

RENDSZERMODELLEZÉS

JELMAGYARÁZAT : DEFINÍCIÓ , PÉLDA ,

I.

Modell: egy valós vagy hipotetikus világ (a „rendszer”) egy részének egyszerűsített képe, amely a rendszert helyettesíti bizonyos megfontolásokban. Egy modell elkészítésének mindig egy kérdés megválaszolása a célja.

{– A modell az eredeti rendszernél kisebb, hiszen az aktuális problémához nem (vagy lazábban) kapcsolódó, elhanyagolható információk nem jelennek meg benne.

– A modell az eredeti rendszernél áttekinthetőbb, hiszen csak az adott probléma szempontjából érdekes, releváns információkat és kapcsolatokat kell vizsgálni. }

Diagram: a modell egy nézete, amely a modell bizonyos aspektusait grafikusán ábrázolja.

{– A modell nem a valóság: az általunk definiált modellen bizonyos állítások igazak lehetnek, amik a valóságban nem állják meg a helyüket.

– A modell nem a diagram: a diagram csak egy ábrázolás módja a modellnek, amivel olvashatóvá tesszük. }

Egy modellezési nyelv négy részből áll:

– Absztrakt szintaxis (más néven metamodell): meghatározza, hogy a nyelvnek milyen típusú elemei vannak, az elemek milyen kapcsolatban állhatnak egymással, és a típusoknak mi a viszonya egymáshoz. Ezt a reprezentációt használják gépek a modell belső tárolására.

– Konkrét szintaxis: az absztrakt szintaxis által definiált elemtípusokhoz és kapcsolatokhoz szöveges vagy grafikus jelölésrendszert definiál. Ezt a reprezentációt használják az emberek a modell olvasására és szerkesztésére.

– Jólformáltsági kényszer: Megadja, hogy egy érvényes modellnek milyen egyéb követelményeknek kell megfelelnie.

– Szemantika: az absztrakt szintaxis által megadott nyelvi elemek jelentését definiáló szabályrendszer. (leírja, hogyan működik a modell) Egy modellezési nyelvhez több konkrét szintaxis is adható.

Zárt világ feltételezés: Minden állítás, amiről nem ismert, hogy igaz, hamis.

Nyílt világ feltételezés: Egy állítás annak ellenére is lehet igaz, hogy ez a tény nem ismert.

Rendszer és környezete: a környezet (vagy kontextus) a rendszerre ható tényezők összessége. Modellezéskor a modellezett rendszernek mindig egyértelműen definiálni kell a határait. A határon belül eső dolgokat a rendszer részének tekintjük, az azon kívül esők adják a környezetet. Szokás még a környezet elemeit két csoportba sorolni: releváns környezeti elemek azok a dolgok, amik a rendszerrel közvetve vagy közvetett módon kapcsolatban állnak, míg irreleváns környezeti elem, ami a rendszerrel nincs kapcsolatban

Finomítás: a modell olyan részletezése (pontosítása), hogy a környezet szempontjából nézve a finomított modell (valamilyen tekintetben) helyettesíteni tudja az eredeti modellt.

Absztrakció: a finomítás inverz művelete, vagyis a modell részletezettségének csökkentése, a modellezett ismeretek egyszerűsítése.

A strukturális modell a rendszer felépítésére (struktúrájára) vonatkozó tudás. A strukturális modell a rendszer alkotórészeire, ezek tulajdonságaira és egymással való viszonyaira vonatkozó statikus (tehát változást nem leíró) tudást reprezentál.

A jellemző egy, a modell által megadott parciális függvény, amelyet a modellelemeken értelmezünk

A típus egy kitüntetett jellemző, amely meghatározza, hogy milyen más jellemzők lehetnek értelmezettek az adott modellelemre, illetve milyen más modellelemekkel lehet kapcsolatban. A többi jellemzőt tulajdonságnak hívjuk.

