OVĚŘOVÁNÍ SOFTWARU BEZPEČNOSTNÍCH SYSTÉMŮ JADERNÉ ELEKTRÁRNY

Ing. Stanislav Roubal - I & C Energo a.s., Pražská 684, 674 01 Třebíč, sroubal@ic-energo.cz
Ing. Zdeněk Piroutek - I & C Energo a.s., Pražská 684, 674 01 Třebíč,zpiroutek@ic-energo.cz
Ing. Michal Kmínek - I & C Energo a.s., Pražská 684, 674 01 Třebíč, mkminek@ic-energo.cz

ABSTRACT:

There is given example of the quality assurance solution during last phases of the software life cycle in the safety systems of the nuclear equipment - nuclear power plant (NPP) Dukovany and Temelin. At first overview of requirements on the quality assurance for this "safety critical software" is given. The following information provide two approaches to software testing: The first approach is simulation models utilization at the commissioning of NPP Temelin to verify algorithms of the safety systems based on the analysis of the measured transients, at the software verification of the real systems in the framework of the I&C modernization of NPP Dukovany. The second approach is the static analysis utilization of the source code of the safety system of NPP Temelin. In both cases there is the compliance verification of the design and partial systems implementation with the system requirements on functionality.

ABSTRAKT:

V referátu je presentován příklad řešení problematiky zabezpečení jakosti softwaru v činnostech závěrečných fází vývojového procesu softwaru. Jedná se o bezpečnostní systémy jaderných zařízení - JE Dukovany a Temelín. Nejprve je uveden přehled požadavků na zabezpečení jakosti pro tento "safety critical software". Následující informace poskytují dva přístupy k testování softwaru: Prvním je použití simulačních modelů pro ověření algoritmů bezpečnostních systémů při spouštění JE Temelín na základě analýzy změřených přechodových procesů a při ověřování softwaru reálných systémů v rámci Obnovy systému kontroly a řízení (SKŘ) JE Dukovany. Druhým je použití statické analýzy zdrojových programů bezpečnostních systémů JE Temelín. V obou těchto případech jde o ověření souladu návrhu a provedení dílčích systémů se systémovými požadavky na funkcionalitu.

KLÍČOVÁ SLOVA:
jakost softwaru, testování, bezpečnostní systémy

1. ÚVOD

V ČR byly provedeny dvě významné etapy implementace softwaru na jaderných elektrárnách: systémy RCLS (limitační a řídící systém reaktoru), PRPS (ochranný systém reaktoru), DPS (diversní ochranný systém), PAMS (systém pohavarijního monitorování) v rámci uvádění do provozu bloku na JE Temelín a systémy RCS (systém řízení reaktoru), RTS (systém odstavení reaktoru) a další v rámci Obnovy SKŘ na JE Dukovany. Na dodávce těchto softwarových systémů se podílely zahraniční a české firmy.

Požadavky na zabezpečování jakosti softwaru jsou v systémech jaderných elektráren na mezinárodní úrovni formulovány Mezinárodní agenturou pro atomovou energii (IAEA) jako „Nuclear Safety Standards“. Českou legislativu pro oblast jakosti a zajišťování jaderné bezpečnosti pak tvoří tzv. „atomový zákon“ č.18/199Sb a zejména vyhlášky Státního úřadu

pro jadernou bezpečnost (SÚJB), zejména č. 214/1997Sb. Na úrovni mezinárodní legislativy pak normy jako řady IEC, ISO a IEEE pro oblast tvorby softwaru a zajištění bezpečnosti.

2. POŽADAVKY NA TVORBU SOFTWARU

Pro životní cyklus softwaru systémů JE je stanoven požadavek, že to musí být dobře strukturovaný cyklus založený na standardním cyklu životnosti obsahující následující procesy:

- plánovací proces,
- vývojový proces,
- průřezové procesy,
- provozní procesy.

Cílem činností, prováděných v rámci plánovacího procesu, je vytvoření souboru dokumentace, sloužící k řízení a kontrole celého životního cyklu softwaru. Vývojový proces software je rozdělen do následujících fází: specifikace požadavků na software, návrh softwaru, implementace softwaru, integrace softwaru a integrace software-hardware, testování. Průřezové procesy obsahují činnosti verifikace a validace, činnosti řízení konfigurace a činnosti z oblasti analýz nebezpečí. Provozní procesy sestávají z instalace u uživatele, provozu a údržby u uživatele.

