

VI : Specification & Description Language (SDL)

Referensi

FJ Redmill and A.R. Valdar, SPC DIGITAL TELEPHONE EXCHANGE

Ch. 13




TUJUAN

- Mengetahui arti notasi-notasi standar (SDL) dalam kaitannya dengan fase-fase proses panggilan
- Mampu membaca/mengartikan gambar/diagram
SDL
- Dapat membuat diagram SDL untuk fase panggilan sederhana



Latar Belakang

- Hasil rancangan suatu sistem yang ideal adalah yang sesuai dengan kebutuhan, lengkap dan tak pernah salah.
- Sebagai suatu hasil produk/desain, software aplikasi sentral SPC perlu dideskripsikan spesifikasinya dalam format yang mudah difahami secara umum tentang bagaimana sistem tsb dalam menangani / merespon pengguna (dalam panggilan telepon) dengan berbagai kemungkinannya seperti penyimpangan prosedur dll.
- Untuk mendeskripsikan spesifikasi (software) sentral tsb, ITU-T (dulu masih CCITT), pada awal th 1970 telah membuat standarisasi berupa notasi gambar/diagram yang disebut **Specification & Description Language (SDL)**.

- 
- Dengan (rangkai) gambar notasi tsb dapat diketahui spesifikasi & deskripsi suatu produk (software sentral) antara lain untuk menentukan, apakah sudah sesuai dengan kebutuhan/keinginan.
 - Atau sebaliknya, untuk merancang suatu software aplikasi sentral SPC, perlu dideskripsikan terlebih dahulu apa yang menjadi spesifikasi kebutuhan (requirement). Deskripsi spesifikasi kebutuhan tsb alangkah baiknya jika ditulis dalam format yang mudah dimengerti secara umum, sederhana, standard, dan memudahkan/berorientasi software design.



Steady state & transisi pada proses panggilan

- Dlm proses panggilan, terdapat dua kondisi : **steady state & transisi**.
- Dlm contoh Gambar 13.3 :
 - **idle = steady state**
 - peralihan **dari idle ke off-hook = transisi (= event)**
 - **selama off-hook**, dan blm mendapat respon = **steady state**
 - saat transisi dari idle ke off-hook memunculkan adanya sinyal ke sentral dan menuntut adanya action dari sistem kontrol di sentral. Action tsb diawali dg start program (sub routine) utk menghubungi pelanggan ke sumber nada pilih.
 - Perubahan state dari off-hook ke munculnya nada pilih dari sentral merupakan **transisi**.
 - Selama nada pilih dan A belum mendial digit, merupakan kondisi **steady state**.
 - dst



Simbol-simbol/notasi SDL & definisinya (rek ITU-T)

- Terdapat 8 notasi/simbol dengan nama-nama dan definisinya berdasarkan ITU-T sbb :

