

Uporaba domensko specifičnega jezika pri merjenju časa na športnih tekmovanjih

Iztok Fister ml.¹, Iztok Fister¹

¹Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, Smetanova 17, 2000 Maribor
E-pošta: iztok.fister@uni-mb.si

Abstract

A manually time measuring on mass sports competitions cannot be imagined today because of long duration and unreliability. On the other hand, automatic timing devices based on RFID technology become cheaper. However, these devices cannot operate stand-alone because to work efficiently they need a computer timing system capable to process the incoming events, encoding results, assigning these to the suitable competitor, sorting these according to the achieved time and print-out the list of results. In this paper the domain specific language is defined that enables the control of agents by writing the events in a database.

1 Uvod

Nekoč so bili za merjenje rezultatov na športnih prireditvah zadolženi časomerilci, ki so čas merili ročno. Čas, ki ga je pokazala štoparica, so povezali s startno številko tekmovalcev in te razvrstili glede na dosežen rezultat ter kategorijo. S pojavitvijo tehnologije RFID (angl. Radio Frequency Identification), se je merilna tehnologija pocenila (ChampionChip [3], RFID Race Timing Systems [4]) in postala dostopna tudi širšemu krogu uporabnikov (npr. športnim društvom, prirediteljem športnih tekmovanj, ipd.), hkrati pa so postala konkurenca monopolnim podjetjem (Timing Ljubljana [5]) pri spremljanju rezultatov na manjših športnih tekmovanjih.

Za spremljanje rezultatov na športnih tekmovanjih poleg merilne tehnologije potrebujemo računalniški sistem, ki omogoča fleksibilnost (spremljanje različnih športnih tekmovanj s poljubnim številom merilnih naprav in merilnih mest), sprotno beleženje časov in izpis rezultatov, zanesljivost ter varnost. Fleksibilnost sistema povečamo z uporabo domensko specifičnega jezika (angl. domain specific language, krajše DSL).

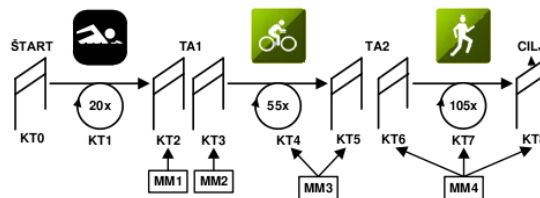
DSL [2] so prirojene aplikacijski domeni in imajo določene prednosti pred splošno namenskimi jeziki v specifični domeni. Te prednosti se kažejo predvsem v večji izrazni moči in s tem večji produktivnosti, lažji uporabi (tudi za domenske strokovnjake, ki niso programerji), enostavnejši verifikaciji in optimizaciji.

2 Merjenje časa na športnih tekmovanjih

V praksi lahko merimo čas na športnih tekmovanjih ročno (*klasična ali računalniška štoparica*) ali avtomatično (*merilna naprava*). Merilna naprava lahko podpira več *merilnih mest* (MM). Ta merilna mesta so dejansko gumirane blazine, ki delujejo kot elektromagnetno antensko polje. Dogodek sproži tekmovalac pri prehodu preko blazine s pomočjo pasivnega čipa, na katerem je zapisana identifikacijska številka. Tekmovalno progo običajno razdelimo na več *kontrolnih točk* (KT), kjer želi organizator zasledovati izmerjeni čas. Ta je lahko *vmesni* ali *končni*. Lokacija kontrolne točke je odvisna od vrste tekmovanja in od konfiguracije proge, na kateri poteka tekmovanje.

2.1 Primer: merjenje časa na triatlonih

Poseben pristop zahtevajo triatlonska tekmovanja, kjer imamo na enem tekmovanju opravka s tremi disciplinami (plavanjem, kolesarjenjem in tekom). Temu problemu se v članku podrobneje posvečamo. Merjenje časa na triatlonskih tekmovanjih delimo v devet kontrolnih točk (slika 1). Na sliki 1 imamo opravka s progo za dvojni triatlon (7,6 kilometrov plavanja, 360 kilometrov kolesarjenja in 84 kilometrov teka), kjer je proga za plavanje dolga 380 metrov (20 krogov), proga za kolesarjenje 3,4 kilometre (105 krogov) in proga za tek 1,5 kilometra (55 krogov).



Slika 1: Definicija kontrolnih točk pri triatlonu.

Skupni čas triatlonskega tekmovanja je sestavljen iz petih končnih (SWIM (KT2), TA1 (KT3), BIKE (KT5), TA2 (KT6) in teka (KT8)) in treh vmesnih časov (KT1, KT4 in KT7). Pri vmesnih časih merimo število krogov ROUND_x in doseženi rezultat INTER_x, kjer $x = 1 \dots 3$ označuje posamezno disciplino.

Predpostavimo, da imamo za merjenje tekmovanja

Program 1 Definicija agentov.

```
1: AGENTS {
2:   {1,MANUAL,“/home/DC2/res.ets”}
3:   {2,AUTO,“192.168.225.100/UDP/9999”} }
```

prikazanega na sliki 1 na voljo merilno napravo z dvema merilnima mestoma (MM3 in MM4) in da tekma poteka na eni lokaciji. V tem primeru lahko zadnji prehod čez MM3 pomeni čas KT5, prvi prehod čez MM4 čas KT6 in zadnji prehod čez MM4 končni dosežek (KT8). Merilni mesti MM1 in MM2 predstavljata ročno merjenje časa. S pravilno postavljenimi kontrolnimi točkami in merilnim sistemom lahko število kontrolnih točk zmanjšamo za dve. Z eno merilno napravo tako izmerimo 162 dogodkov na tekmovalca (oz. 87,6%). Če k temu dodamo še, da merilna tehnologija za merjenje plavanja še ni na voljo, oz. je ta še zelo draga, in zato plavanje merijo sodniki ročno, na tekmi avtomatsko izmerimo kar 98% vseh dogodkov.

