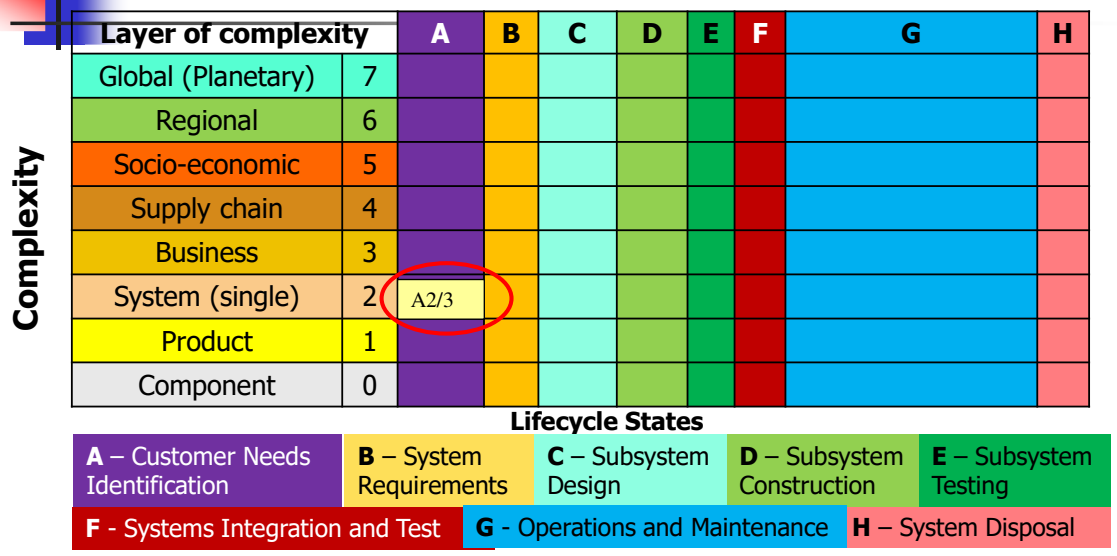
Module 13: Modelling, simulation and using models in building a model over the SDP
Session 3 of 7

Rev 2.1.0

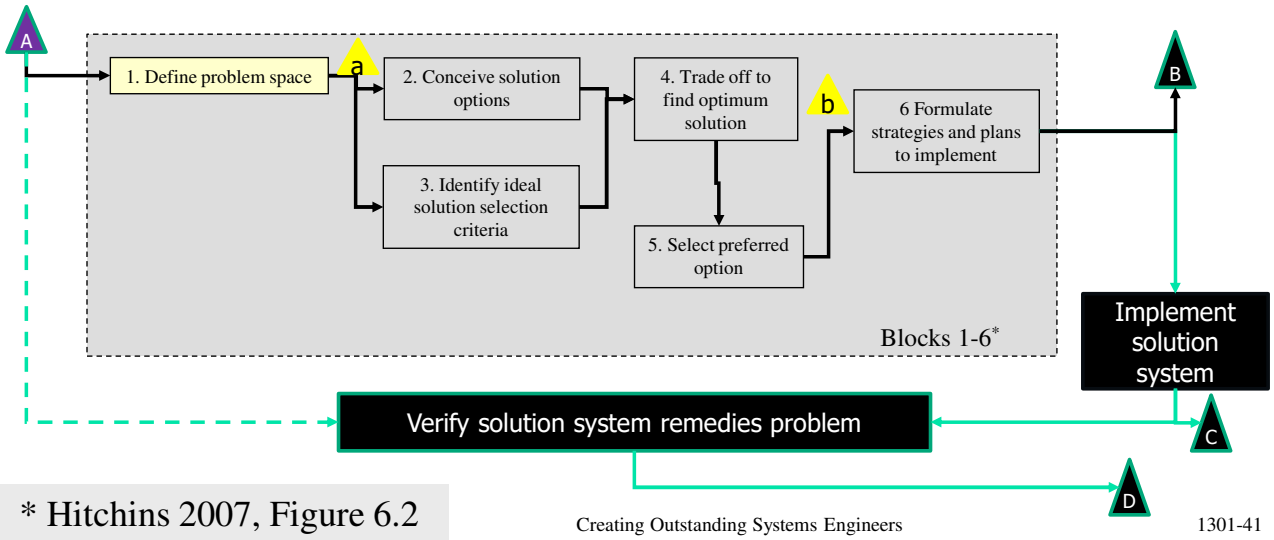
Needs Identification State



Column A

1. This sub-state contains the set of activities that explore/scope the problem, leading directly to Phase A.2. The activities performed in this phase produce a definitive statement of the problem-in-context.
2. **This sub-state contains the set of activities that conceive the whole solution system (which 'emerges' from/"complements" the problem) and produces the concept of operations (CONOPS) that describes how the solution system will operate in its future environment.**
3. This sub-state contains the set of activities that design the whole solution system, identify the environment, other interacting systems, the subsystems, parts, interactions, functional architecture, physical architecture, etc., etc., - but still all of the whole.