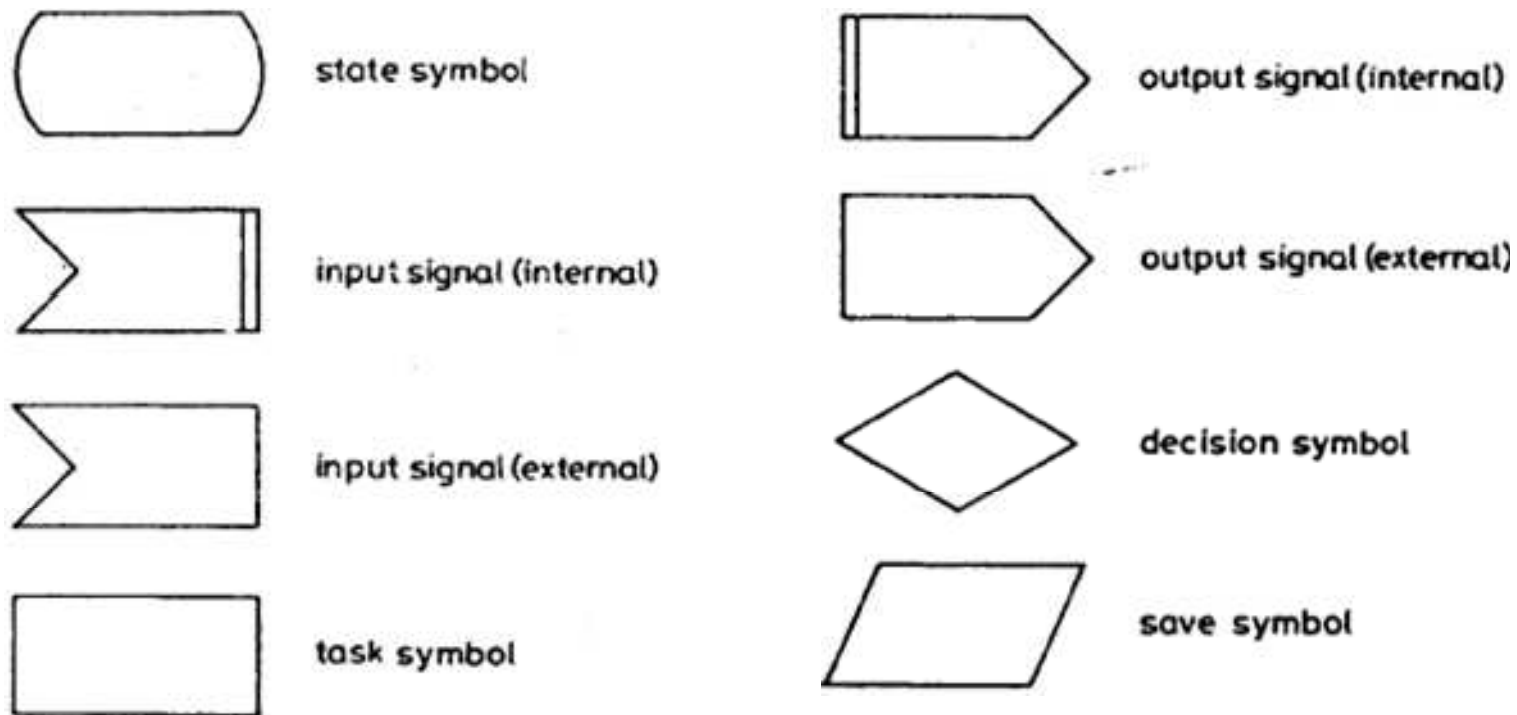



Fig. 13.1 Recommended SDL symbols

- 
- **State** : kondisi dimana suatu proses ditunda (suspended) sementara menunggu input
 - **Input** : sinyal masuk yang dikenali oleh proses
 - **Output** : suatu aksi dalam transisi yang mengeluarkan sinyal *which in turn, acts as an input elsewhere.*
 - **Decision** : Suatu aksi dalam transisi yang merupakan pertanyaan yang harus dijawab dengan satu jawaban/keputusan (dari dua kemungkinan/lebih) untuk melanjutkan transisi.
 - **Task** : *the postponement of recognition of a signal when a process in a state in which recognition of that signal does not occur.*



Definisi/batasan/kriteria :

- **Proses** didefinisikan sebagai pelaksanaan fungsi logika yang memerlukan sejumlah item informasi dimana item ini ada/tersedia pada saat yang berbeda.
- **Signal** didefinisikan sebagai 'flow of data' dalam memperoleh informasi untuk suatu proses.
- **Internal'** dan '**external' signal**, didasarkan kepada model referensi 'functional block' (model dari software modular) sebagai obyek dalam ukuran yang 'managable size' dengan hubungan internal yang relevan yang melaksanakan satu proses atau lebih (lihat Gbr 13.2)

