

Návrh a implementácia frameworku pre Petriho siete v jazyku C#

Fakulta informatiky

Paneurópska vysoká škola

Mgr. Michal Drobný

Školiteľ: prof. RNDr. Gabriel Juhás, PhD.

Motivácia a problém

V posledných rokoch sa v priemysle, logistike aj mechatronike zvyšuje miera distribúcie a paralelizmu. Z pohľadu modelovania a verifikácie to kladie dôraz na formálne presné, ale zároveň prakticky použiteľné nástroje. Tradičné riešenia často vynikajú buď v akademickej presnosti, alebo v inžinierskej integrácii – len zriedka v oboch.

V praxi preto chýba nástroj, ktorý by spojil výhody Petriho sietí (jednoznačná sémantika, čitateľný grafický zápis) s modernou vývojárskou skúsenosťou a prevádzkovými požiadavkami (.NET ekosystém, REST API, monitoring, auditovateľnosť). Potreba je najvýraznejšia pri rýchlom prototypovaní, what-if analýzach a spätnom prehrávaní udalostí.

Cieľom práce je skrátiť cestu od modelu k spoľahlivej simulácii – so zrozumiteľnou API vrstvou, prepojením na webové rozhranie a perzistentným ukladom stavov. Dôraz kladím na deterministické opakovanie behov, sledovanie trasovania (traces) a prípravu na škálovanie bez nutnosti meniť modely.

Riešenie

Riešenie je navrhnuté vrstvene: modelová vrstva, simulačný engine, REST API a prezentačná webová aplikácia. Interný objektový model reprezentuje miesta, prechody a hrany; vývojár pracuje s typovo bezpečnými entitami a builder rozhraním (PetriNetBuilder), ktoré skraca čas od načítania modelu k spusteniu simulácie.

Simulačný engine implementuje štandardné podmienky umožnenia a spustenie prechodov, podporuje krokovanie, batchované spracovanie a kontrolu konfliktov. Dôležitou vlastnosťou je deterministické správanie pri rovnakých vstupných podmienkach a možnosť zachytávať snímky stavu (snapshots) pre rýchly návrat alebo paralelné experimenty.

Komunikačnú vrstvu tvorí REST API (dotazy na marking, povolené prechody, odpálenie, reset, export udalostí). Webová aplikácia (Laravel + Livewire) uľahčuje interaktívnu prácu v reálnom čase. Perzistencia v MongoDB ukladá stav, udalosti a diagnostické informácie – čo umožňuje audit a vizualizáciu priebehov.

Výsledky a testovanie

Testovanie prebieha na troch úrovniach: UNIT testy jadra, integračné testy API a systémové testy s referenčnými modelmi.

Výkon overujem podľa spočítaných nákladov na prechody; optimalizujem kontrolu incidencie a cacheovanie podmienok, meriam latenciu krokov, pamäť a rast logu.

Robustnosť testujem injektovaním chýb, viacnásobnými a paralelnými volaniami API. Výsledkom je prostredie vhodné na opakovateľné experimenty s konzistentnými stavmi a transparentným logovaním.

Využitie a budúcnosť

Bezprostredné využitie je v rýchlom prototypovaní procesov. Jednoduché REST rozhranie umožňuje procesy automatizovať a kombinovať simulačné scenáre s externými nástrojmi vizualizácie. Dôležitá je interoperabilita: framework je pripravený na integráciu s OPC UA a na prepojenie so simulačnými prostredím Factory I/O.

Do budúca plánujem rozšírenia o časové a farebné Petriho siete, podporu stochastických prechodov, základné model-checking vlastnosti a pokročilejšiu vizualizáciu stavov v čase. Cieľom je stabilné SDK, kvalitná dokumentácia a knižnica príkladov, ktoré uľahčia adopciu v akademickej aj priemyselnej praxi.