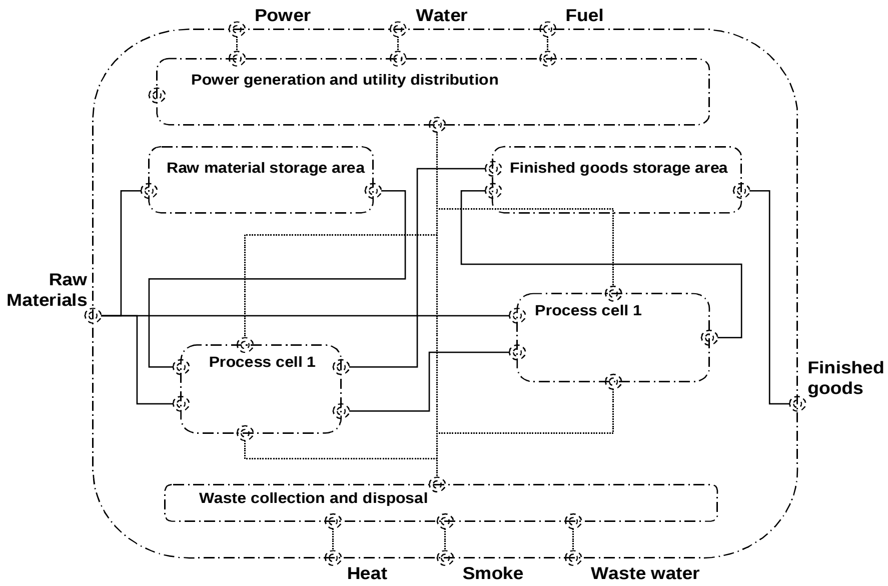

Figure 13: The whole plant as interconnected nodes

5.8.2 Nodes Breakdown principles

Material or energy *flows* are supported by *Paths*. A *node* represents a part of a *Path,* acting as a container of material or energy. The breakdown of the facility into *nodes* is made by applying the following rules:

- Breakdown rules (BK) give guidance to split the *process cell* in *nodes*. Strictly applied, these rules leads to the most refined breakdown and results in the highest number of *nodes*
- Aggregation rules (AG) give guidance to possibly reduce the number of *nodes* (end the subsequent control system complexity) in grouping equipment entities that form a set of equipment, which works independently.

The following table presents the rules that can be applied at the lowest breakdown level (the *nodes* the control system will have to deal with, regardless they are combined together to form higher level entities).

Rule	Type	BK	AG	Description, Comments
Closed (confined) section	Confining	X		A *node* is bounded by *flow breakers* Example, this rule is mentally applied to piped installation by “blowing” in any part of the cell while *flow breakers* are operating (valves are closed). The identified *node* is the under-pressure part of the cell.
Flexibility	All		X	When grouping *nodes*, the resulting diminished flexibility shall be challenged by checking if different parts of the same *node* could be used by different *phases* at the same time.
Inter-CMs Constraints	All		X	Aggregation releases constraints enforcement between aggregated *nodes*. A coarse breakdown may lead to lose the benefit of secured operation, which is basically the intent of the *Flow analysis*.

Reusability	All		X	Looking for reusability suggests to isolating equipment subsets that are fully standards and match the actual equipment once assembled. The number of different *nodes* increases with aggregation. The designer shall compromise between reusability and the number of *nodes*.
Virtuality	Confining	X		It is sometime difficult to identify the equipment the *node* represents. That could be the case for controlled physical measurement that are not part of any mechanical part of the equipment acting on one or several actuators. (Example: a remote vision system)

Table 1: Nodes breakdown rules

5.8.3 Grouping directing and confining nodes

It may be convenient to group *confining* and *directing nodes* in *complex nodes* that can be consistently managed as autonomous equipment entities (Control Modules). As a side result, *directing nodes* are seldom found as separate entities. An outstanding drawback of this grouping is that it cannot comply with bi-directional and merging *flows*.

The resulting *complex nodes* forms the S88 control module layer or the Astrid / Delta Node Resource layer. The following table proposes the rules to attach a particular *directing node* to the appropriate *confining node*.

