

UNIVERZITA PARDUBICE
DOPRAVNÍ FAKULTA JANA PERNERA
KATEDRA INFORMATIKY V DOPRAVĚ

INFORMAČNÍ SYSTÉM OKRUHOVÝCH ZÁVODŮ

DIPLOMOVÁ PRÁCE

AUTOR PRÁCE: Bc. Radek Paclt

VEDOUCÍ PRÁCE: Ing. Karel Greiner

2006

UNIVERSITY OF PARDUBICE
JAN PERNER TRANSPORT FACULTY
DEPARTMENT OF INFORMATICS IN TRANSPORT

INFORMATION SYSTEM OF CIRCLE RACES

THESIS

AUTHOR: Bc. Radek Paclt

SUPERVISOR: Ing. Karel Greiner

2006

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 30. 5. 2006

Radek Paclt

Poděkování:

Děkuji všem, kteří mi poskytli potřebné informace a odborné rady nebo mi jiným způsobem pomohli při zpracování této práce a především děkuji svému vedoucímu diplomové práce Ing. Karlu Greinerovi.

ABSTRAKT

Tato diplomová práce je zaměřena na tvorbu informačního systému pro měření času okruhových závodů. Práce vychází z analýzy současného stavu měření se zaměřením na uspokojení požadavků zákazníků. Dále jsou tyto požadavky v práci podrobněji analyzovány. Z vytvořené analýzy se koncipuje návrhový model informačního systému, který je předlohou pro implementaci. V závěru práce byl daný systém nasazen a otestován v praktických podmínkách okruhového závodu.

ABSTRACT

The aim of the thesis is to develop an information system for measuring the time of circle races. The work is based on the analysis of current state of measuring focusing on the satisfaction of customers demands. The requirements have been analyzed in detail. The analysis forms the base for a draft information system that serves as a model for the implementation. In conclusion, the given system was practically launched and tested in the conditions of a circle race.

OBSAH

Úvod.....	10
1 Charakteristika měření okruhových závodů	12
1.1 Okruhové závody.....	12
1.2 Princip měření času	12
1.2.1 Prezentace závodníků a přejímka strojů	12
1.2.2 Příprava a ověření funkčnosti měřícího zařízení	13
1.2.3 Měření jízdy.....	14
1.2.4 Vyhodnocení výsledků závodu.....	15
1.3 Komponenty pro měření času a jejich umístění	15
1.3.1 Seznam komponent	16
1.4 Komerční programy a systémy pro měření času	17
1.4.1 Orbits systém	18
1.4.2 Alfano	20
1.4.3 KART – DATA systém	21
1.4.4 InformSys for PDA.....	22
1.4.5 Ostatní měřící systémy	24
1.4.6 Systémy používané v České republice	25
2 Modelování v UML.....	26
2.1 Co je to UML.....	26
2.2 Struktura jazyka UML	26
2.2.1 Stavební bloky jazyka UML.....	27
2.2.2 Obecná mechanika jazyka UML	27
2.2.3 Architektura	27
2.3 Unified Process.....	28
2.3.1 Konkrétní aplikace metodiky UP v novém projektu	28
2.3.2 Axiomy metodiky UP	29
2.3.3 Metodika UP je založena na iterativním a přírůstkovém procesu	29
2.3.4 Pracovní postupy iterace.....	29
2.4 Využití UML v modelování informačního systému okruhových závodů	30
3 Požadavky	31
3.1 Pracovní postup požadavků	31
3.2 Informační systém okruhových závodů.....	31
3.2.1 Obecný popis systému:.....	31
3.2.2 Požadavky na informační systém	32