The specification and description language

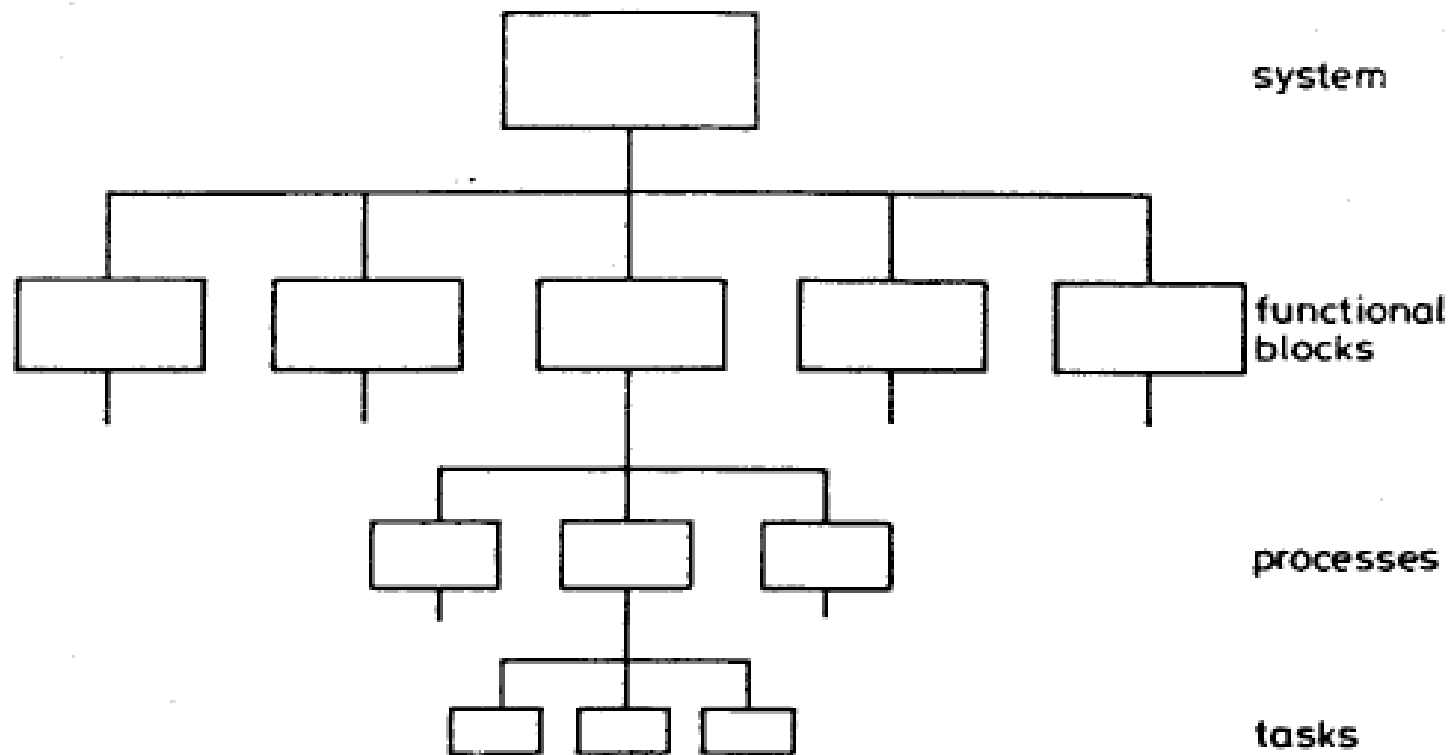




Fig. 13.2 *Functional decomposition of software*



Berdasarkan model Gambar 13.2 serta dari definisi ‘signal’ dan definisi ‘proses’, maka kriteria ‘internal’ dan external’ signal adalah sbb :


- Jika aliran informasi pada suatu proses dalam suatu ‘functional block’ mengalir juga ke dalam proses di ‘**functional block**’ lain, maka aliran informasi (signal) tsb disebut sebagai ‘**external signal**’.
- Namun jika informasi yang mengalir diantara proses-proses yang terjadi masih dalam **satu ‘functional block**’, maka signal tsb disebut sebagai ‘**internal signal**’.

- 
- Gambar 13.3 merupakan contoh penggunaan SDL :
 - high-level SDL (garis besar)
 - dalam panggilan lokal
 - dari sudut pandang (point of view) pelanggan A (bukan dalam sudut pandang operasi sentral, artinya operasi dalam sentral tidak ditulis)
 - Contoh-contoh penggunaan SDL low level (lebih rinci) dalam sudut pandang operasi sentral terdapat pada Chapter 15 (Call Processing)
 - Gambar 14.8 (Chapter 14), merupakan contoh SDL originating call dalam perspektif operasi sentral dengan titik berat pada deskripsi & spesifikasi sentral dalam proses koneksi pelanggan A ke dial-tone sender dan ke MF receiver




