

# Softwarový nástroj CESim pro grafický návrh, simulaci a analýzu C-E Petriho sítí

Ing. Petr NOVOSAD

*FIT VUT Brno,*  
Božetěchova 2, 612 00 Brno  
novosad@fit.vutbr.cz

Vedoucí práce: Prof. RNDr. Milan Češka, CSc.

**Abstrakt:** Prezentovaný článek představuje nový programový nástroj CESim pro práci s C-E Petriho sítěmi. Tyto sítě jsou podtřídou obecných Petriho sítí a doposud pro ně neexistoval vhodný programový nástroj. Vytvořený program CESim obsahuje grafický editor pro návrh sítí, automatický i manuální simulátor a poskytuje prostředky pro analýzu sítí s využitím případového grafu a výskytové sítě. Pro usnadnění práce s případovým grafem byl navíc v rámci projektu vyvinut genetický algoritmus pro jeho automatické zobrazení.

**Klíčová slova:** CESim, C-E Petriho síť, systém, grafická editace, simulace, analýza, případový graf, výskytová síť, genetický algoritmus

## 1 Úvod

Rozšíření Petriho sítí do mnoha vědních a technických oblastí vede k vývoji programových nástrojů pro práci s nimi. Tyto nástroje výrazně usnadňují modelování systémů pomocí Petriho sítí a přispívají tak k jejich dalšímu rozšíření.

Cílem mého projektu bylo vytvořit program CESim pro podporu výuky C-E Petriho sítí. Úkol spočíval v implementaci grafického editoru, simulátoru a analyzátoru těchto sítí. Hlavním důvodem pro vytvoření takového nástroje bylo, že žádný z existujících nástrojů s grafickým rozhraním nepodporuje speciální možnosti analýzy, které C-E Petriho sítě nabízejí. Pro jejich analýzu existují pouze přídavné moduly do nástrojů bez grafického rozhraní.

## 2 C-E Petriho síť

Petriho sítě jsou třídou diskrétních matematických modelů pro popis paralelních a distribuovaných systémů. Jejich základy položil německý matematik Petri Carl Adam v roce 1962. Skládají se z míst, přechodů a orientovaných hran mezi nimi. Tyto sítě rozšiřují modelovací schopnosti konečných automatů zavedením paralelního provádění přechodů a distribucí aktuálního stavu. U konečného automatu je stav jednoznačně určen jedním z jeho míst. U Petriho sítí je stav dán aktuální hodnotou v jednotlivých místech sítě.

C-E Petriho sítě jsou podtřídou obecných Petriho sítí, zkratka vychází z anglického názvu Condition-Event. Místa sítě se nazývají podmínky, modelují

logické podmínky a nabývají tedy pouze dvou hodnot. Tímto omezením rozsahu hodnot získáváme konečný počet stavů. Přesněji maximálních  $2^n$  stavů, kde  $n$  je počet podmínek C-E sítě. Konečný stavový prostor nám dává lepší možnosti analýzy vlastností sítě. U obecných Petriho sítí, kde místa mohou nabývat libovolných hodnot vzniká potenciálně nekonečný stavový prostor. Definice dále zmíněných pojmů jako C-E síť, podmínka, událost, případ, krok, výskytová síť, proces a také grafickou reprezentaci těchto sítí naleznete v [1], [4] a [5].

## 2.1 C-E systémy

Čtveřici  $\Sigma=(B,E,F,C)$  nazveme *C-E systém*, jestliže:

1. síť  $N=(B,E,F)$  je jednoduchá síť bez izolovaných prvků,  $B \cup E \neq \emptyset$
2. *případová třída*  $C \subseteq 2^B$  je třída ekvivalence vzhledem k relaci dosažitelnosti  $R_\Sigma$   
 $R_\Sigma=(r_\Sigma \cup r^{-1}_\Sigma)^*$ , kde  $r_\Sigma \subseteq 2^B \times 2^B$  je definována jako  $c_1 r_\Sigma c_2 \Leftrightarrow \exists G \subseteq E: c_1[G] > c_2$
3. pro každou událost  $e \in E$  musí existovat případ  $c \in C$ , ve kterém je událost  $e$  c-proveditelná

Případová třída reprezentuje stavy, ve kterých se modelovaný systém může nacházet. Abychom mohli C-E síť prohlásit za C-E systém, musí splnit ještě další podmínky, které jsou blíže popsány například v [5].

