Overview

- History of SDL
- Benefits of a specification Language
- Characteristics
- SDL Theoretical Model
- Time
- SDL System Example
History

- Object oriented, formal language
  - Defined by International Telecommunications Union (ITU-T)
- Designed to specify complex, event-driven, real-time, and interactive applications
- Initial release in 1976, new ones every 4 years.

Benefits of a Specification Language

- Well defined set of concepts
- Unambiguous, clear, precise, and concise specification
- Basis for analyzing specifications for completeness and correctness
- Basis for determining conformance and consistency of specifications
- Computer support for generating applications
Characteristics

- Standard
- Formal
- Graphical and symbol-based
- Object-oriented
- Portable, scalable, and open
- Highly reusable and efficient

SDL Theoretical Model

- Extended Finite State machines
  - parallel execution
  - independent FSMs
- Structure - 4 hierarchical levels
  - system
  - blocks
  - processes
  - procedures
- Communication
Partitioning

- Hide information
- Follow natural functional subdivisions
- Create modules of intellectually manageable sizes
- Create a correspondence with actual hardware or software
- Reuse already existing specifications.

Structural View
System

- System name
- Channel descriptions
- Signal descriptions
- Data type descriptions
- Block descriptions

Nesting – recursively break down a system
- important for large team development efforts

Blocks and channels define the static structure of a system
- perceived as a black box at its level

Set of processes can be logically grouped
Process

- Nested hierarchical state machine
- Separate memory space (data is local)
- Processes and signal routes define the dynamic structure
  - can be created at system start or created/terminated during runtime
  - more than one instance can exist (unique PID)
- Processes work autonomously and concurrently

Process Constructs

5 basic constructs
- Start
- State
- Input
- Output
- NextState

Start
State
Input
Output
NextState
Process

Process Example

DCL
Counter Integer := 0;

Process - Variable Manipulation
Procedure

- Substate machine
- Can be recursive
- Local to a process or globally available depending on scope
- Remote procedures

Communication
Communication

- Communication mechanism
  - asynchronous signals
  - events occur instantaneously

- Important aspect of real-time systems
  - timer mechanism
  - expired timer generates a signal

- Defined abstractly
  - efficient mapping to time of target system
  - possible to simulate before target availability

Time
Time

SDL System Example

- Surprise…

The Belt Controller!