2.1. Program testování

Testování softwaru realizujícího bezpečnostní funkce kategorie A podle IEC 61226 probíhá ve dvou hlavních etapách:

- testování samotného softwaru jednotlivých procesních jednotek. Náplní jsou:
  - testy komponent softwaru (statické a dynamické testy),
  - integrační testy softwaru
  - validační testy celkového softwaru jednotlivých procesních jednotek na soulad s návrhem softwaru a na soulad s požadavky stanovenými ve fázi specifikace požadavků na software.
- testování jednotlivých systémů jako celku. Testováním systémů jako celku tzv. *„interconnected tests“* je ověřován soulad návrhu a provedení jednotlivých systémů se systémovými požadavky na funkcionalitu a výkonnost.

V rámci Obnovy SKŘ JE Dukovany je dodávka technických, programových prostředků řídícího systému zajišťující bezpečnostní funkce včetně automatizovaných softwarových nástrojů založena na platformě SPINLINE 3 vyvinutého francouzskou firmou Schneider Electric Industries (nyní Data System & Solution). Provádění testů je automatizováno použitím řady softwarových nástrojů jak na úrovni testování komponent, tak na úrovni integračních testů typu „bílá skříňka" či validačních testů typu „černá skříňka“. Soubory testovacích dat jsou konfiguračními položkami spravovanými v rámci dodaného vývojového prostředí CLARISSE.

V rámci JE Temelín byla dodávka technických a programových prostředků řídícího systému pro bezpečnostní funkce založena na platformě EAGLE vyvinutého firmou Westinghouse. Testování SW bylo provedeno nezávislým posouzením na základě statické analýzy typu „bílá skříňka" za použití automatizovaného nástroje MALPAS vyvinutého anglickou firmou TAG Ltd.

3. POUŽITÍ SIMULAČNÍHO MODELU PRO TESTOVÁNÍ SOFTWARU

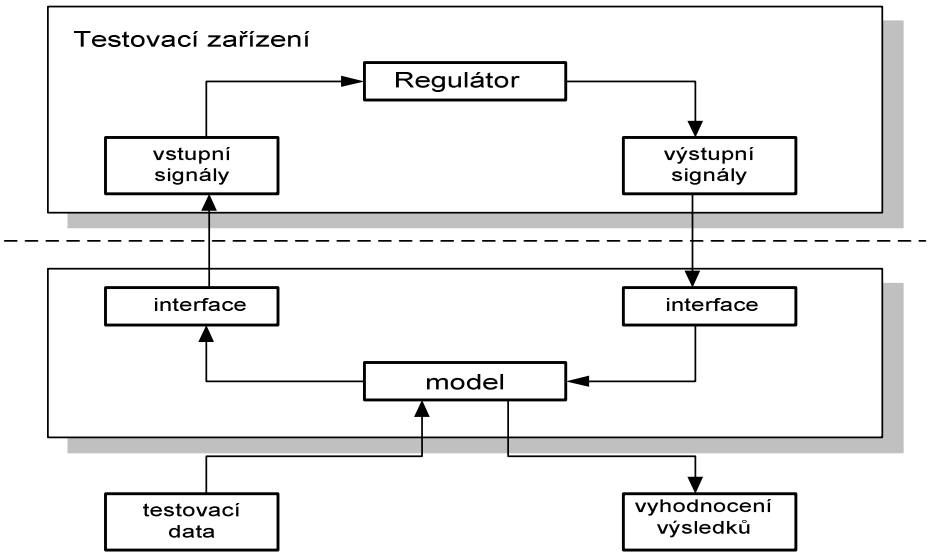
Simulační modely umožňují ověření chování projektových iniciačních událostí, analýzu vzniklých přechodových procesů, návrh a seřízení algoritmů řídicích a bezpečnostních systémů, zvýšení spolehlivosti a bezpečnosti systému, optimalizaci provozu. Pro tyto účely byly vytvořeny simulační modely SIMED a DYTE prezentující dynamické chování technologie, řídícího a bezpečnostního systému JE Dukovany a Temelín. Modely jsou použity především pro predikci dynamického chování bloku JE Dukovany a Temelín v normálních a abnormálních podmínkách provozu bloku.

