EXAMPLES FOR THE BACKUS-NAUR-FORM (BNF)

**BNF**

- meta-symbols used to describe structure:
  
  + sequence
  
  [ ... ] exclusive alternatives
  
  { ... } iteration (>= 1 repetitions)
  
  ( ... ) option

**Examples**

**BNF**

- rules used to specify substitution/refinement/structuring:
  
  data-item ==> data-structure

  person ==> name + address

  name ==> first_name + last_name

**Data Dictionary**

**Data Item Specification in BNF**

- name: left side of starting rule
- informal description: basic layout
- related data items: decomposition/aggregation
- range of values: terminal symbols
- connections to processes: interface consistency
- purpose: requirements conformance

==> formal data structure specification in BNF

**Conference Registration - BNF**

**Version 1**

registration ==> a_name + company + ((tutorial)) + (author)

a_name ==> first_name + last_name + (middle_init)

company ==> c_name + dept_name + address

address ==> building + street + zip + (country)

tutorial ==> ["tut1" | "tut2" | "tut3" | "tut4"]

author ==> session_no + paper_title + (co_author)

co-author ==> a_name + company

==> handling of tutorials?

**Conference Registration - BNF**

**Version 2**

registration ==> a_name + company + tutorials + (author)

a_name ==> first_name + last_name + (middle_init)

company ==> c_name + dept_name + address

address ==> building + street + zip + (country)

tutorials ==> ("tut1") + ("tut2") + ("tut3") + ("tut4")

author ==> session_no + paper_title + (co_author)

co-author ==> a_name + company

End of Section 4e
SYSTEM and MODULE DESIGN

Elements and Definitions

Design is ...

... the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realisation.

(E.S. Taylor - 1959)

Typical Activities

- handling of constraints and alternatives
- (semi-) formal specification
  - graphical (tabular, etc.) notations
- identification of good modules
- application of heuristics
- completeness and consistency checking
- risk analysis and resolution (e.g. prototypes)

Who does system engineering?

requirements engineer

system engineer / designer

software engineer hardware telecom. organisational expert

programmer

Software System Design

- translates SRS into a
  - software system architecture:
    - system's static structure
      - modules (blackbox)
      - module interfaces
    - dynamic behaviour (execution)
    - data structures
    - user interface design
Detailed Design

- refines the logical system architecture into an implementation oriented design of modules:
  - invocation structure (calls, rpc, message, ...)
  - module interface realisation (parameters, ...)
  - algorithms for each module
  - data structure and user interface solutions
  - if necessary, further decomposition

The Concept of Modular Design

- a module is a separately named component
- a system is decomposed into modules
  - stepwise refinement
  - "top down"
- modules are integrated to satisfy the requirements
  - module aggregation
  - "bottom-up"

>>> aggregation/decomposition abstraction

The Concept of Information Hiding

- D.L. Parnas:
  - a module appears as a blackbox to other modules in the system
  - characterised only by its interface
  - hide internal design decisions/details

>>> support reusability & maintainability

>>> improve functional independence

System Structure - Control Hierarchy

Design Documentation

(variation of R.S. Pressman)

1. Scope
   1.1 System objectives
   1.2 Interfaces (HW, SW, user)
   1.3 Major software functions
   1.4 Externally defined databases
   1.5 Constraints, limitations
2. Reference Documents

Design Documentation

3. SW System Design - Software Architecture
   3.1 System structure
   3.2 Data structure
   3.3 User interface
   3.4 Dynamic behaviour
   3.5 Rules, principles & guidelines
Design Documentation

4. Module Design (for each module)
   4.1 Functional description (informal)
   4.2 Interface specification
      4.2.1 Inputs
      4.2.2 Outputs
   4.3 Processing (semi-formal)
   4.4 Aggregation/decomposition relations
   4.4 Constraints, remarks, etc.

Design Documentation

5. Database and File Structure
6. Requirements Cross References
   7.1 Test guidelines
   7.2 Integration strategy
8. Appendices

End of Section 5

coming up:
structure charts