Attachment Rule	Node Type	Comment
Upstream driven	Material Sky	This rule applies to finite *nodes* corresponding to process related *flows*. The *directing node* is attached to the upstream *confining node* considering the *flow* direction.
Process side driven	Energy Utility	This rule applies to *nodes* that control a shared utility to the process. The *directing node* is attached to the process side *confining node* whatever the direction of the utility fluid is.

Table 2: Directing nodes attachment rules

5.9 Generic model

Nodes are connected to each other, and embedded within higher-level *nodes*.

Depending on the primary breakdown, *nodes* can be elementary *confining* or *directing nodes*, as well as combined *directing nodes* or *complex nodes*.

Nodes hierarchy and internal relationship is modeled until the lowest-level defined *nodes*. At the lowest level, it becomes a black box with *terminal* internal relationships.

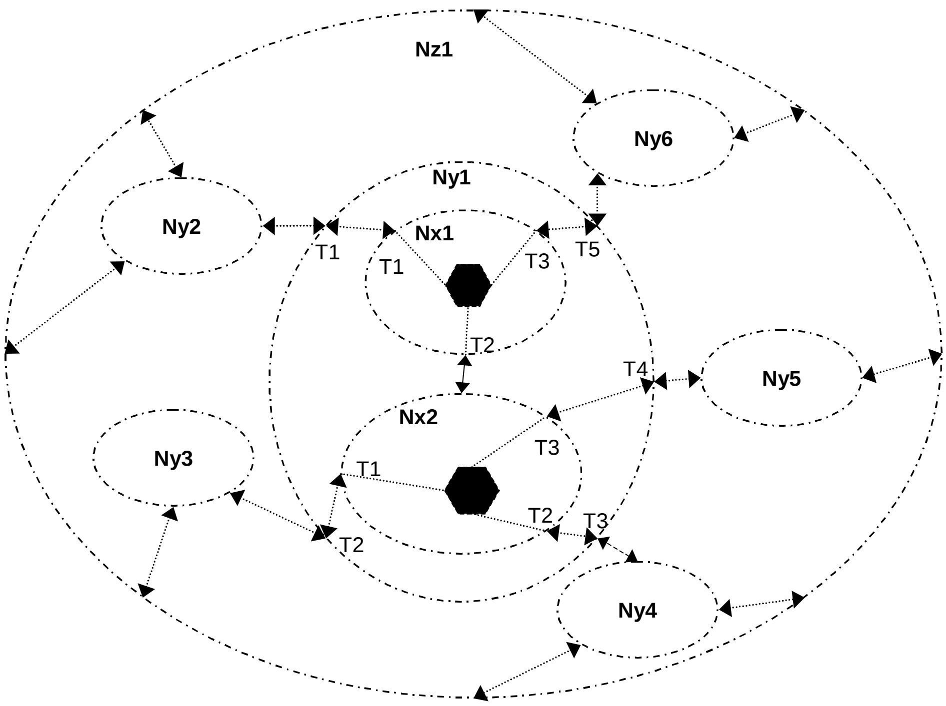

Figure 14: Recursive Nodes - Generic model

5.10 Mapping FA entities on S88 equipment entities

As shown above, any physical entity can be interpreted as a *node*. All S88 equipment entities physical can be represented as *nodes* whatever is their level in the S88 physical model.

The application field of the *flow analysis* is widely open. However, for the purpose of this report, we will concentrate at the lowest physical level where the *flows* are actually controlled, in order to support the protective layer that will secure the actuator operation.

Typically, the finest *nodes* breakdown will correspond to Control Modules.

5.11 Functional view: Equipment Procedural Entities

In order to make a product, a production system must:

1. Provide the required basic process functions its equipment is capable of doing (EPE, *equipment procedural element*). In S88 EPEs can be Phases, Operations, Unit Procedures, Procedures.
2. Orchestrate these EPEs following the product processing rules (Recipes, made of Recipe Procedural Elements or RPEs)

This is the S88 concept of separation between equipment control and process control. *flow analysis* may interprets EPEs as follow (the term *node* is replaced by *control module* :

- In order to execute a specific EPE, the needed control modules (initially considered lying dismantled on the floor) need to be assembled to establish a *path*. The *path* represents the physical extent of its execution and is exclusively allocated to its initiating EPE.
- An EPE can execute only if its *path* can be initialized i.e. the corresponding *Control Modules* are available to its use - not currently used by another EPE allocating an overlapping *path* or disabled for whatever reason - and the necessary *connections* are feasible.