Egy adott típus példányainak nevezzük azon modellelemeket, amelyek típusa t.

A típusgráf egy olyan gráf, amelyben minden csomóponttípushoz egy típuscsomópont, minden éltípushoz egy típusél tartozik.

Egy adott típusgráf példánygráfja alatt olyan gráfmodellt értünk, amelynek elemei a típusgráf csomópont- és éltípusainak példányai, valamint minden él forrása és célja rendre az éltípus forrásának és céljának példánya.

A dekompozíció (faktoring) egy rendszer kisebb komponensekre bontása, amelyek könnyebben érthetők, fejlesztethetők és karbantarthatók.

Egy dekompozíció helyes, ha a dekompozícióval kapott rendszer minden elemének megfeleltethető az eredeti rendszer valamelyik eleme, és az eredeti rendszer minden eleméhez hozzárendelhető a dekompozícióval kapott rendszer egy vagy több eleme.

A szűrés művelet során a modell elemein kiértékelünk egy feltételt és azokat tartjuk meg, amelyek megfelelnek a feltételnek. (tulajdonságmodell) pl: Szeretnénk megtudni, hogy mely telephely alkalmas legalább 100 jármű befogására.

Vetítés során a modell egyes jellemzőit kiválasztjuk és a többit elhagyjuk a táblázatból. Megjegyzés. Érvényes vetítés művelet az is, ha a tulajdonságmodell összes jellemzőjét megtartjuk. pl: Olyan kimutatásra van szükségünk, ami csak az egyes telephelyek helyszínét, funkcióját és kapacitását tartalmazza. (letakaras)

Top-down modellezés során a rendszert felülről lefelé (összetett rendszerből az alkotóelemei felé haladva) építjük. A modellezés alaplépése a dekompozíció.

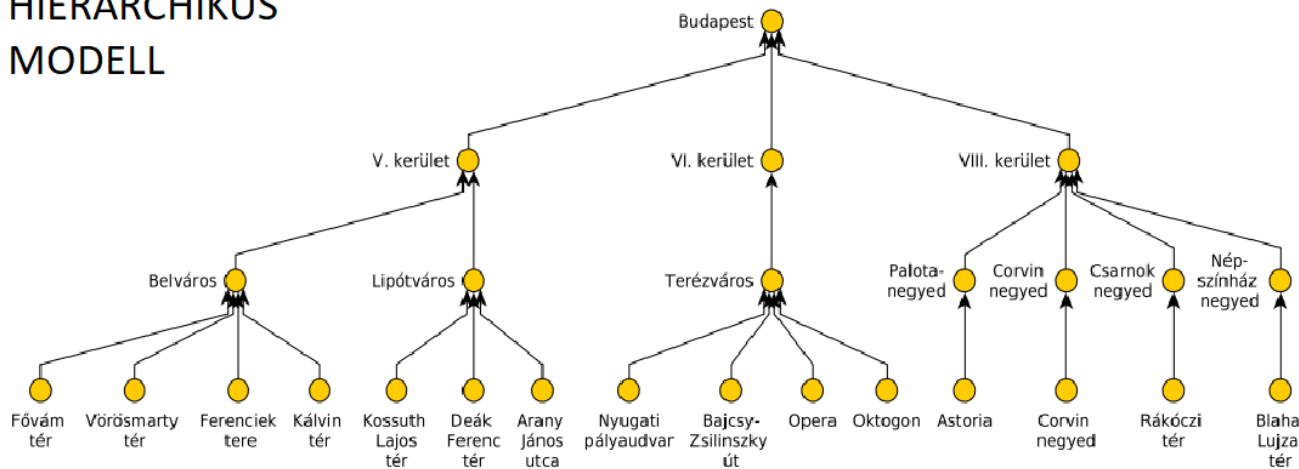
\oplus Részrendszer tervezésekor a szerepe már ismert \ominus „Féldőben” még nincsenek működő (teljes részletességgel elkészített) részek \ominus Részek problémái, igényei később derülnek ki

Bottom-up modellezés során a rendszert alulról felfelé (elszigetelt alkotóelemekből az összetett rendszer felé haladva) építjük. A modellezés alaplépése a kompozíció: az egész rendszer összeszerkesztése külön modellezett vagy fejlesztett részrendszerekből.