4	Analýza	34
4.1	Pracovní postup analýzy	34
4.1.1	Modelování případů užití	34
4.1.2	Slovníček pojmů	35
4.1.3	Sekvenční diagramy	35
4.1.4	Analýza v UML	36
4.2	Případy užití systému	36
4.2.1	Přihlášení	38
4.2.2	Editace krabičky	39
4.2.3	Editace závodníků	41
4.2.4	Definice složek a jízd	42
4.2.5	Měření jízdy	43
4.2.6	Vyhodnocení jízdy	43
4.2.7	Import a export dat	44
4.3	Slovníček pojmů	45
4.4	Zaměření pracovního postupu analýzy	45
5	Návrh	46
5.1	Pracovní postup návrhu	46
5.1.1	Návrhové třídy	46
5.1.2	Diagram aktivit	47
5.1.3	Stavový diagram	47
5.1.4	Datový model	48
5.2	Návrhové třídy	48
5.3	Diagramy aktivit	49
5.4	Stavové diagramy	53
5.4.1	Základní stavy systému	53
5.4.2	Stavový diagram „neměří se“	54
5.4.3	Stavový diagram „měří se“	55
5.4.4	Stavový diagram „data na PDA“	55
5.4.5	Stavový diagram „nastavení aplikace“	56
5.5	Datový model	58
6	Implementace	60
6.1	Pracovní postup implementace	60
6.2	Implementace systému	60
6.2.1	TimeKeeper	61
6.2.2	TimeKeeper pro PDA	61
6.2.3	Ping status	61
6.2.4	Simulátor dekodéru	62
7	Nasazení	63
7.1	Pracovní postup nasazení	63
7.1.1	Komponenta	63
7.1.2	Diagram nasazení	63
7.2	Hardware	63
7.2.1	Diagram nasazení z pohledu hardware	67
7.3	Software	67
7.3.1	Komponenty	68
7.3.2	Diagram komponent	70

8	Testování	71
8.1	Pracovní postup testování.....	71
8.1.1	Mistrovství České republiky Česká Lípa.....	71
8.1.2	Hobby Cup Vysoké Mýto.....	73
8.2	Závěr pracovního postupu testování.....	74
	Závěr	75
	Seznam literatury	77
	Seznam tabulek.....	78
	Seznam obrázků.....	79
	Seznam příloh	80

Úvod

Každým rokem se v České republice pod záštitou Autoklubu České republiky pořádá mnoho okruhových závodů. Tyto závody se dají rozdělit dle druhů závodních strojů. Mezi nejznámější druhy závodů se dají zařadit závody trucků, motocyklů, automobilů a v neposlední řadě také závody motokár.

Při těchto závodech se používá několik druhů měření časů. Prvním, v dnešní době již minoritním, způsobem měření je používání fotobuňky, která za pomoci své optické závory zaznamenává každý průjezd cílovou páskou. Druhým způsobem, v dnešní době majoritním, je měření za pomoci transpondérů a dekodéru. Tento způsob je založen na principu, kdy každý závodník má na svém závodním stroji připevněno zařízení transpondér. Dále je na cílové pásce v dráze zabudovaná magnetická smyčka, která je napojena na zařízení dekodér. Při každém průjezdu jezdce s jeho osobním transpondérem přes tuto smyčku, je o tomto průjezdu informován dekodér.

Měření za pomoci transpondérů a dekodéru je podmíněno používáním informačního systému, který data načítá z dekodéru a provádí další zpracování do podoby konečného vyhodnocení každé jízdy a potažmo celého závodu.

V této diplomové práci jsem se zaměřil na závody motokár a informační systém pro měření času těchto okruhových závodů.

V současné době je jediným oficiálním a doporučovaným systémem pro měření času systém Orbits od společnosti AMB. Tato společnost je celosvětově známou a uznávanou ve světě hardwaru a softwaru pro okruhové závody. S postupem času a s délkou používání tohoto systému se však vyskytlo několik problémů. Tyto problémy souvisely s univerzálností systému. Tento systém nebyl primárně určen pro měření v České republice a z tohoto faktu plyne jeho nepřizpůsobení českým požadavkům.

Jedním z požadavků na systém je zasílání online výsledků o probíhající jízdě na mobilní zařízení. Pro stáj každého jezdce je velice důležité znát přesné hodnoty časů jednotlivých kol svého jezdce, ale také informace o časech ostatních jezdců. Systém Orbits nabízí zasílání online výsledků na notebook. Toto mobilní zařízení je sice přenosné, ale při operování v terénu pro uživatele velice nepohodlné.

Z tohoto důvodu vyvstal požadavek na systém, který bude zpracovávat data z dekodéru a zasílat na mobilní zařízení typu PocketPC, které je velice jednoduše přenositelné a každý člen týmu může toto zařízení po okruhu přenášet a být stále informován o stavu dané jízdy a hodnotách časů svého a ostatních jezdců.

V této diplomové práci jsem se zaměřil na pokus uspokojit daný požadavek. Snahou bude analyzovat informační systém, navrhnout a následně tento systém implementovat.