## 2.2 Případový graf

Graf  $\Phi_\Sigma=(C_\Sigma,H)$  se nazývá *případový graf* (case graph) C-E systému  $\Sigma=(B,E,F,C)$ , kde množina  $H=\{(c_1,G,c_2) \in C_\Sigma \times \varphi \times C_\Sigma \mid c_1[G] > c_2\}$ ,  $\varphi$  je množina všech kroků  $G \subseteq E_\Sigma$  systému  $\Sigma$ ,  $C_\Sigma \subseteq 2^B$  je případová třída a  $c_1, c_2$  jsou případy z  $C_\Sigma$ .

Graf případů je orientovaný graf zobrazující vztah mezi případy z případové třídy daného C-E systému v závislosti na provádění událostí. V grafické reprezentaci uzly grafu představují případy systému a hrany reprezentují kroky mezi příslušnými případy. Hrany jsou ohodnoceny množinou událostí představující daný krok, uzly množinou podmínek představující daný případ.

Případový graf je jednou z nejčastěji používaných možností při analýze C-E systémů. Zobrazuje uceleně všechny případy z případové třídy a všechny proveditelné události v jednotlivých případech.

## 2.3 Dopředně dosažitelné případy

C-E systémy nemají počáteční stav a z každého případu z případové třídy lze zkonstruovat stejný případový graf. Případový graf je tvořen dopředným i zpětným prováděním událostí. Pouze dopředným prováděním událostí tedy nemusíme u obecných systémů dosáhnout z jakéhokoliv případu do všech ostatních případů z případové třídy. Takovou vlastnost mají pouze cyklické systémy, kde platí  $\forall c_1, c_2 \in C_\Sigma: c_1 r^*_\Sigma c_2$ .

Zaměříme-li se například na model výrobní linky, obecně předpokládáme, že události se provádí pouze dopředným směrem, protože odpovídají provedení nějakého děje. Dále se u modelování takového systému bude vždy vycházet pouze z jednoho nebo více případů, které dávají pro daný model smysl jako počáteční stavy. Pokud

zvolíme jeden případ jako počáteční, nemusíme se tedy při dopředném provádění událostí dostat do všech zbývajících případů. Což je pro takový model zcela v pořádku, modelované děje mohou být nevratné a do počátečního stavu se již nelze vrátit.

Lze tedy zavést množinu dopředně dosažitelných případů z určitého počátečního případu. Pro systém  $\Sigma$  množinu dosažitelných případů  $D \subseteq C_\Sigma$  z nějakého počátečního případu  $c_{poč} \in C_\Sigma$  definuji jako  $D = \{ c \mid c \in C_\Sigma \wedge c_{poč} r_\Sigma^* c \}$ .

Informaci o dosažitelnosti případů je možné použít k další analýze C-E systémů. Lze si pomocí ní ověřit, jaké podmínky a stavy nemohou nastat a jaké události se nemohou provést v závislosti na počátečním stavu, ze kterého je simulace spuštěna. Tím se dostáváme k validaci systémů, kvůli které jsem množinu dosažitelných případů zavedl. U rozsáhlejších modelů je těžší ověřit, zda odpovídají modelovaným systémům pouze za pomoci simulace. Vhodné je použít případový graf, který ovšem obsahuje ze své podstaty příliš mnoho informací. Právě využitím počátečního stavu ze simulace lze poté při validaci pracovat s menší množinou případů.

### 3 Program CESim

Koncepci programu jsem navrhl na základě prostudování stávajících programů pro práci s P/T Petriho sítěmi. Název CESim je zkratkou názvu C-E Petri Net Simulator. Je určen pro operační systém Microsoft Windows a výsledný program je v obsažen jediném spustitelném souboru bez nutnosti instalace nebo dodatečných knihoven. Obsahuje tři samostatné pohledy. První pro návrh systému a jeho simulaci, druhý pro zobrazení příslušného případového grafu a třetí pro zobrazení výskytové sítě. Obsah každého pohledu lze exportovat do XML a do grafického vektorového formátu EMF. Samozřejmostí je tisk a náhled před tiskem. Protože je tento nástroj určen pro výukové účely, obsahuje výstupní okno, do kterého jsou přehledně vypisovány provedené akce a další informace o vlastnostech sítě. Tyto informace také napomáhají k pochopení evoluce systému při simulaci. Definice C-E systému je zásadní pro modelování systémů a pro jejich analýzu, program proto ve své funkční části dovolí pracovat pouze s C-E systémy.

