A1

Event list에 system context diagram에 없는 event가 정의되어 있습니다. Code input을 main input으로 적은 것 같습니다. 9p의 main input interface도 마찬가지

9p의 interface들의 output이 표시되어 있지 않습니다.

디지털 신호라는 단어의 의미를 잘못 사용한 것 같습니다.

11p에서 DFD와 process specification의 이름이 일치하지 않습니다. 이후 페이지들에 이름이 이상한 부분들 언급하지 않겠습니다.

14p MergedBlockStructList 어떻게 연결된 것인지 모르겠습니다.

17p Keyword라는 data store에 control block maker가 값을 정해주고 뒤의 process들이 Keyword의 값을 받는 것으로 되어있는데, 20p의 FSM을 보면 control block maker가 Keyword의 값을 보고 각각의 process들을 trigger 하는 것으로 되어있습니다. FSM의 의도대로라면 ArrangedCode를 보고 Keyword값을 정해주는 process를 따로 정해주고 control block maker가 사용하는 방식이 되어야 할 것 같습니다. 한마디로 이렇게 하면 안 돌아갑니다. -_-;

Process specification과 DFD의 배치 신경 좀 써주세요. 보기 힘듭니다.

21p level이 process의 번호를 보면 3 level로 보입니다. 25p는 2 level인 것 같네요.

21p final report에서 save command라는 명령을 주는 부분이 이상합니다. Total CFG가 saver라는 실제 result.txt파일의 내용을 입력하는 process에 전달되는 부분이 안보입니다.

Total DFD에는 generator controller가 ReadLineCount, RouteError, total CFG을 입력으로 받는데 DFD에서는 ArrangedCode를 받습니다.

DFD를 그릴 때 상위와 하위 level을 일치시키는 부분에 주의해주세요.

A6

8p 같은 이름을 가지고 있는 process가 있으면 안됩니다. Process 1과 1.3의 이름이 같습니다. 이후에도 이름 같은 process가 몇 개 보입니다.

DFD Message Interface에서 나오는 message display가 어디로 이어지는지 모르겠습니다. 일단 DFD1에서 validity control에서 나오는 data flow는 isSuccess뿐이며, 혹시 terminator에 연결되는 data flow라면 system context diagram에 표현되어 있어야 합니다.

Process specification에 설명되어 있지 않는 process들이 있습니다. 따라서 전체적으로 해석이 안 됩니다.

Validity data & code data라는 data store 또한 현재 DFD1대로라면 Validity 내부에서만 사용되는 것으로 표현되어 있기 때문에 convert control에서 사용할 수 없습니다.

9p DFD level 2의 validity control에서 나오는 success sign command가 level3에서 어떻게 나가는지 표현되어 있지 않습니다.

Level 3에 level 2의 process인 parsing이 그대로 표현되어 있습니다.

10p 하위 level을 표현할 때는 하나의 슬라이드에 하나의 상위 process의 내부만 표현해주세요. DFD가 전체적으로 이해하기 힘듭니다.

convert controller 2.2.1.0이라는 process가 있는데 level 3의 convert controller 2.2.1과 같은 것 같습니다. 하나의 process내부만 표현한 것이 아니라 여러 개의 내부를 동시에 그리다 보니 이렇게 표현한 것으로 추측되는데, 하나의 process입장에서 봤을 때 입력을 표현하는 것은 data/control flow만 그리면 되지 입력을 주는 process까지 표현할 필요가 없습니다. System context diagram과 DFD level 0의 관계를 생각해보세요.

11, 12p의 level이 잘 못 적혀있습니다.

Condition을 적을 때에는 !을 사용한다고 무조건 false값을 의미하는 것이 아닙니다. isValidFile을 true/false 모두 가질 수 있도록 정의해놨기 때문에 isValidFile = false 이런 식으로 적어주셔야 합니다. RVC 예제에서는 완벽하지 않은 자료를 업로드 해놓은 상태라 R, L등의 표현이 쓰였습니다만, 원본은 R,L등이 '~~방향에 장애물이 있음' 이라고 정의되어 있기 때문에 !을 붙이면 '~~에 장애물이 없음'이 되기에 가능한 표현이었습니다.