▣ **Keuntungan/manfaat SDL**

- SDL merupakan bahasa simbol yang **mudah difahami** secara umum
- Dapat dibuat dalam format high-level (gambaran spesifikasi secara garis besar untuk **konsumsi manajemen**) maupun dalam format low-level/lebih detail (untuk konsumsi **teknisi maupun desainer software**)
- SDL merepresentasikan alur-alur proses panggilan untuk menjamin **tidak terjadinya ambiguous** untuk tiap kondisi steady state dalam proses panggilan
- Untuk tiap sinyal input yang mungkin, dapat ditentukan **alur-alurnya secara akurat**.
- SDL didasarkan atas logika kejadian² (event) dalam proses panggilan yang independen terhadap sistem kontrol (prosesor) maupun bahasa pemrograman komputer.

- 
- Pada SDL dapat dengan mudah dilakukan extensi/modifikasi tanpa mengubah desain utama (misal dalam hal terjadi error karena kekurangan/kesalahan desain software).
 - Sifat mudah difahami dan *unambiguous* SDL membuatnya mudah ditranslasikan ke dalam format dokumen maupun implementasi komputer (software) call processing

▣ **Interpretasi langsung SDL ke dalam software**

- SDL dapat mendeskripsikan alur-alur kondisi steady state proses panggilan beserta transisinya secara pasti dan rinci.
- Karenanya dalam setiap transisi dapat ditentukan secara spesifik *task* apa yang harus dikerjakan oleh sistem kontrol.
- Sehingga interpretasi langsung SDL ke dalam software sentral dapat dilakukan ke dalam struktur data yakni ke dalam *linked list & translation tables*. (Lihat Chapter 12)

- 
- Begitu juga untuk perancangan software call processing, sub-sub rutinnya dapat dirancang dengan translasi langsung dari alur-alur kondisi steady state dan transisi dalam SDL ke dalam bahasa pemrograman tingkat tinggi kemudian ditranslasikan lagi ke dalam bahasa mesin (Assembler) melalui interpreter.
 - Untuk test running juga dapat dilakukan simulasi dengan mengakses *linked list & translation tables*, yang dapat mensimulasikan adanya 'event' yakni dengan mengaktifkan sinyal masukan dan pengecekan data dengan akses pada struktur data, sehingga simulasi running call processing ini disebut juga sebagai 'event driven' dan 'data driven'
 - Adapun record data (struktur data) di sentral tersusun dalam node-node dimana tiap node mempunyai struktur sbb (Gambar 13.4 merupakan struktur data suatu node sinyal input)

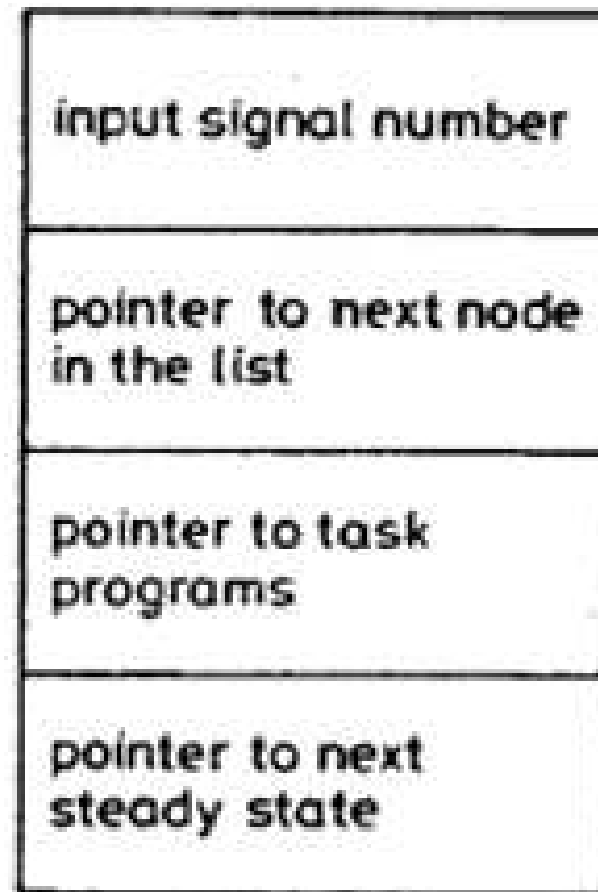


Fig. 13.4 *A node in an input-signal list*

