UNITY

UNITY bol navrhnutý a popísaný v knihe Parallel Program Design – A Foundation ktorej autormi sú K. Mani Chandy a Jayadev Misra z University of Texas. Teória UNITY sa snaží oddeliť (programátorské) rozmýšľanie ohľadne problému – programu od vlastnej implementácie na konkrétnu architektúru. Detaily ohľadne implementácie sú zvažované až pri mapovaní navrhnutých programov na konkrétnu architektúru. Tým vlastne ponúka možnosť navrhovať dizajn programu – abstraktné riešenie, nezávislé na konkrétnej paralelnej architektúre, čím sa rieši problém ktorý nastáva ak je program pevne naviazaný na konkrétnu architektúru ktorá časom zastarala.

Základné črty UNITY

Základnými črtami teórie sú: nedeterminizmus, absencia toku riadenia (control- flow), synchrónnosť a asynchrónnosť, stavy a priradenia (prechodový systém).

Nedeterminizmus

Rôzne spustenia programu môžu vykonávať príkazy v rozdielnom poradí, môžu spotrebovať rôzne množstvá zdrojov, prípadne vyprodukovať rozdielne výsledky. Nedeterminizmus umožňuje vytvárať jednoduché programy, ktorých jednoduchosť je dosiahnutá vďaka tomu že sa vyhneme zbytočnému determinizmu.

Absencia control-flow

Procesy sú definované ako sekvenčné a tým sa teda ponúka možnosť viacerých súbežných - paralelných tokov riadenia. Rozdielne paralelne modely (kooperujúce sekvenčné procesy, paralelne komunikujúce sekvenčné procesy, alternujúce stroje, paralelne RAM, booleovské obvody, atď.) a z nich vychádzajúce architektúry definujú pre programy rôzne formy toku riadenia.

Synchrónnosť a asynchrónnosť

Synchrónne a asynchrónne udalosti musia byť v základoch každej zjednocujúcej teórie paralelného programovania. Delenie systémov na synchrónne aasynchrónne je umelé – neprirodzené. Preto aj UNITY obsahuje synchrónnosť a asynchrónnosť ako základné koncepty.

Stavy a priradenia (prechodový systém)

Systém prechodov stavov v UNITY je reprezentovaný počiatočným stavom a transformáciami, ktoré sú popísané pomocou konštruktov programovacích jazykov a to konkrétne pomocou premenných a priradení. Stav programu je teda definovaný aktuálnymi hodnotami premenných, a prechod do iného stavu sa udeje cez sériu priradení.

Pokračovanie

Vykonávanie programu začína vľubovoľnom stave programu ktorý splňuje počiatočné podmienky. Program sa vykonáva donekonečna. Vkaždom kroku výpočtu je nedeterministický vybratý ľubovoľný priraďovací príkaz aje vykonaný. Každý príkaz je vykonaný nekonečne veľa krát.

Štruktúra programov

program-name je ľubovoľný textový reťazec. Kľúčové slovo sekcie môže byť vynechané ak je daná sekcia prázdna.  
declare-section menuje premenné požité v programe a ich typy. Syntax je podobná deklarovaniu premenných v PASCALe a preto ju ďalej nebudeme popisovať. Zväčša stačí používanie typov integer a boolean ako základných typov. Polia, množiny a sekvencie týchto základných typov sa tiež využívajú.

always-section je miesto kde sa definujú isté premenné ako funkcie iných premenných. Táto sekcia nie je nevyhnutná pre písanie UNITY programov, ale jej používanie je v niektorých prípadoch veľmi vhodné.

initially-section sa používa na definovanie počiatočných hodnôt niektorých premenných. Premenné ktoré nie sú inicializované majú ľubovoľnú počiatočnú hodnotu.

assign-section obsahuje množinu assignment statements (priraďovacích príkazov)

Koniec

Vykonávanie programu začne v stave, kde premenné majú svoje počiatočné hodnoty. Vkaždom kroku je vyhodnotený ľubovoľný statement (príkaz). Statements (príkazy) sú vyberané náhodne, ale pri nekonečnom vykonávaní programu sa každý príkaz vykoná nekonečne veľa krát. Stav programu sa vola fixed point (fixný bod) vtedy a len vtedy ak vykonanie ľubovoľného statement-u (príkazu) zanechá stav nezmenený a teda že všetky premenné budú obsahovať rovnakú hodnotu ako pred vykonaním statement-u. Program nemá žiadne vstupné ani výstupné statement-y (príkazy). Nie sú povolené ani hierarchické štruktúry programov.