Přestože existuje řada metod a softwarových automatizovaných nástrojů pro testování softwaru na různých úrovních jeho tvorby a také byly pro testování na JE Dukovany a Temelín použity, byly uvedené simulační modely použity k testování „reálných“ softwarových systémů jako další testovací metoda. Způsob použití je popsán níže.

3.1 Testování softwaru v uzavřené regulační smyčce (JE Dukovany)

Testování softwaru (jedná se především o regulační obvody) v uzavřené regulační smyčce je rozšířením klasických dynamických testů. Tyto testy jsou v podstatě validačními testy určenými pro ověření funkcionality aplikačního softwaru - samotného regulátoru. Jsou to testy typu "černá skříňka" a testy jsou voleny tak, aby zahrnovaly kombinaci vstupů a výstupů co možná se 100% pokrytím. Testování je však prováděno v otevřené smyčce.

Obr. 1 představuje testování regulačního obvodu v uzavřené regulační smyčce. Smyčka je složena z regulátoru a soustavy (model). Při testování v otevřené smyčce (horní část obrázku) jsou vstupní signály do regulátoru generovány a představují signály normálních i havarijních situací, které mohou nastat během provozu. Výstupní signály z regulátoru se pak vyhodnocují jako odezva regulátoru na tyto vstupní signály a porovnávají proti očekávanému chování samotného regulátoru.

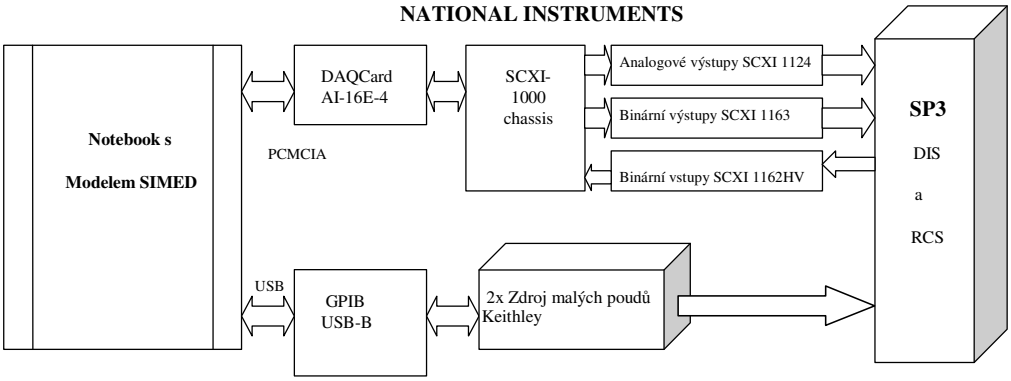


Obr 1: Základní uspořádání dynamického testování v uzavřené regulační smyčce

I když je prokázána shoda s požadavky na daný systém (regulátor), neznamená to jednoznačně, že tento systém se bude v rámci uzavřené smyčky chovat "správně". Při testování v uzavřené smyčce je testovacími daty především ovlivněna soustava a činnost regulátoru je posuzována z pohledu chování bloku jako celku, tj jako chování reálného objektu.