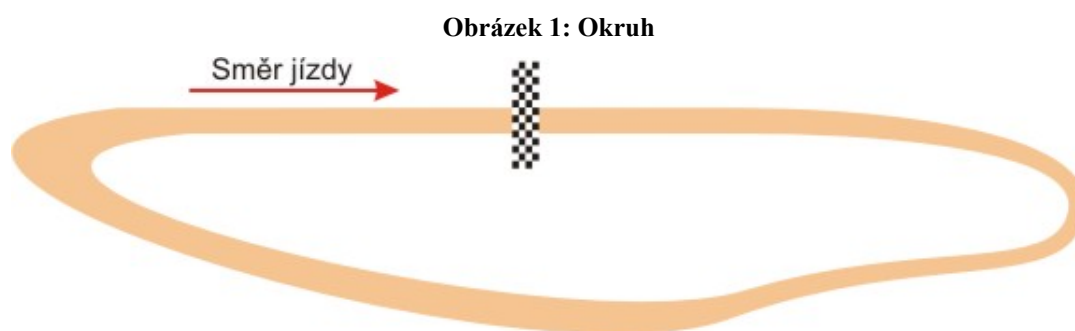
Vytvořit tento systém dle stanovených požadavků je hlavním cílem této diplomové práce.

1 CHARAKTERISTIKA MĚŘENÍ OKRUHOVÝCH ZÁVODŮ

1.1 Okruhové závody

U mnoha druhů sportů se setkáme s pojmem okruhové závody. Okruhový závod je takový typ závodu, kde start a cíl je v jednom a tom samém místě. Pokud se podíváme na obrázek č. 1, můžeme vidět základní nákres okruhu.

Tato diplomová práce se ve svém výkladu omezuje na okruhové motoristické závody, tedy přesněji na závody kartingu¹, ale obecně by se dalo poznamenat, že principy informačního systému okruhových závodů bychom mohli aplikovat na kterýkoliv druh sportu.



Zdroj: Vlastní tvorba

1.2 Princip měření času

Celý systém měření času musíme pojmut v širším pojetí pořádaného závodu. Praktickému měření času, vyhodnocování a publikování výsledků předchází mnoho operací. V této souvislosti si konkrétněji popíšeme chronologický průběh celého závodu. Omezíme se však pouze na operace, které přímo nebo nepřímo souvisejí s měřením času.

1.2.1 Prezentace závodníků a převjímká strojů

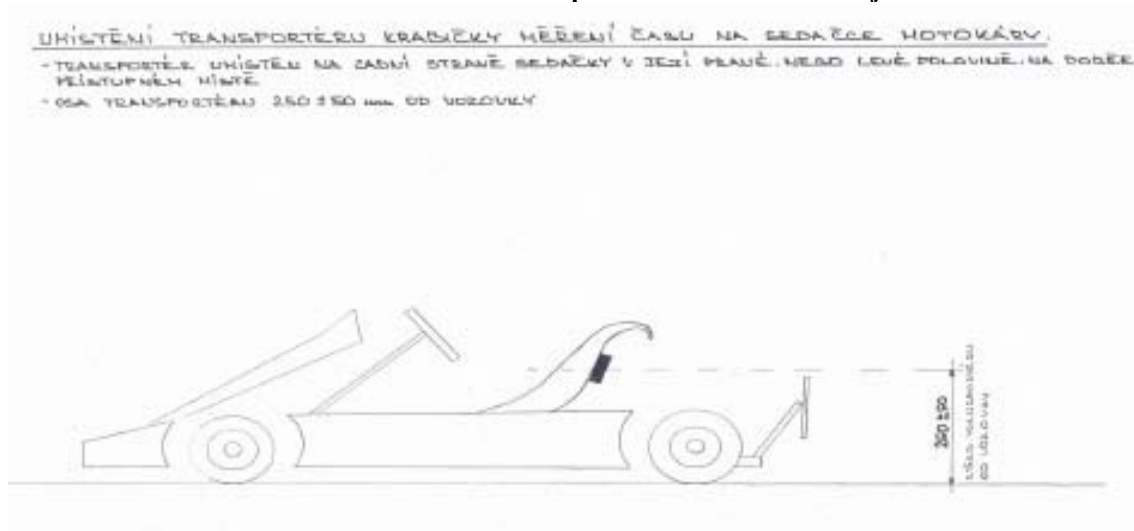
První základní a důležitou částí celého závodu je prezentace závodníků a převjímká strojů. Tato operace obnáší mnoho formálností. Počínáje přihláškou závodníka do závodu a kontrolou stroje technickým komisařem.