3 Razvoj DSL EasyTiming

Za opis merjenja na sliki 1 razvijemo DSL EasyTiming, s katerim na podlagi zapisanih pravil krmilimo *agenta*, ki zapisuje dogodke iz merilnih mest v podatkovno bazo. Z njim lahko opišemo različna merjenja, pri kompleksnejših pa lahko z manj merilnimi mesti pokrijemo več kontrolnih točk. Sestavljen je iz definicij kontrolnih točk, agentov, merilnih mest in pravil. Kontrolne točke predstavljajo delni rezultati tekmovalca shranjenih v podatkovni bazi. Primer opisa agentov je prikazan v programu 1, kjer agent za ročno merjenje (št. 1) shranjuje rezultate v mapi */home/DC2* in agent za avtomatsko merjenje (št. 2) dobiva podatke od naprave z naslovom IP 192.168.225.100 prek protokola *UDP* in vrat 9999.

Merilna mesta v DSL EasyTiming opisujemo s pravili prikazanimi v programu 2. Vsako merilno mesto je označeno s svojo identifikacijsko številko in povezano z ustreznim agentom. Pravila definiramo v obliki $\text{PREDIKAT} ::= \text{OPERATION}_i$, kjer PREDIKAT pomeni pogoj, ki ga testiramo in OPERATION operacijo, ki se izvede, če predikat vrne vrednost pravilno (oz. TRUE). Predikat lahko vsebuje samo vrednost TRUE, kar pomeni, da se operacija izvede brezpogojno.

Za DSL EasyTiming smo razvili sintaktični analizator [1] v jeziku C/C++. Generator jezika generira kodo za navidezni računalnik, ki se izvaja za vsako posamezno merilno mesto posebej. Arhitektura navideznega računalnika je zelo preprosta, saj sestoji iz programskega in podatkovnega segmenta ter programskega števca. Programske instrukcije definiramo kot trojico $\langle op, p1, p2 \rangle$, kjer *op* predstavlja operacijsko kodo, *p1* in *p2* pa parametra (spremenljivka ali konstanta). Nabor instrukcij sestoji iz šestih instrukcij: instrukciji EQ (enako) in NEQ (neenako) uporabljamo pri predikatih, UPD (shrani doseženi čas) in DEC (zmanjšaj vrednost spremenljivke za 1) pri operacijah, STOP (zaustavi) in NOP (operacija nič) pa sta krmilni instrukciji brez parametrov.

Program 2 Definicija merilnih mest.

```
1: MM[1] ::= AGNT[1] {
2:   { (TRUE) ::= UPD $SWIM }
3:   { (TRUE) ::= DEC $ROUND_1 } }
4: MM[2] ::= AGNT[1] {
5:   { (TRUE) ::= UPD $TRANS_1 } }
6: MM[3] ::= AGNT[2] {
7:   { (TRUE) ::= UPD $INTER_2 }
8:   { (TRUE) ::= DEC $ROUND_2 }
9:   { (ROUND_2 == 0) ::= UPD $BIKE } }
10: MM[4] ::= AGNT[2] {
11:  { (TRUE) ::= UPD $INTER_3 }
12:  { (ROUND_3 == 55) ::= UPD $TRANS_2 } }
13:  { (TRUE) ::= DEC $ROUND_3 }
14:  { (ROUND_3 == 0) ::= UPD $RUN } }
```

4 Delovanje agenta

Delovanje agenta, ki ga krmilimo s pomočjo DSL EasyTiming, razbijemo v naslednje faze: rekonstrukcija dogodka, čitanje rezultatov, uvažanje prečitanih rezultatov, interpretacija kode in zapis spremenjenih rezultatov v podatkovno bazo. Pri rekonstrukciji dogodka ugotovimo, čigav je dogodek (identifikacijska številka čipa ali štartna številka) in na katerem merilnem mestu je nastal. Trenutne rezultate identificiranega tekmovalca preberemo iz podatkovne baze in jih naložimo v podatkovni segment navideznega računalnika. Interpretacija programske kode poteka tako, da programski števec postavimo na nič in izvajamo posamezne instrukcije zaporedoma, dokler ne pridemo do instrukcije STOP. Programsko kodo naložimo v programski segment navideznega računalnika ob zagonu agenta iz podatkovne baze.

5 Zaključek

Pri razvoju univerzalne programske opreme za merjenje časa na športnih tekmovanjih se pogosto srečamo s problemom fleksibilnosti merilnih naprav. V ta namen smo razvili DSL, s katerim krmilimo agenta pri zapisu rezultatov posameznih tekmovalcev v podatkovno bazo. Uporaba omenjenega DSL v praksi je pokazala, da za merjenje časa na manjših športnih tekmovanjih ni potrebno več najemati specializirana in praviloma zelo draga podjetja.

6 *

Literatura

- [1] N. Wirth: Algorithms + Data Structures = Programs, 1978, Prentice Hall PTR, Upper Saddle River, NJ, USA.
- [2] M. Mernik, J. Heering, A. Sloane, "When and how to develop domain-specific languages", ACM computing surveys, vol. 37, no. 4, pp. 316-344, 2005.
- [3] ChampionChip2010, <http://www.championchip.com>
- [4] RFID2010, <http://www.rfidtiming.com/>
- [5] Timing2010, <http://www.timingljubljana.si/>