- A *path* may enable one or more *flows*. Any combination of *flows* in a *path* is allowed from the control point of view; however, it should be consistent with the mechanical reality (*connection* check).

As stated in the scope section, this report only concentrates on the protective layer and basic control. The functional (procedural) aspects will not be further discussed here.

6 OBJECT MODEL AND ATTRIBUTES

6.1 Overview

When dealing with an actual application, designers may take benefit of *flow analysis* to help modeling and implementation. Examples of applications examples are given in annexes.

This section presents the detailed FA elements in order to facilitate design and implementation.

6.2 Control Module Object Model

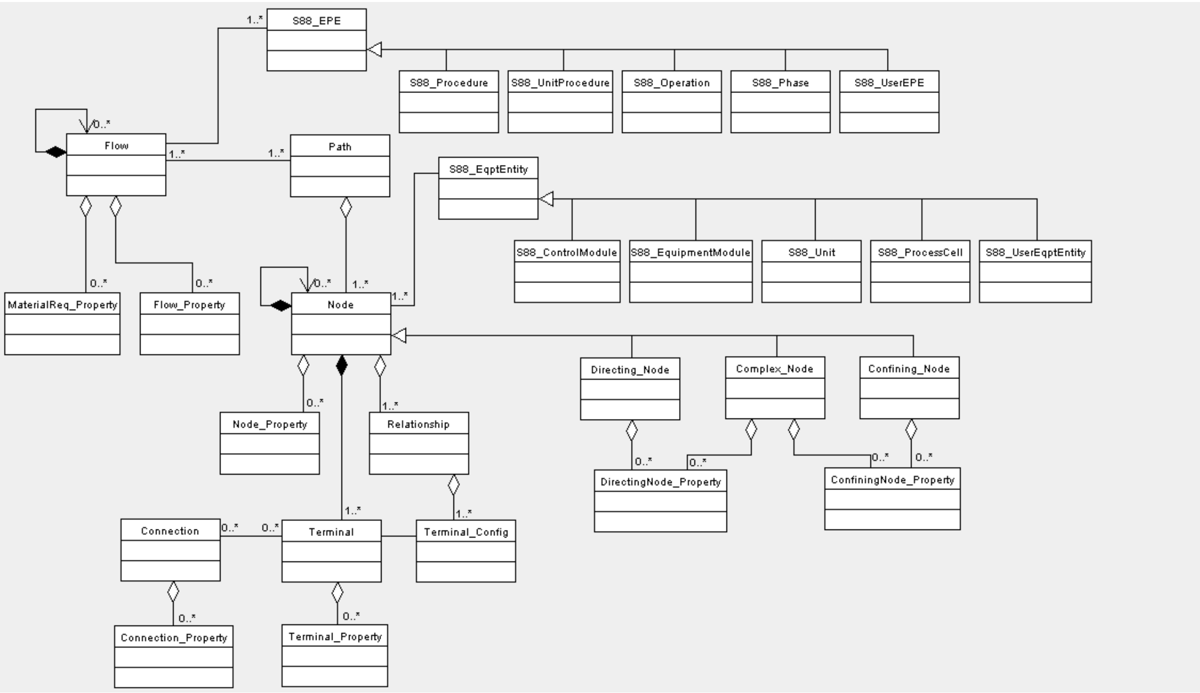

Figure 15: Object model

6.3 S88 Objects

The object model starts by the S88 equipment entity, which is specialized into control module type, equipment module type, unit type or process cell type, as well as any user defined level (extensibility of S88 models).

Whatever they are, S88 entities may be associated to *nodes*.

> *Note : Another option could have been to merge S88 Equipment Entity and Node concepts, which are pretty similar. However, such an approach would have been more complex because types of nodes would have overlapped with S88 physical level. It also allow flow analysis concept to be used with different concepts than S88.*

The corresponding objects are not defined in this report:

- S88_EqptEntity
- S88_ControlModule
- S88_EquipmentModule
- S88_Unit

- S88_ProcessCell
- S88_UserEqptEntity
- S88_EPE
- S88_Procedure
- S88_UnitProcedure
- S88_Operation
- S88_Phase
- S88_UserEPE

6.4 Attributes and Properties objects

In the model each object has a set of generic attributes. These attributes may be optional or required.