고민한 흔적은 많이 보이나 아직 SASD의 개념에 대한 이해가 조금 부족해 보입니다.

A7

7p DFD에 data flow에 어떤 것이 전달되는지 표현되지 않은 것이 있습니다. 이후 페이지에도 많네요 -_-;

Null pointer Error, command Error, Non standard error는 어떻게 처리되는 건지 모르겠습니다. 이후에 쓰는 곳도 없는 것 같고, 만약 terminator로 전달되는 출력이라면 system context diagram에 표현되어 있든지 event list에 표현되어 있어야 할 것 같습니다.

Divide and conquer의 개념을 생각해보세요.

9p cc cv의 느낌이 나는 것은 제 착각일까요? ^^;

17p 2.1이 나오지 않은 상태에서 2.1.0이 나옵니다. 어떻게 연결되는지 모르겠습니다.

enable/ disable이 제가 생각하는 것 대로라면 data flow가 아니라 control flow가 되어야 합니다. 그리고 하나의 flow가 아니라 각각의 control flow로 연결되어야 합니다.. 하나의 data/control flow당 하나의 화살표로 표현해주세요. 다른 페이지에도 묶어놓은 곳들이 몇 개 보입니다.

Source parser가 하는 일은 control process인데 data process로 표현되어 있고 FSM도 없네요.

Total DFD 잘 안보입니다.

내용 파악이 잘 되지 않아 내용에 대한 comment는 하지 않겠습니다만 많은 수정이 필요해 보입니다.

A8

system context diagram과 event list의 이름이 다릅니다.

6p CFG generator에 numbering 1이라고 되어있어야 할 것 같네요.

7p terminator의 개념을 모르시는 것 같습니다. DFD에서 네모로 표현되는 것은 terminator인데 소프트웨어가 담당하는 부분이 아니라 외부환경과의 입출력을 담당하는 부분입니다.

Terminator에 numbering하지 않습니다.

왼쪽 네모 두 개는 data process인데 terminator로 표현한 것 같고 오른쪽 네모 두 개는 terminator인데 numbering을 해놓았네요.

DFD의 level이 올라간다는 것은 대상 data process의 내부를 보겠다는 말입니다. Level0의 CFG generator 내부를 7p에 표현해 놓은 것인데 그럼 CFG, Result같은 terminator가 표현되어 있으면 안됩니다. 이미 상위레벨에서 CFG, Result가 표현되어 있으므로 지금 그려놓은 DFD대로라면 같은 이름을 가지는 terminator가 두 개씩 있는 것이 됩니다.

같은 이름을 가지는 process가 있으면 안됩니다. CFG generator가 여러 번 사용되어있네요.

한마디로 이상합니다.

8p 1.3.1은 control process를 표현한 것 같습니다. Control process는 점선으로 표현되어야 합니다.

Level 1에서 1.3에서 CFG list, result list가 나왔으니 level 2에서 같은 것이 나와야겠죠?

9p cd, C, A등의 정의를 명확하게 해주세요. 명확하지 않아서 해석이 잘 안됩니다.

대략적으로 보면 에러 없으면 분석 시작하고 Analyze C code에서 block, edge 만든다는 것 같긴 한데 제가 보기엔 CFG를 그리기에는 너무 심플하게 되어 있고 앞뒤도 잘 맞지 않는 것 같습니다.

Analyze C code state만으로 control process하나는 나올 정도로 더 분석되어야 디자인을 할 수 있을 것 같습니다.

18p의 제목은 처음 보는 것이네요. Total DFD를 이야기 하려고 한 것인가요?

Trigger 연결 안되어 있는 process가 있네요.