V režimu grafického návrhu poskytuje program prostředky moderních vektorových editorů pro vytvoření všech částí grafické reprezentace sítě a k nastavení jejich vlastností z plovoucí nabídky. Kromě základních grafických objektů C-E sítě má uživatel k dispozici i grafické objekty pro zpřehlednění významu navrhované sítě.

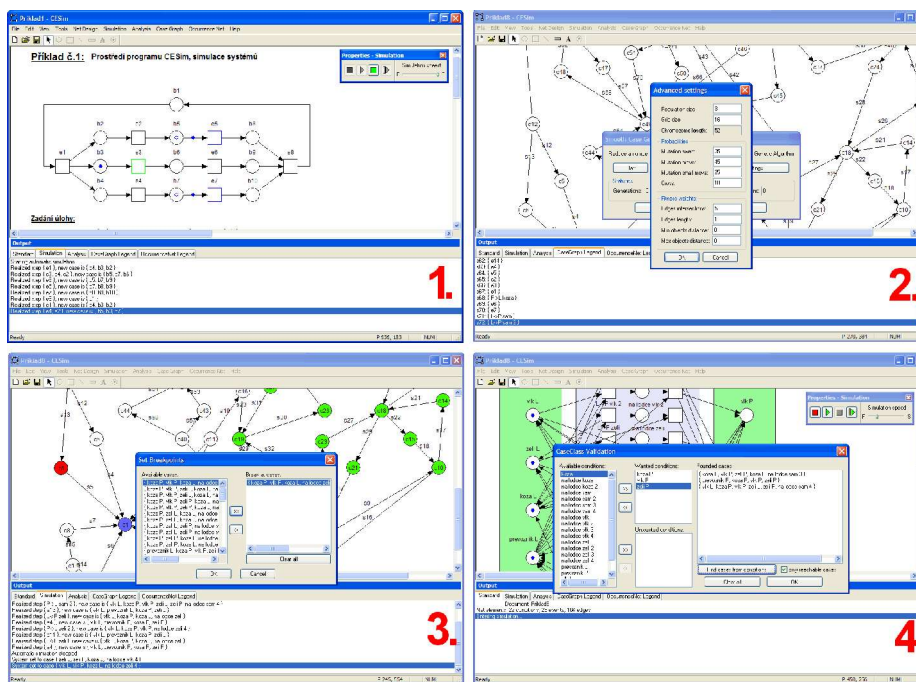
V režimu simulace poskytuje program interaktivní i automatickou simulaci, které pracují přímo s grafickou reprezentací sítě. Při vstupu do režimu simulace je provedena kontrola zda navržená síť splňuje požadavky na C-E systém a je případně indikováno, který z požadavků nebyl splněn. Změna značení při simulaci je prováděna názorně animovaným přesunem jednotlivých značek mezi podmínkami. Interaktivní simulace je řízena uživatelem. Automatická simulace je řízena programově náhodným výběrem proveditelných událostí. Uživatel má možnost měnit její rychlost, spustit krokování a nastavovat zarážky v provádění na určité podmínky systému.

V režimu simulace je dostupná také analýza C-E systému pomocí případového grafu a výskytové sítě. Případový graf je generován automaticky a lze jej automaticky rozložit po pracovní ploše s minimem průsečíků mezi hranami. Uživatel má následně možnost provést jeho korekci změnou poloh jednotlivých uzlů a případně přidat i

## Softwarový nástroj CESim pro grafický návrh, simulaci a analýzu C-E Petriho sítí

vysvětlující text. Zobrazený případový graf plně spolupracuje s interaktivní simulací systému a simulaci lze z něj i řídit. Uživatel má také možnost nad případovou třídou pokládat dotazy na dosažení určité kombinace podmínek. Další možností analýzy je zobrazení výskytové sítě, která se tvoří v průběhu celé simulace a zaznamenává tzv. proces simulovaného systému.