Note :

Data types are not defined, interoperability is not address in this report (and shall not be a concern within the scope of this report).

Properties are process related information. All property objects are based on the same meta-class

Property meta-class object attributes		
Attribute Name	**Description**	**Examples**
ID	An identification of a property of the associated *object*.	
Description	Contains additional information and descriptions of the *property*.	
UOM	The unit of measure of the associated property value, if applicable.	
DesignValue	Design or typical value, set of values, or range of the property.	
CurrentValue	Current value of the property	
MaxValue	Maximum value of the property	
MinValue	Minimum value of the property	

Table 3: Property object attributes

This meta-class defines the design information (design, maximum and minimum values), as well as the current value of the same information. This is the basic information to check flow consistency by comparing design and actual values:

- design value of the node at the downstream side of a flow breaker against actual value of the node at the upstream side of the flow breaker
- Actual and design values within the same equipment

The following classes are based on this meta-class:

- Flow_Property
- Material_Property
- Node_Property
- DirectingNode_Property
- ConfiningNode_Property
- Connection_Property
- Terminal_Property

Properties can also be used for additional information as a mean to extend the model without compromising its consistency and easily identify user / implementation specifics.

6.5 Path object

The path establishes the relationship between a flow and the nodes.

Editor's note:
Do we need the path object?

6.6 Flow object

A *flow* shall correspond to an S88 *equipment procedural entity*, which is the S88 equipment process oriented functionality. (Astrid / Delta Nodes gives use both "Functions" and "Flow" terms do designate the functionality and the flow which is built to uniquely support it).

This object is presented on the model, because it holds material requirement properties, which are needed to control the proper conditions of the allocated *nodes*.

Nodes are actually acquired by *flows* at run-time. If the acquired *node* is made of lower level *nodes*, the owning flow information is propagated downstream in the hierarchy, allowing any *node* to control its current conditions against material requirements.

In actual implementations, several EPEs may run together to form a given *flow*. It is the case when a TRANSFER phase is synchronized with a RECEIVE phase on 2 different units. This is represented by a *flow* containing 2 lower-level *flows*, each of them being associated to the corresponding phase.

Flow is a recursive structure, and each *flow* may be related to 0 or several EPE.

6.6.1 Flow object attributes

Node object attributes		
Attribute Name	**Description**	**Examples**
ID	An identification of the object.	
Description	Contains additional information and descriptions of the object.	
S88_ProceduralLevel	Specifies the S88 EPE procedural hierarchy level	Possible values are: - Procedure - Unit procedure - Operation - Phase
BatchID	Batch identifier.	
Product ID	Product identifier	
OperatorID	Operator Identifier.	
ActivationStatus	Activation status of the *flow*.	Possible values are: - IDLE - RUNNING - STOPPED - HOLD - FAULT
ActivationSubStatus	Additional identifier, documents the ActivationStatus. (Application specific values) It identifies specific running behaviors or mitigation methods	

Table 4: Node object attributes

6.6.2 Flow properties

To be completed

Example of flow properties		
Property Name	**Description**	**Examples**

Table 5: Flow properties

6.6.3 Material requirement properties

Material requirement information is attached to the *flow* because this information is defined in the recipe and has to be downloaded into EPEs/*flows* at run-time. It propagates from the Recipe Execution domain to the operating EPE/*flow* and finally to the acting control modules to feed *flow* control interlocks.

Example of material properties		
Property Name	**Description**	**Examples**
Quantity	Quantity of product to be processed	
Specific mass		
SizeL		
SizeW		
FlowRate	Required flow	
Temperature	Required Temperature	
Pressure	Required pressure	
PH	Required PH	
…	*Note :* *Other attributes may be defined by the user*	

Table 6: Material properties

6.6.4 Flow object operations

Any *flow* can have incorporated operations that can be invoked for performing procedural control as EPEs.

There are no specifics regarding *flow analysis* regarding procedural control.

6.7 Node object

6.7.1 Types of nodes

Nodes can be specialized in 2 types: *directing nodes*, *confining nodes*.