Nejdůležitější operací v rámci budoucího měření časů je, že každý závodník dostane svůj vlastní transpondér. Tuto krabičku si musí závodník umístit na svůj stroj. Transpondér je

¹ Karting nebo také jinými slovy závod motokár

nepřenosný a umístění je dáno technickým řádem. Umístění krabičky je na obrázku č. 2. Tato striktní pravidla jsou z důvodů možnosti objektivního změření časů každého závodníka.

Obrázek 2: Umístění transpondéru na závodním stroji



Zdroj: Ročenka kartingu 2006

Z předcházejícího odstavce se dá odvodit, že pro časoměřiče a celkově pro měření času je závodník reprezentován pouze svým vlastním transpondérem.

Závodník zodpovídá za správné umístění, funkčnost a ztrátu transpondéru. V této souvislosti se nejedná o obavy z odcizení zařízení. Závodník je však na trati reprezentován pouze svým transpondérem, a tedy pokud by došlo k poškození nebo ztrátě z důvodu příkladem kolize na trati, nebude závodník pro systém vidět a je zde zkomplikováno správné změření takového závodníka.

Jelikož je tato část závodu velice časově náročná, je započata den před závodem a dokončována v den závodu. Veškeré operace musí být dokončeny před započítáním první měřené jízdy.

1.2.2 Příprava a ověření funkčnosti měřícího zařízení

Tato část závodu jde paralelně s prezentací závodníků a přejímkou strojů. Můžeme zde mluvit o dvou hlavních druzích činností.

Jeden okruh činností se zabývá instalováním měřícího zařízení a jeho odzkoušením. Pro dostatečný prostor k řešení problému je instalace prováděna den před měřením první jízdy.

Ve druhém okruhu činností se načítají informace z prezentace závodníků. Programově se závodník přihlásí do závodu a přiřadí se mu jeho transpondér, případně transpondéry. Je zde totiž možnost, že závodník bude startovat ve dvou různých kategoriích.

1.2.3 Měření jízdy

V této fázi již dochází k praktickému měření času. V místě startu a cíle je příčně v asfaltu zabudována měřicí smyčka. Závodníkův průjezd je za pomoci transpondéru detekován touto smyčkou, která signál posílá neprodleně do dekódovacího zařízení. Toto zařízení signál zpracuje a dále distribuuje, již v podobě strukturované informace do PC, na kterém běží měřicí program. Tento program data následně zpracovává a ukládá. V průběhu závodu je program schopen již upravená data dále distribuovat pro jejich online zobrazení účastníky pořádaného závodu. Po ukončení jízdy jsou data exportována, archivována a tisknuta.

Pokud jste si všimli, stále zde zaznívá pojem jízda, ale nikde nefiguroval pojem závod. Tyto dva pojmy jsou lehce zaměnitelné, ale každý má vlastní specifický význam.

Závod

Závod je definován jako kolekce nezávislých jízd. Každý závodník se účastní jednotlivých jízd, které se dají dále rozdělit na dva druhy.

Měřená jízda

Přesné označení této jízdy je měřená tréninková jízda. Každý závod obsahuje minimálně dvě měřené jízdy. U těchto jízd je pro závodníky nejdůležitější jejich nejrychlejší dosažené kolo. Při této jízdě neexistuje žádný hromadný start. Jízda je definována svojí časovou délkou. Závodníci se mohou dle uvážení vydávat na trať a pokoušet se dosáhnout svého nejlepšího času. Výsledkem takové jízdy je pak pořadí závodníků podle nejrychlejšího dosaženého kola. Jelikož je to však měřená tréninková jízda, výsledky této jízdy, popřípadě těchto jízd se používají pro stanovení pořadí na startovním roštu bodované jízdy. Závodník, který v rámci této jízdy zajede do depa, již nemůže vyjet zpět na trať.