Na následujícím obrázku uvádím čtyři ukázky rozhraní programu CESim. Na prvním je zobrazen režim automatické simulace s právě probíhajícím animovaným přesunem značek mezi podmínkami systému. Ve spodní části okna se vypisuje průběh evoluce systému. Na druhém obrázku je zobrazen pohled na případový graf v režimu návrhu s otevřeným oknem nastavení genetického algoritmu pro automatické rozložení grafu. Na třetím obrázku je zobrazen pohled na případový graf v režimu simulace s otevřeným oknem pro nastavení zářezek v provádění simulace. Jsou zde také vidět zeleně zvýrazněné dopředně dosažitelné případy. Na posledním obrázku je uveden složitější systém v režimu simulace s otevřeným oknem pro jeho validaci.



Obr.1. Ukázky rozhraní programu CESim

## 4 Implementace

Program jsem implementoval pomocí vývojového nástroje Microsoft Visual C++ 6.0. Celé rozhraní programu i zdrojové kódy včetně komentářů jsou psány v anglickém jazyce. Implementace zabírá přibližně 19 tisíc řádků. Důraz při vývoji jsem kladl především na přehlednost a jednoduchost ovládání a na snadnou rozšiřitelnost programu o nové funkce. Program je objektově orientovaný. Grafické části sítě jsou

reprezentovány třídami, které nesou jejich vlastnosti a starají se i o samotné vykreslení na pracovní ploše. Uchovávají jsou jako seznam ukazatelů na jejich společnou virtuální bázeovou třídu.

Rozhraní aplikace používá architekturu SDI typu dokument-pohled. Tato architektura odděluje data dokumentu od jejich zobrazení a přináší zpřehlednění odpovědnosti jednotlivých tříd za data. Veškerá data aplikace nese třída dokumentu, která poskytuje rozhraní pro jejich zpřístupnění. Zobrazení dat mají na starosti příslušné třídy pohledu.

#### 4.1 Algoritmus vytvoření případového grafu

Výpočet případového grafu tvoří stěžejní část pro následnou analýzu C-E systémů v programu CESim. Tento algoritmus je u větších C-E systémů výpočetně i paměťově náročný, bylo proto nutné provést řadu optimalizací především kvůli problému stavové exploze. Pro představu krok sestávající se  $zn$  událostí generuje  $2^n - 1$  nových uzlů případového grafu. Například již poměrně malý C-E systém obsahující osm událostí proveditelných v jednom kroku vygeneruje případový graf, který bude mít 256 případů a 6305 hran mezi nimi.

Algoritmus výpočtu případového grafu:

1. Vloží do grafu počáteční případ, ze kterého byla spuštěna simulace
2. Prochází existující případy v grafu a pro každý provádí následující body
3. Vytvoří množinu kroků v daném případě následujícím postupem
  - a. Nejprve vytvoří množinu proveditelných událostí v tomto případě
  - b. Z této množiny vytvoří seznam všech skupin událostí, tedy kombinací událostí mezi sebou. Nejprve uvažuje samostatné události, potom více událostí současně a až poslední množina bude obsahovat všechny události.
  - c. Postupně prochází seznam skupin událostí a testuje vstupní a výstupní množiny všech událostí, které jsou spolu ve skupině. Pokud mají některé události společné prvky vstupní nebo výstupní množiny, jsou spolu v konfliktu a ze seznamu se tato kombinace odstraní. V seznamu skupin událostí tedy ponechá jen kroky.
4. Prochází množinu kroků a provádí je dopředným i zpětným směrem
5. Přidá do grafu nově vytvořené případy
6. Vytvoří propojení případů grafu hranami podle provedeného kroku
7. Pokračuje krokem 2, dokud neprojde všechny případy v grafu

#### 4.2 Automatické rozložení případového grafu

V programu jsem implementoval automatické optimální rozložení případového grafu pomocí genetického algoritmu. Genetický algoritmus je stochastický algoritmus prohledávání stavového prostoru za pomoci populace jedinců. Jedince v populaci nazýváme chromozomy a je v nich zakódováno hledané řešení. Pro ohodnocování chromozomů slouží funkce vhodnosti, která udává jako moc se dané řešení blíží

## *Softwarový nástroj CESim pro grafický návrh, simulaci a analýzu C-E Petriho sítí*

optimálnímu řešení. Genetický algoritmus využívá pro obnovu populace genetické operátory křížení a mutace.