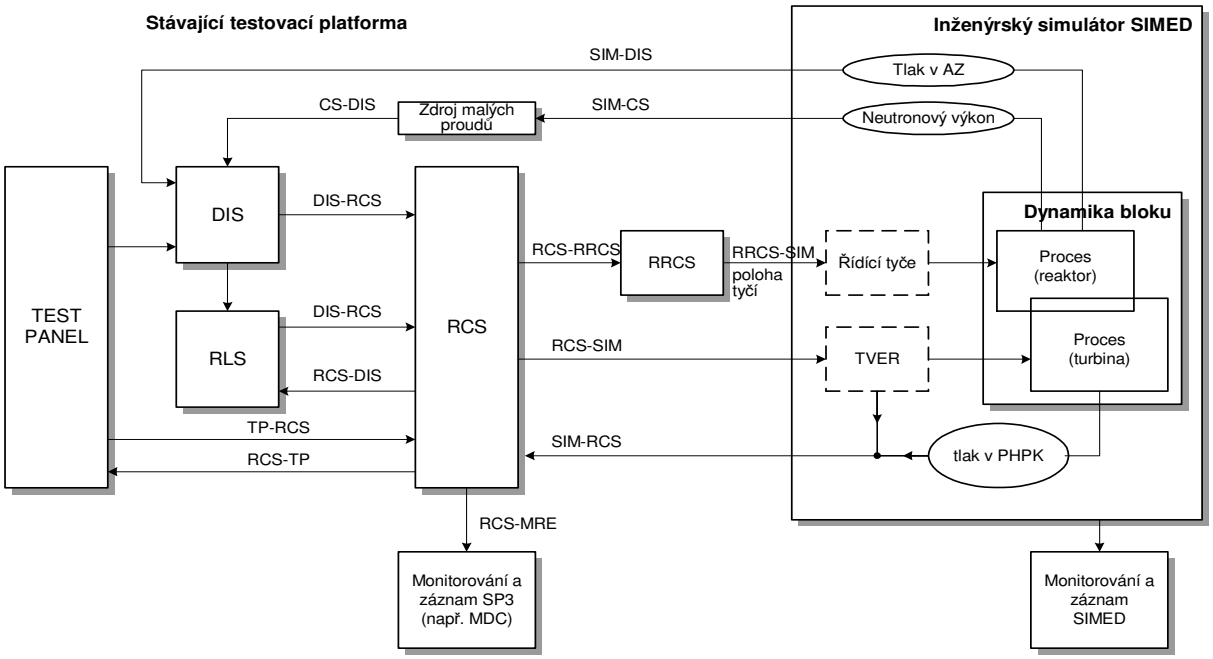
V rámci Obnovy SKŘ na JE Dukovany byl při testování části digitálního řídícího systému SPINLINE 3 vznesen požadavek pro ověřování algoritmů s cílem co nejvíce se přiblížit reálným podmínkám. Na základě tohoto zadání bylo rozhodnuto u digitálního regulátoru RCS (regulátor řízení reaktoru, kde algoritmy ze stávajícího analogového regulátoru ARM (regulátor reaktoru sovětské výroby) byly převedeny do softwarové podoby) provést testování v uzavřené smyčce za použití inženýrského simulátoru SIMED. Toto testování znamenalo, že v simulátoru SIMED byl model regulátoru reaktoru RCS nahrazen reálným regulátorem implementovaným v konečné hardwarové podobě v mikroprocesoru. Testování bylo prováděno ve dvou fázích. První fáze probíhala u výrobce ve Francii, kdy byl nahrazen pouze samostatný regulátor reaktoru, druhá fáze probíhala na JE Dukovany na testovací divizi, kdy byl kromě regulátoru RCS použit i reálný systém pro řízení pohybu kazet (RRCS). Tato druhá fáze měla prověřit spolupráci zařízení RCS a RRCS a vliv zpoždění signálů mezi těmito systémy na regulační proces.

Pro potřeby propojení inženýrského simulátoru k reálnému regulátoru vytvořenému na platformě SP3 bylo vytvořeno následující rozhraní. Měřicí ústředna National Instruments byla s notebookem, na kterém byl instalován simulátor SIMED, propojena pomocí měřící karty. Pomocí ústředny byly realizovány analogové proměnné z technologie ve formě proudové smyčky a stavy regulátorů. Zdroj malých proudů, zařízení Keithley, připojený přes sběrnici GPIB, a to přes převodník USB/GPIB, sloužil k simulaci signálů z neutronových detektorů reprezentujících výkon reaktoru.



Obr 2: Schéma uspořádání HW komponent pro testování v uzavřené regulační smyčce

Pro testování v uzavřené smyčce byl připraven soubor testovacích scénářů, který vycházel z provozního předpisu JE Dukovany pro testování během najíždění bloku. K testům bylo přidáno i několik dynamických průběhů (výpadek turbín apod.).