HKM²F Column A2 The Needs State (problem language)



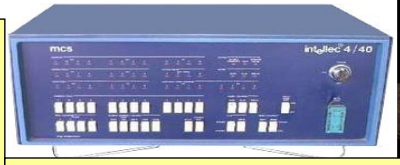
* Hitchins 2007, Figure 6.2

A.2

- Problem
  - Produce a CONOPS for the simulation – not design the simulation
- Solution CONOPS
  - Use before contest
    - to practice strategies for when to operate on which bands
      - To contact Sections when propagation is possible
      - Knowing which Sections are ***still needed, adjust strategy*** so as to re-plan when to operate on which bands, namely do I ?
        1. Go for Section multiplier
        2. Go for higher contact rate
        3. Go for both

CONOPS (hardware)

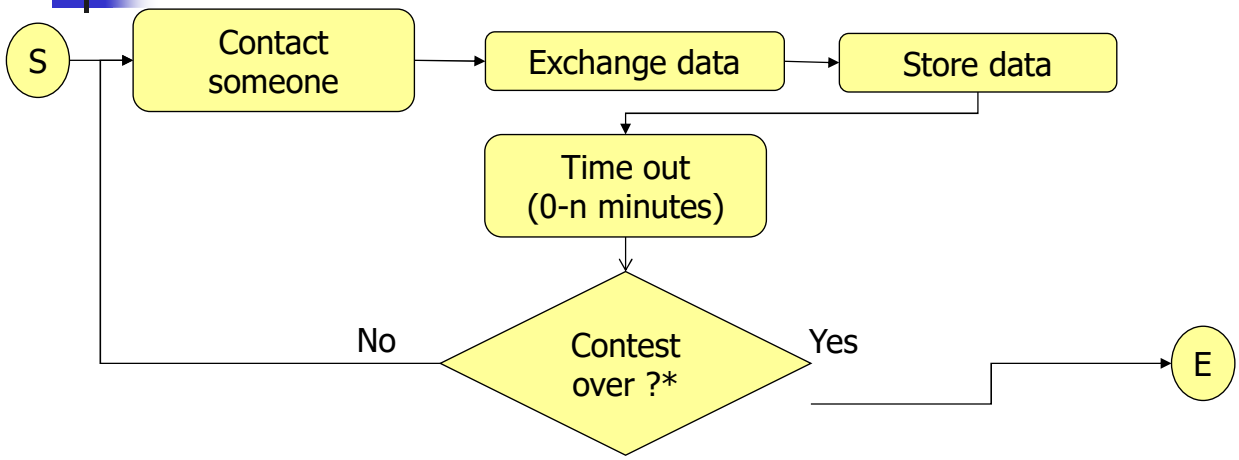
Line Printer and Punched Paper Tape Reader



Simulation software
BASIC
Operating system

No affordable disk drives in 1978

What the operator does (F_QSO)



* Contest over = (24 hours of operation [elapsed time]) or (end time GMT)

Operational Perspective includes …

- OS1 Contacting someone – takes time
  - OS1.1 Calling someone (CQ)
    - Sticking on one frequency and letting people call you
  - OS1.2 Finding someone
    - Tuning bands, finding someone and calling them
- OS2 Exchanging data – takes time
  - sending data
  - receiving data
    - Interference
- OS3 Storing (logging) the data - ~ instantaneous

Example:OS1.1 Calling someone (CQ)

- Sticking on one frequency and letting people call you
- OS1.1.1 Call CQ (F_CQ) and listen (F_RX)
  - Nobody calling
    - Call again or do something else (tune)
  - One station calling
    - If non-duplicate make contact (if dup then B4)
  - More than one station calling
    - Pick one, if non-duplicate make contact
    - If duplicate, pick another one