\oplus A rendszer részei önmagukban kipróbálhatók, tesztelhetők \oplus Részleges készütségnél könnyebben előállítható a rendszer prototípusa \ominus Nem látszik előre a rész szerepe az egészben

Példa. Budapestnek 23 kerülete van, amelyek további városrészekből állnak. Melyik városrészben van az Opera metrómegállója? Melyik városrészben van a legtöbb metrómegálló? Ezekhez hasonló kérdésekre úgy tudunk hatékonyan válaszolni, ha készítünk egy hierarchikus modellt a problémáról.

HIERARCHIKUS MODELL



4. ábra. Budapest kerületei, városrészei és metrómegállói (részleges modell)

A hierarchikus gráf összefüggő, nem tartalmaz kört, és fa.

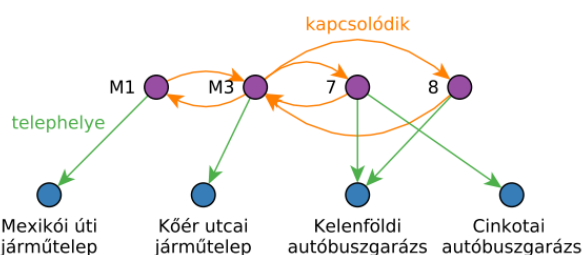
Tulajdonságmodell:

azonosító	helyszín	funkció	kapacitás	vágányhossz	max. üzemanyag
T1	Mexikói út	metró járműtelep	24	8 500 m	
T2	Kőér utcai	metró járműtelep	60	16 512 m	
T3	Cinkota	autóbuszgarázs	265		250 000 liter
T4	Kelenföld	autóbuszgarázs	322		200 000 liter

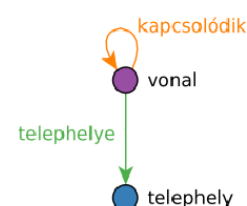
1. táblázat. A BKK telephelyei (részleges modell)

kapacitás(T2) = 60, kapacitás(T3) = 265.