Bodovaná jízda

Obvykle se jedou 3 – 4 bodované jízdy. Tyto jízdy již probíhají s hromadným startem na světelnou signalizaci a již přímo souvisí s výsledkem závodu. U této jízdy je pro závodníky nejdůležitější čas a pořadí v cíli dané bodované jízdy. Za každou bodovanou jízdu dostane

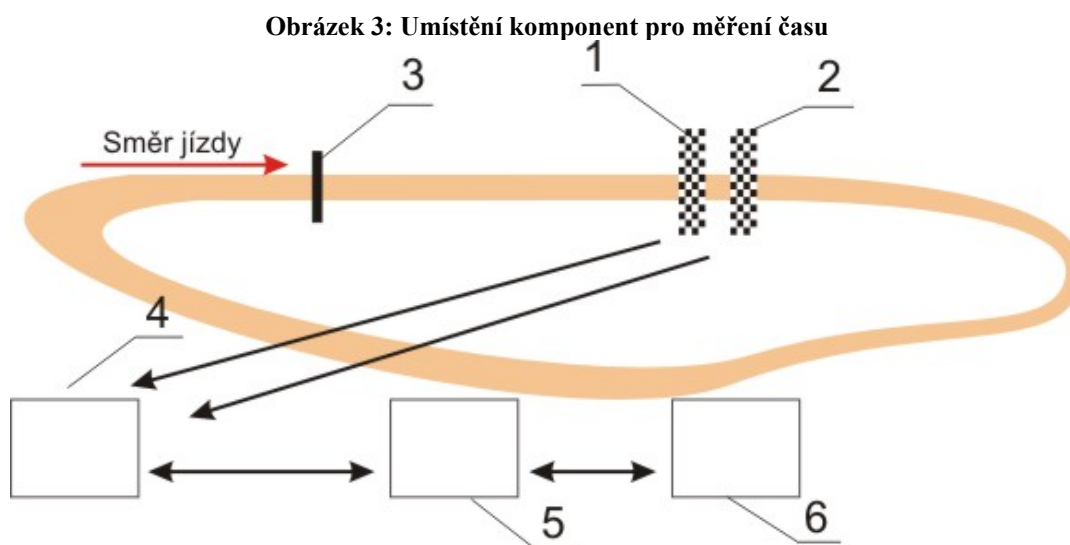
závodník body. Tyto body se na konci závodu u jednotlivého závodníka sečtou a výsledkem je absolutní pořadí závodníků v daném závodě. Vítězem se stane závodník s nejvyšším počtem dosažených bodů z bodovaných jízd.

1.2.4 Vyhodnocení výsledků závodu

Po ukončení poslední bodované jízdy se provádí vyhodnocení celého závodu. Stanovuje se výsledek závodu. Dále se data ukládají a exportují ve speciálním formátu na oficiální stránky daného poháru závodů, kde jsou následně návštěvníkům ke zpětnému nahlédnutí.

1.3 Komponenty pro měření času a jejich umístění

Obecně každý systém pro měření času na okruhových závodech se skládá z více komponent (částí). Tyto jednotlivé komponenty jsou níže vyobrazeny na obrázku č. 3 a následně i systematicky popsány v podkapitole „Seznam komponent“. Každá komponenta je specifikována účelem a umístěním.



Zdroj: Vlastní tvorba

1.3.1 Seznam komponent

V této kapitole se seznámíme s jednotlivými komponentami pro měření okruhových závodů.

- **Hlavní cílová smyčka [1]**

Drát ve formě smyčky, který je pevně zapuštěn do asfaltového povrchu závodní dráhy nebo je jiným pevným způsobem ukotven na cílové pásce. Předává signál dekódovacímu zařízení o průjezdech jednotlivých transpondérů. Cílová páska je obvykle umístěna uprostřed cílové rovinky.

- **Záložní cílová smyčka [2]**

Existují dva druhy záložní cílové smyčky. Jde o technologicky rozdílné pojetí záložního snímání průjezdů závodníků. V prvním případě je záložní smyčka obdobou hlavní smyčky. Ve druhém případě se obecně jedná o jiný způsob měření, kdy se nejčastěji používá optické závory. Optická závora je mobilní zařízení, které za pomoci fotobuněk snímá dráhu a při průjezdu závodníka vysílá signál do dekódovacího zařízení. Tato záložní smyčka je umístěna za hlavní smyčkou ve směru jízdy závodníka. Rozmezí mezi záložní a hlavní smyčkou je otázkou parametrů každého závodiště. V případě mobilní optické závory je to otázkou nastavení pozice fotobuňky. Tato pozice je cca. 4 – 8 m.

- **Alfano smyčka [3]**

Alfano smyčka je velice specifickým zařízením na závodní dráze. Tato smyčka je použitelná pouze pro měřicí systém stejné značky. Měřicí systém Alfano je podrobněji popsán v kapitole „Komerční programy a systémy pro měření času“.