아직 많이 손봐야겠습니다.

A9

Terminator의 개념에 대해서는 A8의 comment 참조해주세요.

4p System context diagram은 외부환경과 대상 시스템의 경계를 결정하는 모델로서, 여러 개의 terminator와 event, 하나의 소프트웨어(중앙의 동그라미)로 이루어집니다.

따라서 6p처럼 terminator에서 외부로 data flow가 나가는 것은 불가능하며 terminator 바깥의 환경은 고려하지 않는다는 것을 명확히 해주는 것이 system context diagram입니다.

같은 이름이 쓰인 다른 process가 있습니다.

14p tick을 사용하셨는데 tick을 사용한 transition과 안 사용한 transition에 차이점을 둔 것인지요? 발표를 들은 게 아니라 단언하지는 못하겠지만 원래 방식과 조금 다른 방식으로 사용한 것 같습니다.

Check command에서 check analyze data로 넘어올 때, 그리고 check command에서 stop으로 넘어갈 때 잘못된 사용을 하셨습니다. Control process는 주로 다른 process에 명령을 내릴 때 사용되며 action부분('/ '뒷 부분)에서는 control process에서 output으로 나가는 부분에 명령을 주는 용도로 사용됩니다. 지금 두 transition에서 사용하신 방법은 중 check analyze data는 그 때만 analyze data라는 data store에 있는 값을 확인하겠다는 의도로 보이는데 이런 방법으로 사용되는 것이 아닙니다. Command value와 analyze data는 input으로 들어오는 것으로 그의 값을 control process에서 항상 고려하고 있으며 값을 조건으로 삼아 action이 결정되는 방식입니다. 그래서

RVC 예제 설명할 때 모든 경우에 대해 조건을 명확하게 정해달라는 이야기를 하기도 했습니다. 다른 transition에서 stop을 action부분에 적어놓은 부분이 있는데 이렇게 적어놓는다고 FSM이 종료되는 것이 아닙니다. Stop이라는 state로 간다고 해서 종료되는 것도 아니고 그냥 이름이 stop일 뿐입니다. Action에 stop을 써놓는 것의 여부와 관계없이 단지 stop이라는 state가 end에 연결되어 있으므로 자연스럽게 종료된다고 보시면 되겠습니다.

B9

8p 같은 이름을 가지는 것은 같은 의미이기 때문에 event list에 있는 것을 굳이 다시 적어줄 필요는 없습니다. ^^; notation들도 마찬가지.

DFD level0을 제외한 data dictionary가 없습니다. 따라서 전체적으로 해석하는데 어려움이 있습니다.

9p command input이 두 개 있네요. 오타로 생각됩니다.

15p make block과 make edge에서 나오는 data flow가 무엇인지 적혀있지 않습니다.

15p print complete가 없는데 20p의 print start state에서 code analyzer state로 넘어갈 때 print complete가 사용되고 있습니다. 정의된 페이지가 있나 찾아 보았습니다만, 없는 것 같네요. Code analyzer process를 trigger하려는 것인가요?

20p print help state에서 initial state인 command analyzer로 연결되어 있습니다. 화살표가 반대로 연결되어 있는 것인가요? 반대로 연결된 게 아니라면 FSM은 initial state에서 시작하는 것이기 때문에 이상한 FSM이 됩니다.

End state가 있는 FSM으로 보입니다. End state들 표현해주세요.

FSM의 code analyzer에서 실질적으로 CFG를 그리는 것 같은데 현재 상태만으로는 CFG를 그리는데 부족해 보입니다. Code data를 받아 바로 block edge를 그리는 방식인데, 앞에 block과 edge를 구별하는 process를 추가하든지 더 자세히 state를 나눠야 될 것 같습니다. 현재는 Code analyzer state만으로 하나의 control process가 나올 수 있는 수준입니다.

발표자료를 급하게 만드신 것 같습니다.