Node structure is recursive; a node can be made of other *nodes*:

- *Directing nodes* can content other *directing nodes*. At the last level, they content *flow breakers*
- *Confining nodes* can content other *confining nodes*
- A *complex node* is made of both type of *nodes*

Information attached to the *node* (attributes or properties) is mainly equipment related.

6.7.2 Node object attributes

Node object attributes		
Attribute Name	**Description**	**Examples**
ID	An identification of the object.	
Description	Contains additional information and descriptions of the object.	
S88_EqptEntity	Corresponding S88 equipment entity	
S88_EqptLevel	Specifies the S88 physical hierarchy level.	Possible values are: - Control Module, - Equipment module, - Unit, - Process Cell
ContainedIn	*Node* this node is contained in	
Type	Type of the *node*. The complex type can be self-specified if the *node* contains both type of *nodes* or a *complex node*. It has to be specified if the breakdown does not model the *node* components.	Possible values are: - Confining - Directing - Complex
Category	- Category of the *node*. Other values can be defined in properties	Possible values are: - Material/Powdery - Material/Liquid - Material/Solid - Material/Gaseous - Energy - Utility
BatchID	Batch identifier. - It may be set by the owning EPE and shall propagate downstream in the *node* hierarchy - It may by set by the *directing nodes terminals* and shall propagate upstream in the *node* hierarchy. Multiple batch identifiers may result in *flow* merging depending on the *node* composition and equipment usage	
ProductID	Identifier of the last product the node contained	
FlowID	The *flow* identifier is set by the acquiring EPE. It propagates downstream in the *node* hierarchy.	
OperatorID	Operator Identifier. It is set by the owning EPE It propagates downstream in the *node* hierarchy	
ActivationStatus	Activation status of the *node*.	Possible values are: - STOPPED - RUNNING - FAULT
ActivationSubStatus	Additional identifier, documents the ActivationStatus. (Application specific values) It identifies specific running behaviors or mitigation methods	
OperationalStatus	Status of the current operational conditions	Possible values are: - Normal operation - Direct manual operation - Broken down - Under maintenance - Repaired
SanitaryStatus	Cleanness, sterility, sanitary Status.	Possible values are: - clean - contaminated - sterile - unknown - Product

1

2

Node object attributes		
Attribute Name	**Description**	**Examples**
…	*Note :* *Other attributes may be defined by the user*	

Table 7: Node object attributes

These attributes apply to any *node* whatever is its type.

6.7.3 Node properties

Node properties include the physico-chemical information of the equipment: are configured at the engineering level providing additional, application specific information that describes:

Example of node properties		
Property Name	**Description**	**Examples**
Capacity	capacity of the equipment	
Weight		
Level		
Volume		
DimensionL		
DimensionW		
Flow	Flow of the equipment	
Temperature	Temperature	
Pressure		
PH		
…	*Note :* *Other attributes may be defined by the user*	

Table 8: Node properties

6.7.4 Node object operations

Any node can have incorporated operations that can be invoked for performing elementary actions.

There are no specifics regarding *flow analysis* at this level (however, *directing nodes* have specifics regarding *flow* control)

6.8 Directing Node object

6.8.1 Directing node object attributes

Directing node object attributes are specific information about this type of *nodes*

Directing_Node object attributes		
Attribute Name	**Description**	**Possible values**
OperationMode	The mode of operation is the way the device is controlled. Unlike most methods, *flow analysis* is concerned by all actuators regardless they are connected to the control system or not. -	Possible values: - **Handled** (driven by the operator - no connection with the control system) Ex:A blind flange - **Controlled** (driven by the control system). Ex: actuated on-off valve - **Virtual** (there are no corresponding device to control). Ex: a glass fiber can be broken at the outlet of the glass oven, corresponding to a "virtual" *flow breaker*
…		

Table 9: Directing node object attributes

6.8.2 Directing node properties

To be completed

Example of directing node properties		
Property Name	**Description**	**Examples**

Table 10: Directing node properties

6.8.3 Directing node operations

Directing nodes actually controls the *flow*. It gets information on conditions at its *terminals* and takes the appropriate protecting decision depending of them.

The Figure 16 shows the different control a *directing node* is under. *Flow* control is actually part of the *directing node*. It acts as an interlock between automatic/manual control and the actuators.