Obr 3: Schéma uspořádání SW komponent pro testování v uzavřené regulační smyčce

Vysvětlení jednotlivých použitých zkratek:
DIS - Digital Instruments System - systém na platformě SPINLINE 3 sloužící ke zpracování signálů z procesu – analogové a binární vstupy, vstupy z neutronových detektorů;
RLS - Reactor Limitation System - Limitační systém reaktoru (platforma SPINLINE 3)
RCS - Reactor Control System - Regulátor reaktoru (platforma SPINLINE 3)
Test Panel - panel k simulaci zásahů operátora a pro signalizaci o stavu regulátoru reaktoru
RRCS - Reactor Rod Control System - systém pro řízení pohonů reg. tyčí reaktoru (platforma fy. Škoda Energo Technologie).)
TVER - Regulátor turbíny
PHPK - Tlak v hlavním parním kolektoru
AZ - aktivní zóna reaktoru

Během podrobného testování v uzavřené smyčce se podařilo odhalit dva nedostatky, které nebyly zjištěny během standardního testování v rámci FAT zkoušek (zkoušky u výrobce), protože testy FAT jsou zkouškami v otevřené smyčce. Tyto nedostatky by byly zjištěny během spouštění, ale jejich odstranění by si vyžádalo zásah do aplikačního softwaru. Prvním nedostatkem byla nevhodně zvolena hodnota hystereze v režimu regulace výkonu. Druhým nedostatkem bylo nastavení zesílení ve zpětné vazbě s integrátorem. V tomto případě se potvrdilo, že nelze jednoduše převést regulaci z analogové techniky na digitální. Řešením byly změny v konfiguračních datech a změny v následující verzi softwaru.

Testování regulačního obvodu v uzavřené smyčce pomocí inženýrského simulátoru SIMED potvrdilo oprávněnost jeho použití. Kromě uvedeného příkladu interface (použití měřící ústředny) pro napojení na reálné zařízení je možno také použít protokol Ethernet.

3.2 Testování softwaru off-line (JE Temelín)

Inženýrský simulátor DYTE (obdoba simulátoru SIMED) byl použit pro modelování dílčích zkoušek zařazených do energetického spouštění JE Temelin. Simulátor zahrnoval funkční modely jednotlivých softwarových systémů (PRPS-ochranný systém reaktoru, RCLS-limitační a řídící systém reaktoru, založené na platformě EAGEL a systém PCS-řídící systém bloku postavený na platformě řídících systémů fy. Westinghouse).

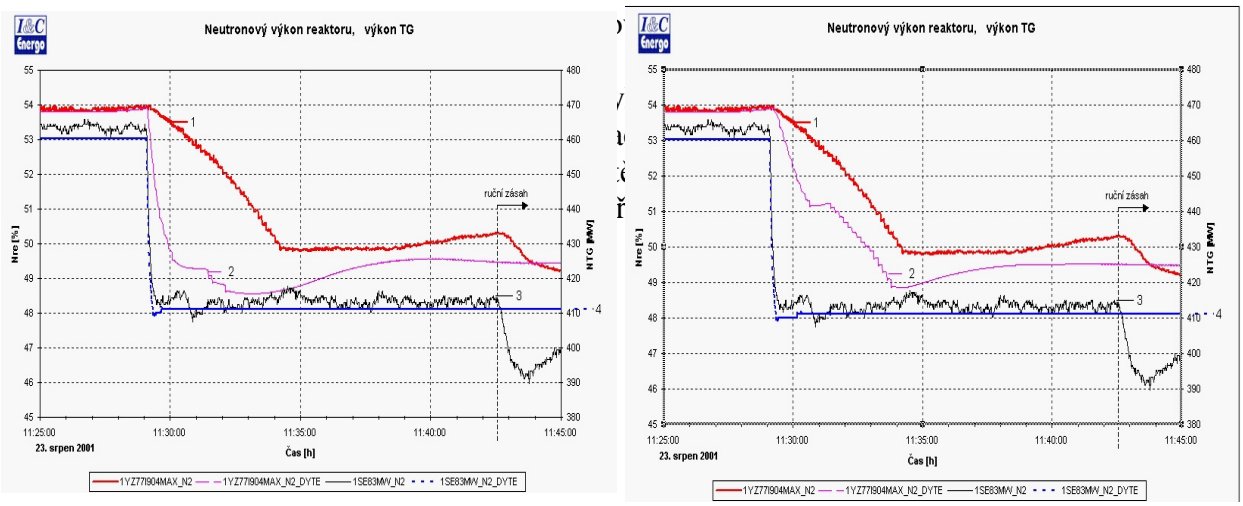
Významnou částí simulací v průběhu spouštění 1. bloku bylo vyhodnocení funkcí systémů RCLS a PCS. Dílčí analýzy se zabývaly ověřením součinností hlavních regulací bloku se zaměřením na nastavení parametrů systému RCLS a hlavních regulátorů sekundárního okruhu. Výchozí nastavení bylo převzato z projektové dokumentace. S tímto nastavením byly prováděny analýzy a při nich identifikovány nedostatky systémů. V rámci plnění jednotlivých etap zkoušek byla provedena řada výpočtů na různých výkonových hladinách, které sloužily jako podklad pro přípravu zkoušek a pro instruktáž operátorů. Umožnily odhalit slabá místa zkoušek a tím přijmout předem taková opatření, aby se zabránilo předčasné aktivaci systému RCLS.