OS1.2 Finding someone

- Tuning around, hearing someone
  - In a contact
  - Calling CQ
- OS1.2.1 Call station (F_Call) and listen (F_RX)
  - Reply received
  - Reply not received
    - Gone away (for now)
    - Responding to someone else
    - No signals

Column A

1. This sub-state contains the set of activities that explore/scope the problem, leading directly to Phase A.2. The activities performed in this phase produce a definitive statement of the problem-in-context.
2. This sub-state contains the set of activities that conceive the whole solution system (which 'emerges' from/"complements" the problem) and produces the concept of operations (CONOPS) that describes how the solution system will operate in its future environment.
3. **This sub-state contains the set of activities that design the whole solution system, identify the environment, other interacting systems, the subsystems, parts, interactions, functional architecture, physical architecture, etc., etc., - but still all of the whole.**

Functions – sample listing

- Call CQ (F_CQ)
- Receive a call from another station (F_RX)
- Check for duplicate (F_CK)
- Exchange message (F_QSO)
  1. Send message (F_TXM)
  2. Receive message (F_RXM)
- Log contact (F_LOG)
- Tune band (F_QSY)
- Hear another station (F_QRV)
  1. In contact with a third station (QSO)
  2. Calling CQ
  3. Not in contest
- Time outs (F_QRX)
- Etc.

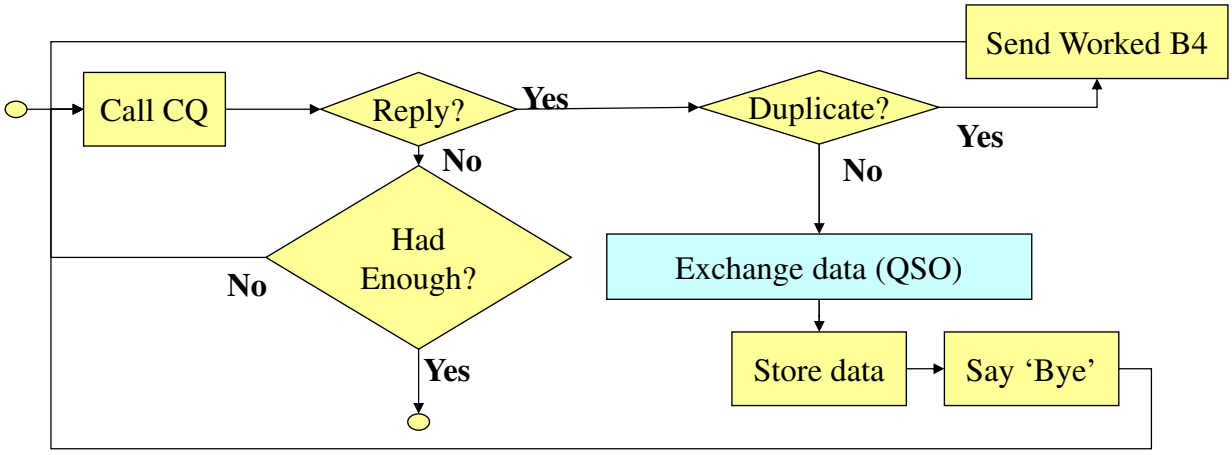
Note. "Q" codes are application domain language

Use facilitates customer involvement

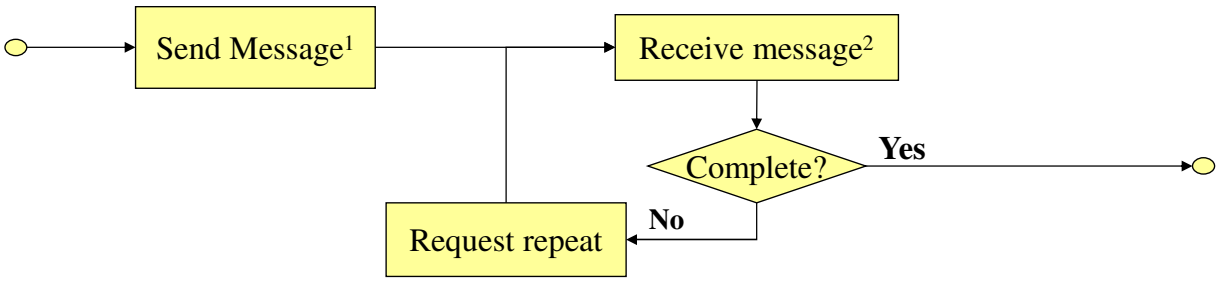
Functional flow diagrams

- Start documenting and expanding ideas
- Intuitive (with application domain knowledge)
- Identify sequences
- Functions can be simple or complex
- Identify dependencies between functions
- May not provide best grouping
- Are a tool to provide a view
- May not be best tool to create relationships between functions