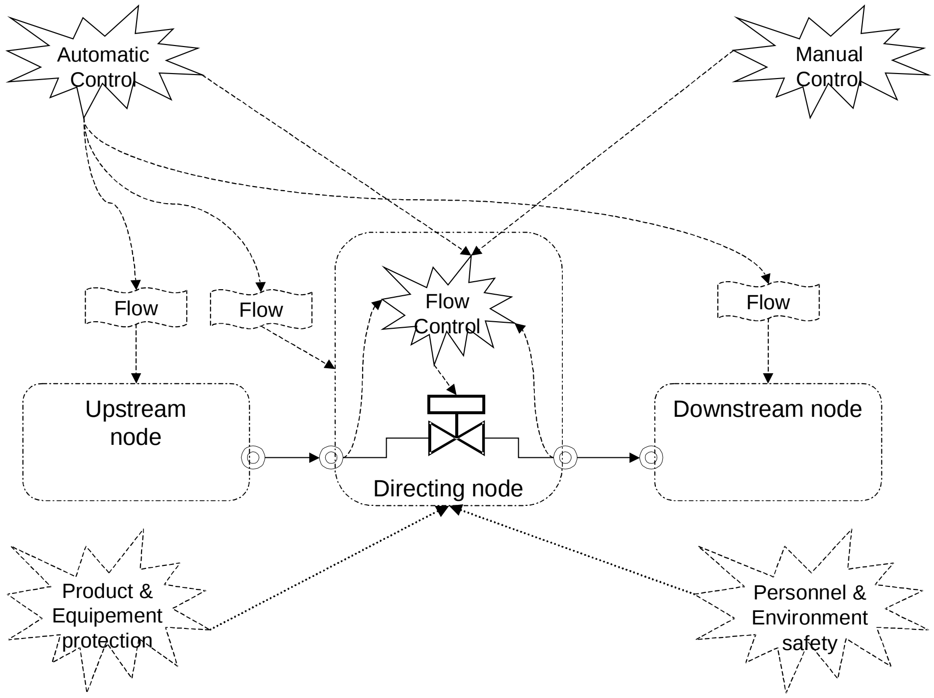

Figure 16: Directing node control

Directing nodes support the *flow analysis* safety layer. This safety layer act as an interlock between automatic and manual control, and the actuators within the *node*. They perform the following *flow* control operations:

- Set the direction
  - Basic control that acts on *flow breakers* and other actuators

Before establishing a flow

- Control the equipment compatibility
  - Check for the invoked *terminal* relationship if the “open” *terminals* have compatible statuses with *terminals* of the connected *nodes* (a “PayReceive” terminal shall be connected to either a “MaySend” or a “MaySendOrReceive” terminal
- Control the *flow* integrity
  - Check if the nodes that will be linked are part of the same flow
- Control the product-process / equipment compatibility
  - Check if the process conditions of the nodes that will be linked are compatible with the equipment capabilities (ex: maximum temperature) and status (ex: clean)
- Control the product-process compatibility
  - Check if the products that will be mixed are compatible

When the flow is estblished

- Control the process conditions against material requirements
  - Check if the operating conditions are compatible with the material requirements

To be developed

6.9 Confining node object

6.9.1 Confining node object attributes

To be completed

Confining_Node object attributes		
Attribute Name	**Description**	**Examples**
…		

Table 11: Confining node object attributes

6.9.2 Confining node properties

To be completed

Example of directing node properties		
Property Name	**Description**	**Examples**

Table 12: Confining node properties

6.9.3 Confining node object operations

To be completed

6.10 Complex node object

To be completed

6.10.1 Complex node object attributes

To be completed

6.10.2 Complex node properties

Complex nodes don’t have specific properties. They can be either *directing* or *confining nodes* properties.