Při spuštění optimalizace je nejprve provedena příprava populace jedinců vygenerováním grafů s náhodně rozmístěnými uzly a je provedeno jejich ohodnocení. Samotný genetický algoritmus potom v cyklu provádí výběr dvou jedinců z populace za pomoci turnajové selekce, jejich případné křížení a mutaci a umístí je do nové populace. Jedná se o generativní algoritmus s úplnou náhradou rodičovské populace za populaci potomků s využitím jednoduchého elitizmu. Po obnově populace je provedeno ohodnocení nové populace a cyklus se opakuje. Bližší vysvětlení zmíněných pojmů a další informace o genetických algoritmech naleznete v [3].

### *Kódování chromozomů*

K zobrazení grafu používám pomocnou čtvercovou mřížku tak, že uzly případového grafu mohou být umístěny pouze na souřadnicích v této mřížce. Tím zajišťuji minimální vzdálenost mezi uzly a také zmenšuji prohledávaný stavový prostor. Uživatel má možnost velikost této mřížky měnit. V chromozomu jsou zakódovány pozice jednotlivých uzlů grafu v mřížce a počet genů v chromozomu tedy odpovídá počtu uzlů. Pro reprezentaci jednotlivých genů používám celočíselné kódování jejich souřadnic. Propojení uzlů není součástí chromozomu, protože je pro všechny jedince společné a v průběhu optimalizace se nemění.

### *Výpočet fitness funkce*

Funkce vhodnosti jedince zohledňuje počet protínajících se hran, jejich délku a vzdálenost mezi jednotlivými uzly grafu. Lepší jedinci mají větší hodnotu fitness funkce. Výpočet se skládá z výpočtu čtyř základních částí této funkce. První je počet průsečíků mezi všemi hranami navzájem. Neprovádím výpočet průsečíků pro každou hranu s každou navzájem, ale vždy jen s hranami následujícími v poli hran. Počet testů pro  $n$  hran tedy není  $n^2$ , ale jen součet aritmetické posloupnosti, tedy  $(n/2)*(1+n)$ . Pro další urychlení obsahuje každá hrana i svůj ohraničující obdélník a nepočítám průsečík mezi hranami, které nemají žádnou oblast společnou a nemohou se tedy ani protínat. Druhá část výpočtu fitness funkce počítá sumu délek hran mezi propojenými uzly. Třetí a čtvrtá část počítá sumu minimálních a maximálních euklidovských vzdáleností mezi jednotlivými uzly.

Každá část funkce má svoji váhu, jakou se podílí na celkové funkci. Aby bylo možné použít váhy, provádím nejprve normování jednotlivých částí na hodnoty od nuly do jedné. Pokud je některá z vah nulová, příslušná část se pro urychlení algoritmu vůbec neprovádí.

### *Operátor křížení*

Způsob kódování chromozomů vyžaduje speciální operátor křížení, který zachovává správnou permutaci. Zvolil jsem proto operátor křížení pořadím. Nejprve jsou náhodně zvoleny dva body křížení. První potomek dědí geny mezi těmito dvěma body z prvního rodiče ve stejném pořadí a na stejných pozicích, jak jsou uloženy v rodiči. Zbývající geny jsou zděděny z druhého rodiče v pořadí, ve kterém se vyskytují v jeho chromozomu. Druhý potomek obdobně dědí geny mezi body křížení přímo z druhého rodiče a zbývající jsou doplněny z prvního rodiče.

### *Operátory mutace*

V programu používám tři operátory mutace. Zvolil jsem je tak, aby při prohledávání stavového prostoru dovolily plně pokrýt celou mřížku. První z nich je mutace záměnou dvou náhodných genů v chromozomu. Dojde tím k prohození souřadnic dvou uzlů. Tento operátor je sám o sobě dost silný a dokáže výrazně zmenšit počet protínajících se hran. Druhý operátor přesune náhodně zvolený uzel na nové náhodně zvolené souřadnice. Tento jednoduchý operátor je nejdůležitějším pro nalezení správného řešení. Třetí operátor přesunuje náhodně zvolený uzel pouze o jednu pozici v mřížce. Posune jej tedy náhodně na některou z osmi možných sousedních pozic.