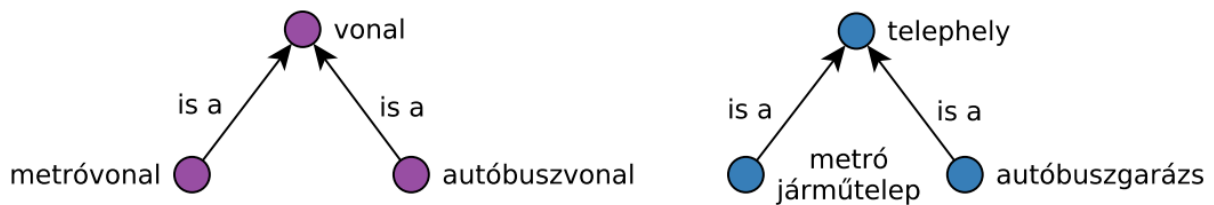
Pédánygráf - Típusgráf



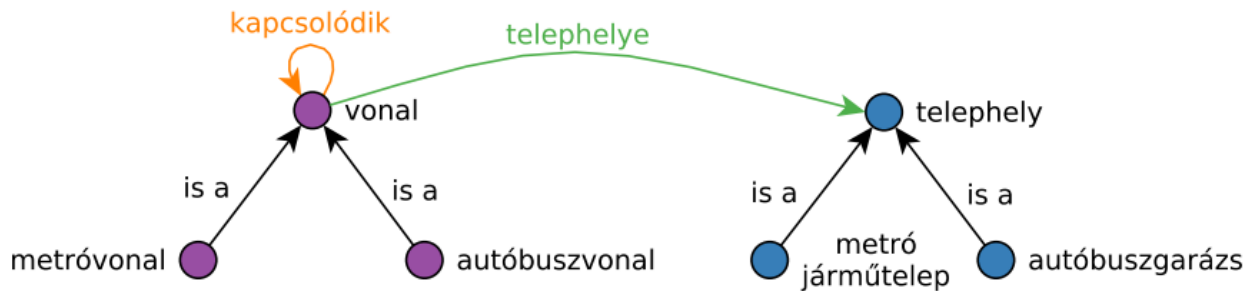
7. ábra. Közlekedési vonalak és telephelyek



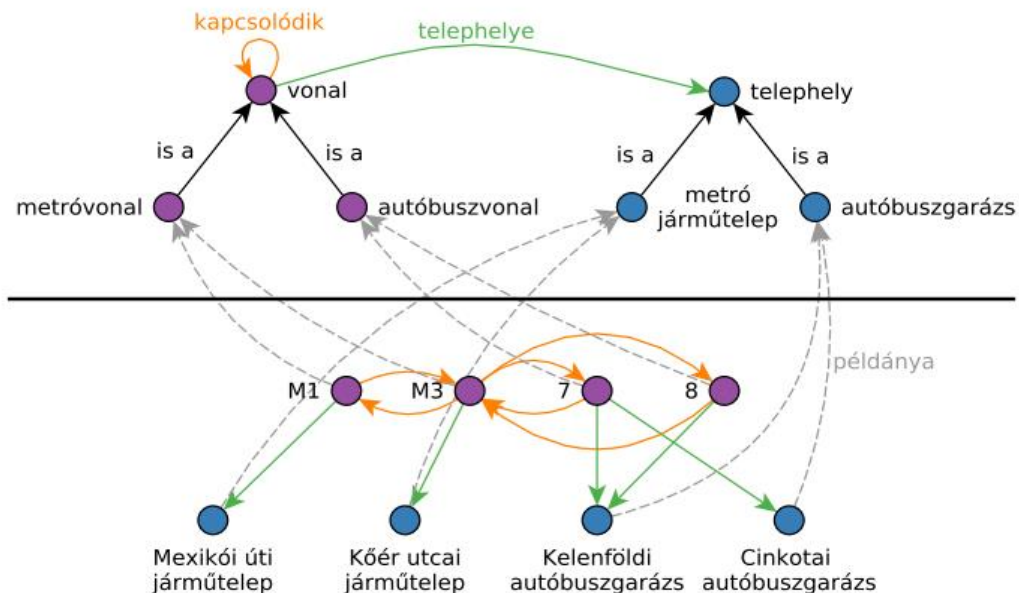
8. ábra. Közlekedési vonalak és telephelyek típusgráfja



10. ábra. Közlekedési vonalak és telephelyek típushierarchiája



11. ábra. Közlekedési vonalak és telephelyek (részleges) metamodelje



12. ábra. Közlekedési vonalak és telephelyek példánymodellje és (részleges) metamodelje egy gráfon ábrázolva

III.

Az állapottérnek az alábbi két kritériumnak kell megfelelnie:

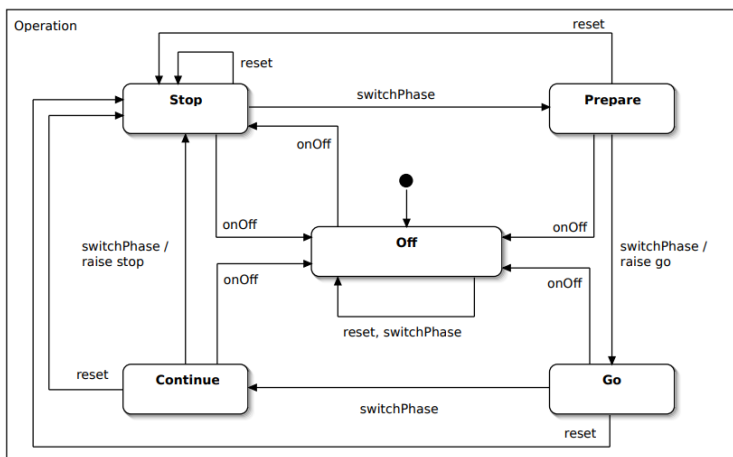
Teljesség. Minden időpontban az állapottér legalább egy eleme jellemzi a rendszert.