6.10.3 Complex node object operations

To be completed

6.11 Terminal object

6.11.1 Terminal object attributes

Terminal object attributes		
Attribute Name	**Description**	**Examples**
ID	An identification of the object.	
TerminalStatus	Status of the *terminal* regarding the *flow* that determines if the *terminal* is or not an end of the *flow*. An open *terminal* indicates that the *node* is part of the *flow*, but not the last. A close *terminal* indicates this *node* is an end of the *flow*.	Possible values are: - Open - Close
FlowStatus	Status of the *terminal* regarding the *flow*.	Possible values for a closed terminal: ▪ MaySend ▪ MayReceive ▪ MaySendOrReceive Possible values for an open terminal: ▪ Receiving ▪ Sending
…		

Table 13: Terminal object attributes

6.11.2 Terminal properties

To be completed

Example of terminal properties		
Property Name	**Description**	**Examples**

Table 14: Terminal properties

6.11.3 Terminal object operations

To be completed

6.12 Relationship object

Relationships object defines the possible ways through the *node*.

These relationships are normally associated with / activated by *node* operations, which set the different *flow breakers* (actuators) patterns.

At least one relationship shall be defined for each *node*.

Relationship may be active or inactive (at run-time).

Active relationships determine all actual *node* internal *connections* between its *terminals*.

If no relationships are active, default relationship apply .

Several relationships may be active at the same time. That means that several separated flows All relationships sharing at least 1 terminal are all connected together.

Example:

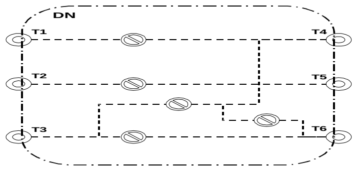

Figure 17 : Internal node relationships

The following table represents all possible Relationships:

	Terminal 1	Terminal 2	Terminal 3	Terminal 4	Terminal 5	Terminal 6
RelationShip 1						
RelationShip 2						
RelationShip 3						
RelationShip 4						
RelationShip 5						
RelationShip 6						
RelationShip 7						
RelationShip 8						

Table 15: Possible node relationships

For example, relationship 1 lets communicate terminal 1 and 4

6.12.1 Relationship object attributes

Editor's note
The model only consider simplistic relationships between terminals. More complex relationships regarding physical information transformation (flow, temperature, pressure…) are not taken into account. Is that a problem?

Relationship object attributes		
Attribute Name	**Description**	**Examples**
ID	An identification of the object.	
ActiveStatus	Flag for the active relationship. Several relationship can be active at the same time	Possible values: - True - False
DefaultRelation	Identifies the relationship that is enable when no relationships are invoked (i.e. the ActivationStatus of the node is "stopped", or if the node is not controlled) Several relationship can be default	Possible values are - True - False

Table 16: Relationship object attributes

6.12.2 Relationship object operations

To be completed

6.13 Terminal configuration object

Terminal configuration set the attributes to all *terminals* for each identified relationship. The attachment of terminals to relationships must be more

6.13.1 Terminal configuration object attributes

Terminal configuration object attributes		
Attribute Name	**Description**	**Examples**
TerminalID	Identification of the terminal	
TerminalStatus	Set the corresponding terminal status. All unused / blind terminals are set to "Close". Other are open	Possible values are: - Open - Close
FlowStatus	Set the terminal ability regarding the direction of the flow	Possible are: - MaySend - MayReceive - MaySendOrReceive

Table 17: Terminal configuration object attributes

6.13.2 Terminal configuration object operations

To be completed

6.14 Connection object

Connection objects defines links between *nodes*. It tells a *terminal* which other *terminals* from external *nodes* are connected to it.

I allows the *directing node* to know the actual conditions in front of each closed *terminal*, in order to check *flow* consistency and authorize the corresponding *flow breaker* to operate.

6.14.1 Connection object attributes

Connection object attributes		
Attribute Name	**Description**	**Examples**
ID	An identification of the object.	
Terminal1	First *terminal* the *connection* links	
Terminal2	Second *terminal* the *connection* links	

Table 18: Connection object attributes

6.14.2 Connection properties

To be completed

Example of connection properties		
Property Name	Description	Examples

Table 19: Terminal properties

6.14.3 Connection object operations

To be completed

7 IMPLEMENTING FLOW ANALYSIS

This section provides guidance in order to facilitate FA implementation in control systems.

To be completed.

Editor's note:

- *Shall include an actual example and possibly its implementation.*
- *Shall address the possible overhead when implementing the full methodology*
- *Shall discuss the location of all information attached to the objects (within the PLC/Controller, from an external data-base…) While I thought it was a design / integration only issue, several comments revealed that it is already a concern at our conceptual level*
- *Shall discuss Internal exception handling*