- **Dekódovací zařízení [4]**

Toto zařízení funguje jako cílový bod pro signály z hlavní a záložní smyčky. Dekódovací zařízení, jinými slovy dekodér², zpracovává přijaté signály do podoby strukturované informace, kterou následně odesílá přes komunikační port na PC, kde se data upravují a dále distribuují. Dekodér je velice komplexní zařízení a jeho další specifikace a podrobné informace jsou uvedeny v příloze č.1 „Komponenty pro měření času a jejich technická specifikace“.

² Více než dekódovací zařízení se vžil pojem „dekodér“

- **System pro zpracování dat [5]**

Upravují se zde data načtená z dekodéru. Způsob zpracování je závislý na implementaci. Nejpoužívanější systémy pro měření času jsou naznačeny v kapitole „Komerční programy a systémy pro měření času“.

- **Prezentace online výsledků jízdy [6]**

V závislosti na systému pro zpracování dat se používají programy pro zobrazování online výsledků jízd. Tyto programy jsou koncipovány jako klienti, kteří načítají data ze systému pro zpracování dat viz. „System pro zpracování dat“, odrážka výše.

- **Transpondér³**

Mobilní zařízení, které je umístěno na závodním stroji každého jezdce. Má vlastní zdroj energie. Každý transpondér má uloženo své unikátní číslo. Umístění transpondéru na závodním stroji je naznačeno na obrázku č.2.

Každé zařízení, které se používá pro měření času, musí být odsouhlaseno federací daného sportu. Každoročně jsou zařízení kontrolována a testována. Nejdůležitějšími parametry pro schválení daného zařízení jsou důraz na funkčnost, přesnost a spolehlivost bez ohledu na stav počasí.

Seznam s úplnou specifikací a vyobrazením je uveden v příloze č.1 „Komponenty pro měření času a jejich technická specifikace“.

1.4 Komerční programy a systémy pro měření času

V současné době je na trhu k dispozici několik produktů, které se specializují na oblast informačních systémů pro okruhové závody. Tyto programy jsou obecně nezávislé na druhu závodu, tedy přesněji na sportovním odvětví. Jedinou a nutnou podmínkou je okruh, na kterém se daný závod měří. Tedy jinými slovy je to závod, kde start a cíl je ve stejném místě.

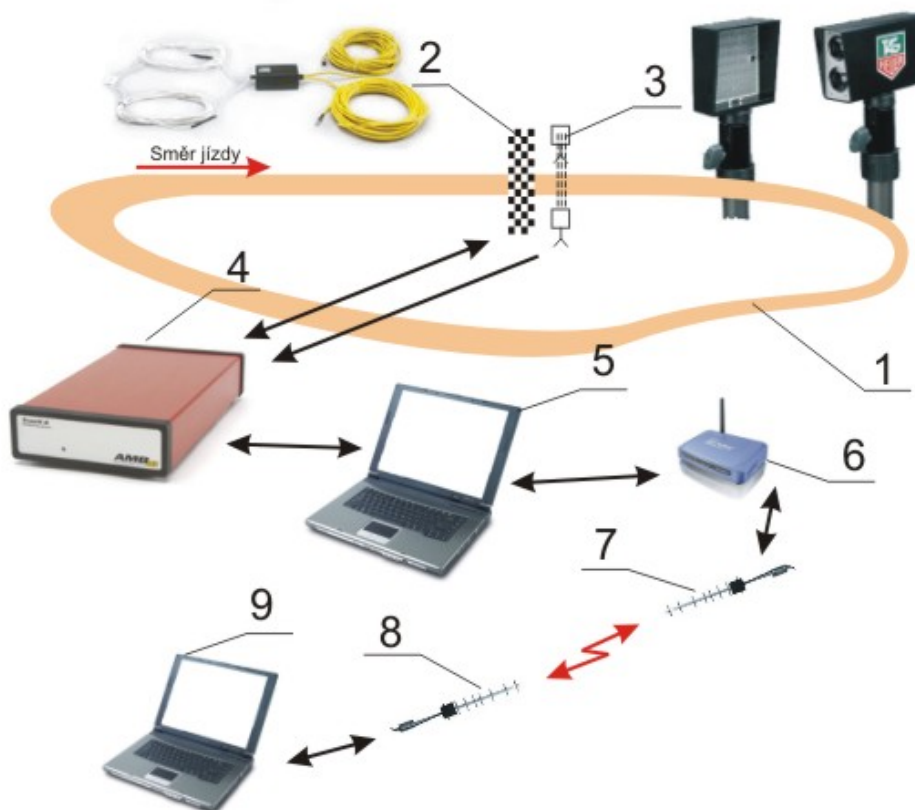
³ Můžete se setkat s označením „krabička“. Pojem transpondér je totožný s pojmem „krabička“