## **5 Závěr**

Hlavním přínosem mé práce je program CESim, který jsem v rámci projektu vytvořil. Jak vyplývá z mého průzkumu stávajících nástrojů, který jsem provedl v rámci diplomové práce [4], doposud neexistoval vhodný programový nástroj pro práci s C-E Petriho sítěmi. Vytvořený nástroj CESim tuto mezeru důstojně zaplnil a je plně srovnatelný s podobnými programy ve své kategorii. V projektu jsem také zavedl nový pojem množiny dopředně dosažitelných případů z určitého počátečního případu při simulaci C-E systému. Tuto množinu případů lze použít k dodatečné analýze C-E systému a uživatel má v případovém grafu tyto informace k dispozici. Program CESim bude sloužit především studentům jako nástroj pro seznámení s teorií C-E Petriho sítí. Navrhl jsem k tomuto účelu několik zadání úloh, které představují možnosti programu CESim a samotných C-E Petriho sítí.

Za pozornost stojí také navržený genetický algoritmus pro optimalizaci zobrazení případového grafu. Tento algoritmus plně splnil požadavky, které jsem na něj v mém projektu kladl. Pro optimální zobrazení grafu je možné použít i analytické metody, například pružinový model [2]. Ovšem výhodou genetického algoritmu pro můj projekt spočívá ve velké variabilitě jeho optimalizační funkce, kdy uživatel může zvolit pro každý graf jiné parametry optimálnosti zobrazení. Další velkou výhodou genetického algoritmu je možnost přesně řídit dobu optimalizace. Již po několika desítkách sekund je v případovém grafu značně zredukován počet protínajících se hran a stačí tedy pro rychlé seznámení s grafem pro určení vlastností C-E systému. Pokud chceme případový graf dále prezentovat, je možné nechat běžet optimalizaci delší dobu a dosáhnout optimálního zobrazení.

### **5.1 Další vývoj programu**

Program je od začátku vyvíjen tak, aby byl snadno rozšiřitelný. Do budoucna se předpokládá pokračování ve vývoji programu především v jeho možnostech analýzy C-E Petriho sítí. A to především v rozšíření o výpočet tzv. synchronizační vzdálenosti a o tzv. fakta C-E sítí. Tomuto tématu se podrobněji věnuji v mé diplomové práci [4].

Samostatný rozvoj si zaslouží i vyvinutý genetický algoritmus pro optimalizaci zobrazení případového grafu. Směr rozšíření vidím především v doplnění heuristiky do operátorů křížení a mutace, aby byl algoritmus rychlejší. Zajímavou možností jsou také například operátory, které místo změny pozic uzlů mění přímo pozice hran.

Program CESim bude umístěn do databáze nástrojů pro práci s Petriho sítěmi (<http://www.daimi.au.dk/PetriNets/tools/>). Tato databáze sdružuje volně šiřitelné i komerční programy pro všechny rozšířené operační systémy. Program zde bude

### *Softwarový nástroj CESim pro grafický návrh, simulaci a analýzu C-E Petriho sítí*

k dispozici s otevřenými zdrojovými kódy. Věřím, že program CESim bude užitečný i pro studenty na jiných univerzitách. To byl také jeden z důvodů proč bylo rozhraní programu i implementace provedena v anglickém jazyce.

#### **Literatura**

1. Češka, M.: *Petriho sítě*. Akademické nakladatelství CERM, Brno, 1994.
2. Demel, J.: *Grafy a jejich aplikace*. Academia, Praha, 2002.
3. Kvasnička, V., Pospíchal, J., Tiňo, P.: *Evoluční algoritmy*. STU Bratislava, 2000.
4. Novosad, P.: *Programové nástroje pro práci s C-E Petriho sítěmi [diplomová práce]*. FIT VUT Brno, 2004.
5. Reisig, W.: *Petri Nets – an Introduction*. Springer-Verlag, Berlin, 1985.

#### **Annotation:**

##### *Software tool CESim for graphical design, simulation and analysis of C-E Petri nets*

This paper presents a new computer tool CESim for editing, simulating and analyzing of the C-E Petri nets. These nets are subclass of general Petri nets. The developed program CESim consists of a graphical editor for a net design and an automatic and a manual simulator. The tool also provides facilities for analyzing C-E systems by case graphs and occurrence nets. New term forward reachable cases are introduced. A genetic algorithm for automatic drawing of the case graph was also developed. Algorithms used for an analysis of the C-E Petri nets are described. The tool is prepared for further expansion in analyzing nets by computing synchronic distances and implementing facts of a net.