

UPDATES GENERATOR FOR FINANCIAL TRADING SYSTEMS

Ondřej Cienciala

Bachelor Degree Programme (4), FIT BUT

E-mail: xcienc02@stud.fit.vutbr.cz

Supervised by: Milan Dvořák

E-mail: idvorakmilan@fit.vutbr.cz

Abstract: This paper provides a description of design of stock exchange updates generator which is capable of generating constrained-random messages. It can be used for testing software that handles messages from an electronic stock exchange. Technique of coverage-driven verification and constrained-random stimulus generation are discussed. Message generation is based on XML template and because of that the generator can be adjusted for various exchanges.

Keywords: exchange updates generator, NYSE, ISE, functional verification, constrained-random messages

1 ÚVOD

Elektronické obchodování na burzách s sebou nese různé problémy technického typu. Jedním z nich je přenos informací z burzy ke koncovým uživatelům (obchodníkům) a naopak. Děje se tak pomocí real-time datových proudů, ve kterých se přenášejí jednotlivé zprávy aktualizující stav burzy. Software na straně obchodníka zpracovává tyto zprávy a vytváří obraz (model) burzy, na jehož základě může obchodník činit obchodní rozhodnutí (nakoupit nebo prodat). Chyby v těchto systémech mohou způsobit katastrofální finanční ztráty. Například v roce 2012 klíčový tvůrce trhu Knight Capital vytvořil vlivem softwarové chyby během několika hodin ztrátu 440 milionů dolarů [1]. Proto jsou na tento software kladeny vysoké požadavky na bezchybovost a musí být řádně testován.

Má práce se zabývá návrhem konfigurovatelného nástroje pro generování velkého množství náhodných zpráv (vyhovujících burzovním specifikacím), pomocí nichž lze otestovat software vytvářející obraz burzy. Během návrhu jsem se primárně soustředil na burzy NYSE Arca (New York Stock Exchange Arca) [2] a ISE (International Securities Exchange) [3]. Ze zpráv se zaměřuji především na ty, které aktualizují obraz burzy — hloubku trhu (okamžitá nabídka a poptávka na burze na různých úrovních kurzu).

2 ZPRÁVY

Informace zasílané burzou jsou ve zprávě rozděleny do jednotlivých položek. Jedná se o různé identifikátory obchodovaných instrumentů, časové údaje, cenové hladiny, kódy událostí apod. Burzy si definují vlastní protokol a formát zpráv, který uvádějí ve specifikačních dokumentech.

Zprávy, které přichází z burzy, jsou během jejího normálního provozu velmi podobné — množství stavů, do kterých se obchodování během reálné činnosti dostane, je velmi omezené oproti množině všech možných stavů definovaných protokolem burzy. Pro testování je tedy nutné generovat zprávy tak, aby postihly všechny stavy, které protokol umožňuje vytvořit (např. pozastavení obchodování, resetování obrazu pro určitý obchodovaný instrument a podobně). Dále je třeba testovat tvorbu obrazu burzy. Během reálného provozu obvykle nedochází k velkým cenovým fluktuacím a kurz se nepřibližuje k problematickým krajním hodnotám, testování tedy musí pokrýt různé mezní cenové pohyby a aktualizace obrazu.

3 FUNKČNÍ VERIFIKACE

Pro účely testování jsem se rozhodl využít principů funkční verifikace, která se využívá při testování hardwarových systémů. Lze ji však aplikovat i pro generování zpráv. Základem funkční verifikace je generování množiny náhodných testovacích vektorů, které je omezeno jistými podmínkami. Tyto vektory se poté aplikují na testovaný systém a srovnává se reálný a očekávaný výsledek. Snažím se zde zkombinovat techniky verifikace řízené pokrytím a generování náhodných vstupních vektorů [4].

3.1 GENEROVÁNÍ NÁHODNÝCH VSTUPNÍCH VEKTORŮ

Technika generování náhodných vstupních vektorů (*constrained-random stimulus generation*) testuje systém proti velkému množství náhodných vstupů. Korektnost formátu těchto vstupů je zajištěna pomocí omezujících podmínek. To v praxi umožňuje dodržet specifikaci danou protokolem burzy a také otestovat požadovaný rozptyl vstupních hodnot.

3.2 VERIFIKACE ŘÍZENÁ POKRYTÍM

Verifikace řízená pokrytím (*coverage-driven verification*) se snaží řádně prověřit vlastnosti či stavy testovaného systému a vytváří statistiku pokrytí. U systému můžeme sledovat např. pokrytí zdrojového kódu, provedených cest v systému, navštívené stavy konečného automatu apod. Pro účely této práce pokrývám stavy, do kterých se může burza dostat. Všechny tyto stavy jsou definovány v komunikačním protokolu dané burzy — uvedeny obvykle výčtem jako kódy daných událostí. Stavy, které chceme pokrýt, musí být zapsány v XML šabloně (viz níže). Tímto se dá testování omezit pouze na stavy, které chceme prověřit. Během generování zpráv si generátor zaznamenává, které stavy vytvořil a po skončení programu vypíše zmíněnou statistiku pokrytí na základě srovnání vygenerovaných stavů a stavů poskytnutých v XML šabloně.