Function flow chart – Call CQ (OS1.1)

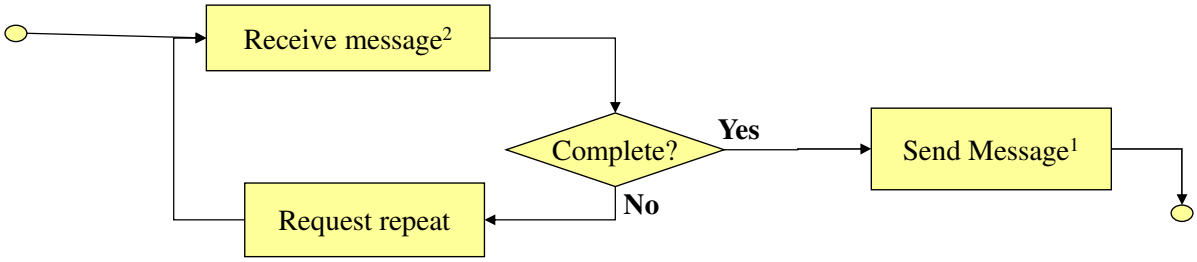


Exchange data (QSO) (1)



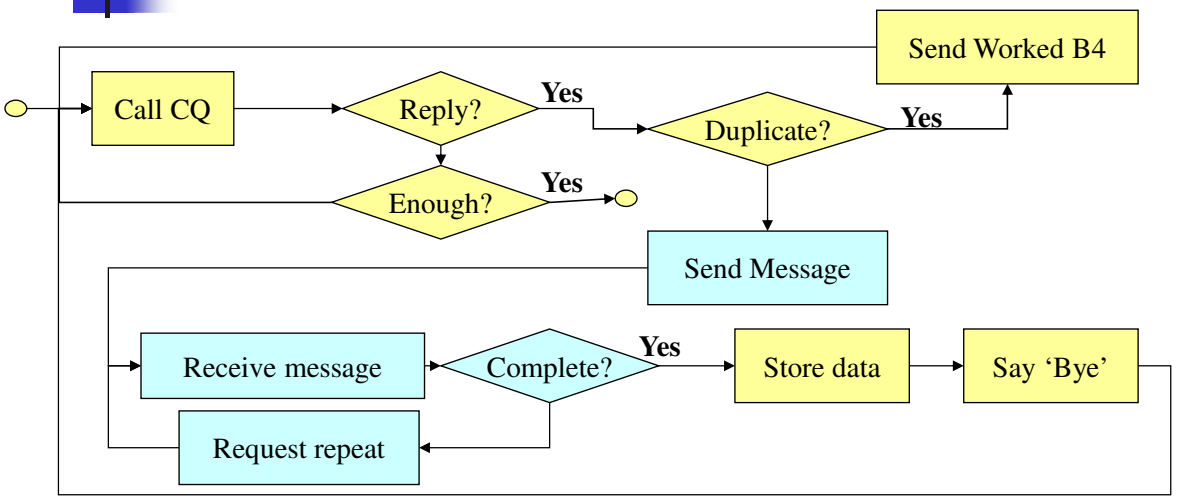
Note 1, sending station may also request you to resend message
Note 2, message may be incomplete due to interference

Exchange data (QSO) (2)



Note 1, sending station may also request you to resend message
Note 2, message may be incomplete due to interference

Alternate flow chart – Call CQ (OS1.1)