Výsledky predikčních výpočtů dynamického chování bloku sloužily jako:

- technická podpora pro odhad chování bloku,
- eliminace rizik během realizace zkoušky,
- podklad spolu s naměřenými daty pro analýzu reálného procesu k potvrzení správné funkčnosti algoritmů dílčích systémů nebo ke zjištění odchylek dynamického chování těchto systémů vůči projektu

Identifikace nedostatků byla dvojí:

- rozpory v dokumentaci vztažené na údaje parametrů. Dokumentace byla dána do souladu se skutečností,
- vliv nastavení parametrů na dynamické chování, kdy odchylky technologických veličin v průběhu přechodových procesů se mohou blížit k limitním hodnotám, při kterých může být aktivováno zapůsobení limitačního, ochranného systému (RCLS, PRPS) nebo lokální ochrany. Byla doporučena opatření v nastavení limitních hodnot. .
- na základě analýzy byly zjištěny nesrovnalosti technologického charakteru např. netěsnost armatury, odpojené měření,.. nebo softwarového charakteru: nevhodná nastavení parametrů, časových konstant,.... Byla doporučena jiná nastavení parametrů regulací a logik v dílčích systémech nebo úprava algoritmů.



Obr 4: Přechodový proces (před analýzou, po analýze)

4. POUŽITÍ STATICKÉ ANALÝZY PRO TESTOVÁNÍ SOFTWARU

Principem statického testování je analýza zdrojového kódu formou "bílá skříňka" bez provedení vlastního výpočtu. Podkladem pro tuto analýzu je projektová dokumentace,

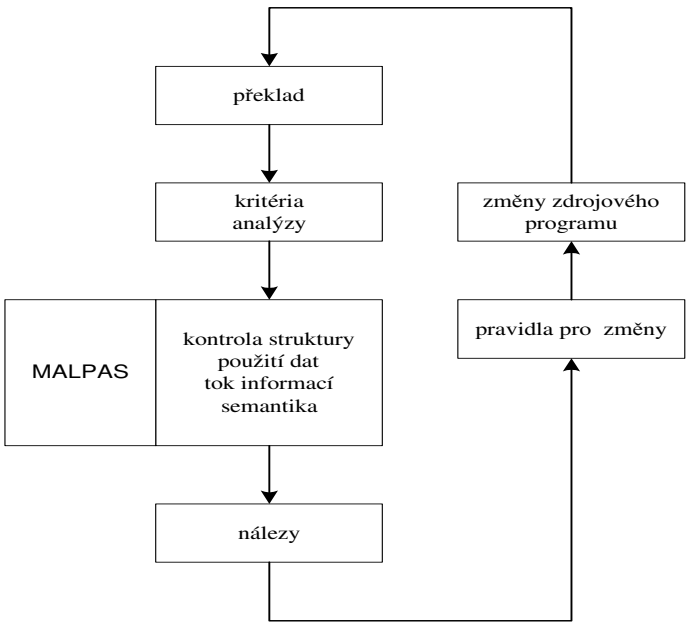
konfigurační data a zdrojový kód. Toto testování bylo aplikováno při nezávislém hodnocení systémů PRPS a DPS (diverzní ochranný systém) na JE Temelín..