1.4.1 Orbits systém

Celý systém se skládá ze dvou základních modulů programu a jednoho doplňkového programu pro zobrazování online výsledků během jízdy. Prvním modulem je lokální databázový server, který udržuje data o jízdách, závodnících, transpondérech a dalších důležitých informacích k závodě. Druhým modulem je měřicí software, který načítá data z dekodéru, zpracovává je a ukládá do lokální databáze. Oba moduly jsou úzce propojeny. Třetím doplňujícím programem je R-monitor, který nemá žádný vliv na běh základních dvou modulů. Tento program běží mimo oba moduly. R-monitor se pouze připojí na měřicí modul přes jeho IP adresu a specifický port, kde naslouchá. Měřicí modul průběžně vyhodnocuje výsledky, které posílá přes port do R-monitoru, který již podle názvu provádí činnost zobrazování výsledku na monitoru uživatele.

Na obrázku č. 4 je náčrt komponent celého systému.

Obrázek 4: Orbits systém



Zdroj: Vlastní tvorba

Popis obrázku: 1 – závodní okruh , 2 – hlavní smyčka, 3 – záložní smyčka, 4 – dekodér, 5 – PC s programem Orbits, 6 – AccesPoint, 7 – vysílací směrová anténa, 8 – přijímací směrová anténa, 9 – PC s programem R-monitor.

Použité komponenty

Komponenty jsou použity od firmy AMB, která patří mezi celosvětovou jedničkou v měřící technice. Pro přenos výsledků do R-monitoru se využívá WiFi bezdrátové sítě.

Hardware požadavky

- PC s procesorem Intel Pentium III (min. 600 Mhz),
- nejméně jeden volný port COM,
- Windows XP Pro (preferovaný), XP Home, Windows 2000,
- 128 MB vnitřní paměť,
- přibližně 50 MB místa na disku,
- CD-ROM mechanika.

Hodnocení systému

Výhody:

- robustní spolehlivý systém,
- využívá vlastní databázový engine,
- jednoduché a intuitivní ovládání programu,
- online výsledky prostřednictvím programu R-monitor, který je součástí systému Orbits. R-monitor má možnost zobrazování různých výsledků, řazení záznamů podle různých kritérií.

Nevýhody:

- program běží pouze na PC/Notebook – nevhodné mobilní zařízení,
- nemožnost ukládat záznam o proběhlé jízdě,
- při přihlášení programu v průběhu jízdy – data jsou zobrazována od přihlášení, nikoliv od začátku jízdy – pokud se program přihlásí v pátém kole a jezdec zajel nejrychlejší kolo ve třetím kole, potom tato informace při zobrazování online výsledků chybí,
- nutnost zakoupit celý systém Orbits – samostatný program R-monitor pouze zobrazuje data zasílaná Orbitsem,
- po pozdním přihlášení je nutné manuálně obnovit zobrazení dat v R-monitoru.

1.4.2 Alfano

Tento systém je, dalo by se říci, nesrovnatelný s konkurenčními produkty. Nejedná se zde o žádnou formu hromadného vyhodnocování výsledků závodu. Systém je specifický tím, že podává výsledky pouze každému závodníkovi zvlášť. Celý systém je namontován přímo na stroji závodníka. Je reprezentován snímačem, který detekuje průjezd přes speciální Alfano smyčku na závodním okruhu a displejem na volantu stroje.

Na obrázku č. 5 je náčrt komponent celého systému.



Zdroj: Vlastní tvorba

Popis obrázku: 1 – displej, který si závodník přimontuje na volant svého stroje, 2 – čidlo, které musí být umístěno na stroj co nejbližší k vozovce

Tento systém nabízí:

- čas dosažený na kolo plus až tři mezičasy a signalizace rozdílů proti nejlepšímu kolu,
- teplota – indikace teploty a záznam maximální dosažené teploty,
- otáčky – indikace a záznam max. otáček pro každé kolo (dvou i čtyř takt),
- „motoshodiny“ - registrace provozního času pro pět motorů,
- „infraport“ – přenos dat do počítače.