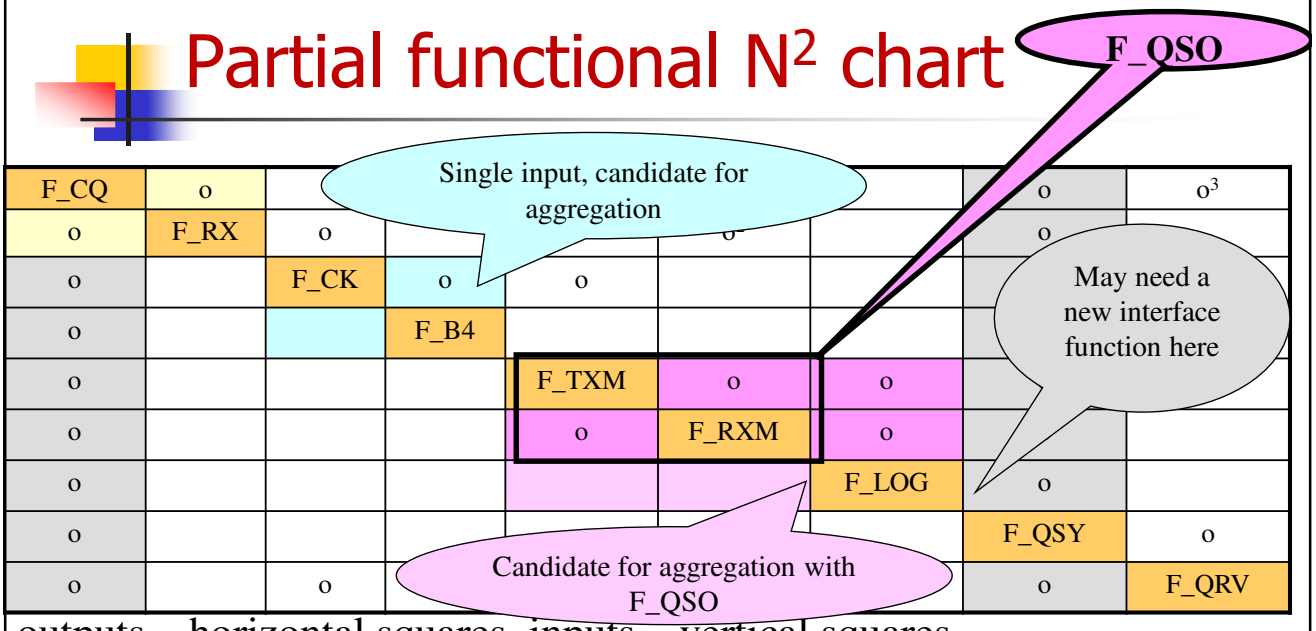


Partial functional N^2 chart

F_CQ	o						o	o^3
o	F_RX	o		o^1	o^2		o	o^4
o		F_CK	o	o			o	
o			F_B4				o	
o				F_TXM	o	o	o	
o				o	F_RXM	o	o	
o						F_LOG	o	
o							F_QSY	o
o		o					o	F_QRV

Outputs – horizontal squares, Inputs – vertical squares

Partial functional N^2 chart



outputs – horizontal squares, inputs – vertical squares

How do you determine system functions?

- Top down?
- Bottom up?
- Middle out?
- **All of the above**
- Tendency to flowchart functions
- N^2 chart is better way for many functions
- Either approach can be best way
  - Checks and balances

Tool usage

- Tools can be used for inputs and outputs (displays)
  - Logic flows
  - Functional flow diagrams
  - IDEF0
  - If-then statements
  - N^2 charts
  - SySML
- Which do you use?
  - For inputs, any as appropriate
  - For displays, use customer's language

Function – determine if contact can be made

- Used in F_Call CQ to determine if a reply will be received
- Use in F_QSY to determine if you hear anyone
- Contains five factors affecting *probability* of radio communication between two locations
  - Needs to be modeled at design time, not now

Development time passes

- All functions implementing CONOPS documented appropriately
- Different ways of functional aggregation into different functional architectures
  - Based on different grouping in N^2 charts
  - Lessons learnt include
    - Incorrect aggregation leads to aggravation

At end of this State – HKM^{2}F Column A

- Statement of problem
- Functional architecture of whole system
- Understanding of factors and issues pertaining to providing the solution
  - Application and implementation domains
- TPMs/risks/constraints
  - e.g. Size of software execution code

Exercise 13-31 Developing the CONOPS

1. Create a high level CONOPS of the software based on the work do you for exercises 13-11 and 13-21. Add additional information if necessary
2. Expand one of the high level blocks in your CONOPS in more detail
3. Prepare a <5 minute presentation containing
    1. The high level CONOPS
    2. The expanded details of one of the high level blocks in the CONOPS
    3. This slide and the version number of the Module
    4. The exercise problem formulated per COPS problem formulation template
    5. A compliance matrix for the exercise
    6. Lessons learned from exercise
4. Save as a PowerPoint file in format Exercise13.31-abcd.pptx
5. Post/email presentation as and where instructed

Any questions ?

1. Best
2. Worst
3. Missing

Email: beyondsystemsthinking@yahoo.com
Subject: <class title> BMWQ Session #