Kizárólagosság. Minden időpontban az állapottér legfeljebb egy eleme jellemzi a rendszert.

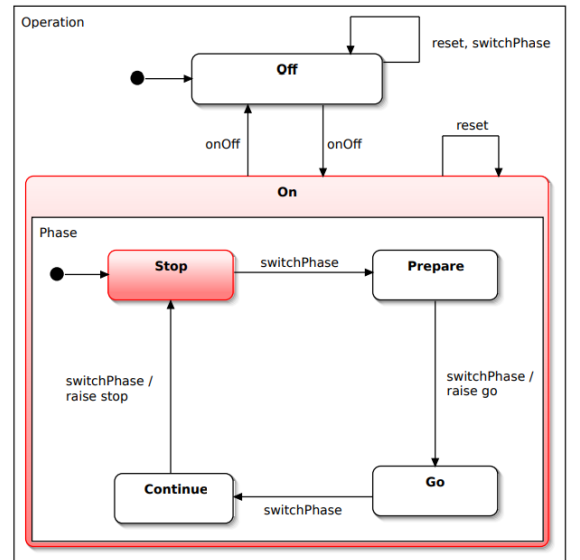
Determinisztikus. Az állapotgépnek legfeljebb egy kezdőállapota van, valamint bármely állapotban, bármely bemeneti esemény bekövetkeztekor legfeljebb egy tranzíció tüzelhet.

Teljesen specifikált. Az állapotgépnek legalább egy kezdőállapota van, valamint bármely állapotban, bármely bemeneti esemény bekövetkeztekor legalább egy tranzíció tüzelhet. Megjegyzés. Ennek egyik következménye, hogy a rendszer holtpontmentes, azaz nem tartalmaz olyan állapotot, amelyből nem vezet ki tranzíció.

Összetett állapot: amely a négy állapot közös tulajdonságait és viselkedését általánosítja. 2. ábra:



3. ábra. Jelzőlámpa állapotgépe reset eseménnyel ☠



Példa. A jelzőlámpa érvényes állapotterei:

- { Off, On }
- { Off, Stop, Prepare, Go, Continue }

Ezen felül az állapotkonfigurációk halmaza is tekinthető állapottérnek:

- { { Off }, { On, Stop }, { On, Prepare }, { On, Go }, { On, Continue } }

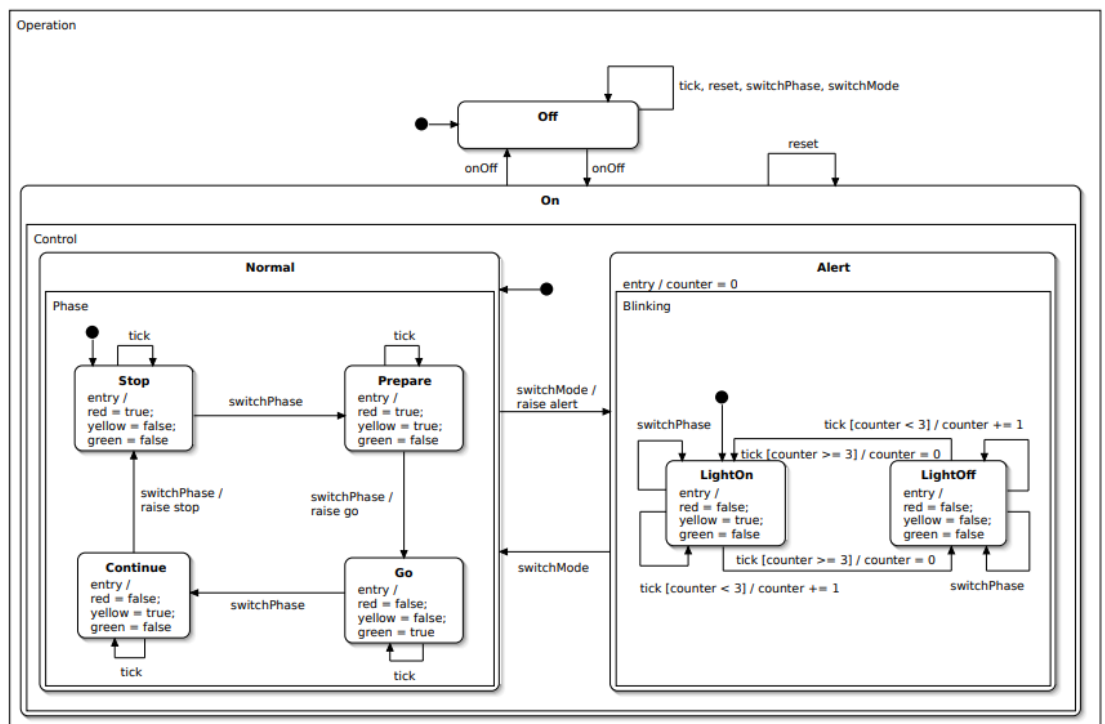
Ugyanakkor az { Off, On, Stop, Prepare, Go, Continue } nem jó állapottér, hiszen ebben az esetben például az { On, Prepare } részhalmaz sérti a kizárólagosságot. Általános szabály, hogy egy állapottér vagy az összetett állapotot, vagy annak összes részállapotát tartalmazza, de nem mindkét változatot.

Őrfeltétel : A tranzakcióra felírt feltétel.

Interfaceváltozók

szemléltetése :

red-green-yellow:



Állapotgépek szinkronizációján (más néven randevú, esetenként handshake, vagy programnyelvek esetén barrier) azt értjük, hogy két kooperáló állapotgépben bizonyos állapotátmenetek csak egyszerre történhetnek meg. A szinkronizálódó tranzíciókat szinkronizációs címkével jelöljük meg. Jelölése az állapotátmeneten: trigger [őrfeltétel] / akció.

Ortogonalis dekompozíció : Párhuzamos állapotgép működés, ortogonalis régiók a párhuzamos szálak

IV.

A folyamat tevékenységek összessége, melyek adott rendben történő végrehajtása valamilyen célra vezet.

Az elemi tevékenység olyan időbeli kiterjedéssel rendelkező tevékenység, amelynek a megkezdésén és befejezésén túl további részleteit nem modellezzük. minden elemi tevékenység önmagában egy háromelemű állapotteret határoz meg: {még nem kezdődött el, folyamatban van, már befejeződött}

[Szekvencia]: A folyamatmodellek igazi erőssége, hogy a tevékenységekből folyamatot építenek fel, amely azt fejezi ki, hogy az egyes tevékenységek egymáshoz viszonyítva mikor hajthatók végre. A legegyszerűbb ilyen konstrukció a szekvencia, ahol a tevékenységeket úgynevezett vezérlési élek (vezérlésifolyam-élek) kötik össze.

A szekvencia tevékenységek szigorú végrehajtási sorrendjét definiálja.

A vezérlési elem (vezérlési csomópont / vezérlésifolyam-csomópont) olyan csomópont a folyamatban, mely a folyamatmodell tevékenységei közül választ ki egyet vagy többet végrehajtásra.