4 XML ŠABLONA

Problémem při testování systémů pro obchodování na burze je různorodost používaných komunikačních protokolů a přenášených zpráv. Aby byl generátor změny obrazu burzy obecný a mohl být použit pro různé burzy bez jakýchkoliv modifikací v implementaci, byla navržena šablona pro generované zprávy ve formátu XML. Šablona se tvoří na základě protokolu dané burzy a může obsahovat všechny protokolem povolené prvky nebo může být pro účely testování omezena pouze na vybranou množinu.

4.1 POPIS XML ŠABLONY

Šablona definuje požadovaný formát zpráv z burzy a v každé zprávě jednotlivé položky až na úrovni bajtů. Jednotlivé zprávy jsou označeny elementem *message* s atributem *name* pro pojmenování zprávy. Položky jsou uvozeny elementem *field*, opět s atributem *name* pro název položky. Dále je každá položka ve zprávě definovaná atributy *format*, *length*, *type* a *value*. *Format* určuje, zda jsou data v binárním nebo textovém formátu, *length* určuje velikost položky v bajtech v případě, že formát je binární; *type* definuje, zda se bude jednat o konstantu, množinu více hodnot, hodnotový interval, prázdné pole (null) nebo tzv. typ *special*. Typ *special* označuje uživatelem definovanou funkci, protože některé položky ve zprávě (časová známka, sekvenční číslo zprávy) nelze generovat náhodně a nemohou být ani konstantní. Aby tedy implementace nezávisela na konkrétní burze, může uživatel vytvořit vlastní funkci, která se pro generování typu *special* použije, a přidat ji do programu.

Element *value* pak obsahuje samotnou hodnotu položky. V případě typu *interval*, jsou použity další elementy *min-value* a *max-value* určující rozsah intervalu. Intervalů je možné zadat více a lze u nich určit pravděpodobnost, s níž se mají generovat hodnoty z jednotlivých intervalů.

Příklad zjednodušené XML šablony s jedním druhem zprávy s jedinou položkou — změnou ceny:

```

<exchange name="nyse">
  <message name="aggregate-quote-msg">
    <field name="price">
      <format>binary</format><length>4</length>
      <type>interval</type>
      <value probability="0.8">
        <min-value>0</min-value><max-value>10</max-value>
      </value>
      <value probability="0.2">
        <min-value>11</min-value><max-value>90</max-value>
      </value></field></message></exchange>

```

5 GENEROVÁNÍ ZPRÁV

Pro generování náhodných hodnot v rámci zadaného intervalu se používá rovnoměrné (*uniform*) rozložení pravděpodobnosti, které je pro účely testování dostačující. Pokud by to bylo žádoucí, lze implementovat i jiná rozložení (Gaussovo, exponenciální atd.).

Počet generovaných zpráv je možné parametrizovat a omezit na určitý počet zpráv nebo na celkovou výslednou velikost všech zpráv. Generátor také může běžet, dokud nedosáhne stanoveného pokrytí všech stavů, případně cenových hladin. Je možné zkombinovat všechny uvedené volby s tím, že se generátor zastaví při splnění alespoň jedné z nich.

6 ZÁVĚR

V této práci jsem popsal princip fungování generátoru změn obrazu burzy, umožňujícího simulovat tvorbu zpráv z elektronických burz a testovat chování navazujících programů. Generátor je možné použít pro tvorbu zpráv z libovolné burzy díky návrhu XML šablony, kde lze detailně definovat generované zprávy. Pro otestování všech stavů, do kterých se burza může dostat, se zavádí funkční verifikace řízená pokrytím — sleduje se pokrytí těchto stavů a generuje se statistika. Samotný obraz burzy a cenové pohyby jsou testovány proti velkému množství náhodně generovaných hodnot, definovaných v různých intervalech a s různou pravděpodobností generování.

V rámci pokračování práce bych chtěl srovnat pokrytí dosažené vygenerovanými zprávami s reálným záznamem dat z burzy a vyhodnotit výsledky.

REFERENCE

- [1] Knight Capital's \$440m trading loss 'caused by disused software'. The Telegraph [online]. 2012-09-14 [cit. 2014-03-02]. Dostupné z: <http://www.telegraph.co.uk/finance/newsbysector/banksandfinance/9475292/Knight-Capitals-440m-trading-loss-caused-by-disused-software.html>
- [2] NYSE Euronext: NYSE ArcaBook for Options Client Specification. 2014-02-13. Dostupné z: <http://www.nyxdata.com/doc/4416>
- [3] International Securities Exchange, LLC: Market Data Interface (MDI) Programming Manual. 2011-10-27. Dostupné z: <https://members.ise.com/>
- [4] Šimková, M., Lengál, O., Kajan, M.: HAVEN: An Open Framework for FPGA-Accelerated Functional Verification of Hardware. Brno, FIT VUT, 2011.