Statická analýza zahrnuje:

a) analýzu syntaxe programu, která má dílčí části:
   - analýza toku kontrolující strukturu příslušné procedury a definující všechny vstupní a výstupní proměnné včetně identifikace všech uzavřených smyček s jejich vstupy a výstupy,
   - analýza použitých dat, která kontroluje data (proměnné a formální parametry), použitá v rámci procedury a na základě toho kontroluje, jak jsou vstupní veličiny popsány uvnitř procedury,
   - analýza toku informace identifikující všechny informace, na kterých závisí výstupní proměnné, a poskytující kontrolu, že výstupní proměnné procedur jsou závislé na jejich správných vstupech,

b) analýzu sémanticky. Cílem této analýzy je prokázat souhlas zdrojového kódu s jeho projektovou dokumentací. Analýza má následující výstupy:
   - kontrola, že každý SW modul vykonává funkci podle zadané specifikace,
   - kontrola možnosti přetečení aritmetických výrazů,
   - kontrola typu a rozsahu základních charakteristik parametrů,
   - kontrola, že pouze správné hodnoty jsou připsány k proměnným,
   - kontrola, že základní pointry jsou správně přiřazeny proměnným,
   - kontrola, že proměnné vrací správně svoji hodnotu

Jako SW nástroj byl použit program MALPAS, který umožňuje práci zefektivnit a objektivizovat výstupy analýzy (omezit vliv pracovníka provádějícího analýzy na její výsledky).



Obr 5: Schéma statické analýzy

Hlavní etapy analýzy lze popsat takto:

- překlad zdrojového kódu do MALPAS Intermadiate Language (IL),
- zadání cílů analýzy. Cíle spadají do dvou kategorií: identifikovat integritu programu a identifikovat skryté předpoklady,
- provedení syntaxní analýzy,
- provedení semantické analýzy,
- vytvoření PROSPEC. PROSPEC je IL representace analyzované procedury a je používán při statické analýze procedur vyšších úrovní..
- nálezy-vyhodnocení analýzy. Cílem je ověření shody vytvořeného zdrojového programu s požadavky a návrhem na software. Dále zajištění, že všechny nalezené anomálie softwaru jsou správně zpracovány a vyhodnoceny.
- pravidla s nálezy, představuje jejich kategorizaci a návrh na řešení změny

5. ZÁVĚR

Tvorba softwaru pro systémy na JE zajišťující funkci kontroly a řízení pro iniciaci ochranných zásahů vyžaduje systematický přístup jak v oblasti organizace, tak v oblasti technických aspektů v průběhu celého cyklu životnosti. Velká pozornost je kladena na otázku verifikace a validace softwaru. Vedle automatizovaných nástrojů pro testování softwaru se jako další nástroj zařadil i simulační model, který se doposud používal jako prostředek pro analýzu přechodových procesů nebo jako prostředek pro predikci dynamického chování systému.

6. LITERATURA

[1] Code on the Safety of Nuclear Power Plants: Design; IAEA Safety Series No. 50-C-D/Rev.1, 1988.

[2] Vyhláška č. 214/1977 Sb.Vyhláška SUJB o zabezpečení jakosti při činnostech související s využíváním jaderné energie a činnostech vedoucí k ozařování a o ustanovení kriterií pro zařazení a rozdělení vybraných zařízení do bezpečnostních třídování

[3] IEC 880, Software for computers in the safety systems of nuclear power plant

[4] IEC 61226, Nuclear power plants - Instrumentation and control systems important for safety - Classification

[5] ČSN IEC 61513 Jaderné elektrárny - systémy kontroly a řízení důležité pro bezpečnost - Všeobecné požadavky na systémy

[6] Independent Assessment of I&C Software in Safety System for NPP Temelin , Contract CEZ, 1999

[7] Obnova SKŘ JE Dukovany, Stavba T54440000, Projekt ČEZ, 2000

[8] Z.Piroutek, S.Roubal, J.Rubek: Static Analysis of the Software used in Safety Critical System of the NPP Temelin, CNRA/CSNI Workshop, str. 91, Hluboká n/Vlt, 2001.

[9] S.Roubal, Z.Piroutek, M.Kmínek: Využití simulačních modelů na čs. jaderných elektrárnách, konference CP&HS, str. 138, Zlín, 2006

[10] Analytická a systémová dokumentace AP SIMED, Zpráva I&C Energo, 2000

[11] Program DYTE, Zpráva EGU, 1999