A döntési csomópont olyan csomópont a folyamatban, amely a belé érkező egyetlen vezérlési él hatására a belőle kiinduló ágak (vezérlési élek) közül pontosan egyet választ ki végrehajtásra, az esetleges őrfeltételekkel összhangban.

A merge (besorolódási) csomópont olyan csomópont a folyamatban, amely a belé érkező ágak közül akármelyik végrehajtásakor kiválasztja a belőle kiinduló egyetlen vezérlési élet további végrehajtásra.

A ciklus olyan folyamatmodell (részlet), amelyben egy elágazás valamelyik ágán az elágazást megelőző merge csomópontba jutunk vissza

Két bekövetkező tevékenység vagy esemény konkurens, ha a bekövetkezési sorrendjükre nézve nincs megkötés.

A fork csomópont olyan csomópont a folyamatban, amely a belé érkező egyetlen vezérlési él hatására a belőle kiinduló összes párhuzamos folyamat (vezérlési élet) kiválasztja végrehajtásra.

A join (találkozási / szinkronizáló) csomópont olyan csomópont a folyamatban, amely a belé érkező összes párhuzamos folyamat végrehajtása után kiválasztja a belőle kiinduló egyetlen vezérlési élet további végrehajtásra.

Definíció. Minden folyamat egy start (flow begin) csomópont vezérlési elemmel indul, és egy cél (flow end) csomópont elemmel fejeződik be. Az start (flow begin) csomópont a folyamat elindulását jelentő elem, melynek pontosan egy kimenete van. A cél (flow end) csomópont a folyamat befejezését jelentő elem, melynek pontosan egy bemenete van.

Egy folyamatmodellhez tartozó folyamatpéldány olyan diszkrét eseménysor, amelyet a következő jellegű események alkotják, a folyamatmodell által megszabott időrendben:

- a folyamat kezdete,
- a folyamatot alkotó egyik tevékenység kezdete,
- a folyamatot alkotó egyik tevékenység vége,
- a folyamat vége

Egy folyamatnak több folyamatpéldánya lehet; sőt, több olyan példánya is, amely ugyanolyan eseményeket tartalmaz ugyanolyan sorrendben.

(control flow) : Egy program által meghatározott folyamatmodellt, amely az elvégzendő lépéseket, ill. a végrehajtásukra előírt sorrendet írja le, a program vezérlési folyamának (control flow) nevezzük.

imperatív programnak : Azon programokat, amelyek vezérlési folyamatot határoznak meg, imperatív programnak nevezzük.

A vezérlési folyamhoz tartozó ciklomatikus komplexitás: $M = E - N + 2$, ahol E a vezérlési élek, N a vezérlési csomópontok és tevékenységek száma.

A következő (egy belépési és egy kilépési pontú) részfolyamatokat tekintjük jólstrukturált blokknak (más néven jólstrukturált részfolyamatnak):

- egyetlen elemi tevékenység önmagában;
- egyetlen folyamathivatkozás/hívás (máshol definiált folyamatmodell újrafelhasználása);
- üres vezérlési élszakasz;
- „soros kapcsolás”: a P1, P2, ..., Pn jólstrukturált blokkok szekvenciája (egyszerű vezérlési éllel egymás után kötve őket);
- „fork-join kapcsolás”: a P1, P2, ..., Pn jólstrukturált blokkok egy n ágú fork és egy n ágú join közé zárva;
- „decision-merge kapcsolás”: a P1, P2, ..., Pn jólstrukturált blokkok egy n ágú decision és egy n ágú merge közé zárva;
- „Ciklus”: egy kétágú merge csomóponttal kezdődik, amely után egy jólstrukturált P1 blokk következik, majd egy kétágú decision, melynek egyik ága a részfolyamat vége, a másik a P2 jólstrukturált blokkokon keresztül az előbbi merge-be tér vissza.

Egy teljes folyamatmodell jólstrukturált, ha egyetlen belépési pontja (Flow begin) és kilépési pontja (Flow end) egy jólstrukturált blokkot zár közre.