Použité komponenty

Pro tento systém se využívá snímač a displej od firmy Alfano.

Použité komponenty

Komponenty jsou, podobně jako u systému Orbits, použity od firmy AMB.

Hardware požadavky PC

Základním požadavkem je operační systém Windows 2000 a novější.

Hardware požadavky PDA:

- Operační systém PocketPC verze 2000 a novější,
- procesor typu SH3 / StrongARM / Intel XScale / Intel PXA250.

Hodnocení systému

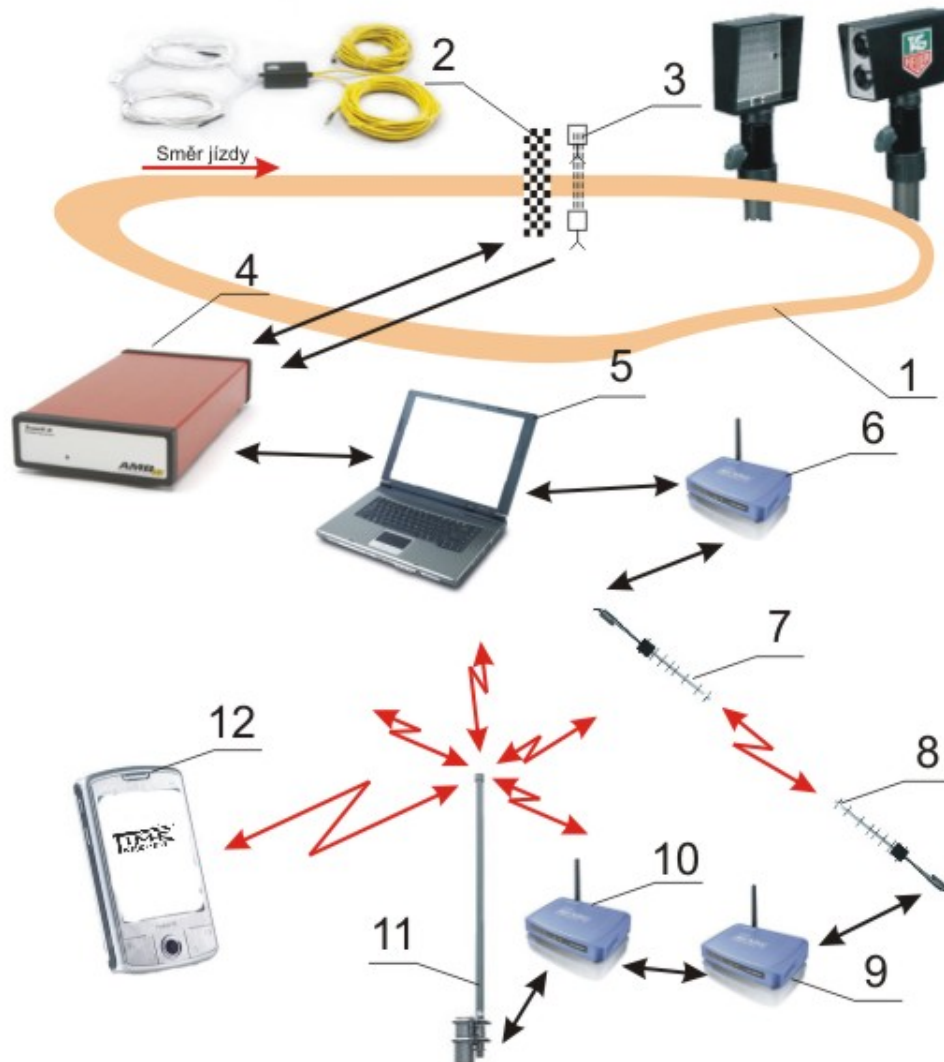
K hodnocení systému nejsem schopen se vyjádřit. V České republice se tento systém měření času nepoužívá a bohužel se mi nepodařilo najít osobu, která by měla s tímto systémem nějakou podrobnější praktickou zkušenost. Tento systém se používá například v Německu.

1.4.4 InformSys for PDA

Jedná se o můj vlastní informační systém. Nedá se sice zařadit mezi komerční programy, ale lze jej zařadit mezi konkurenční programy. Tento systém byl navržen s hlavním důrazem na zobrazování online výsledků závodníků na mobilních zařízeních typu PDA. Pracuje na podobných principech jako Orbits plus R-monitor.

Na obrázku č. 7 je náčrt komponent celého systému.

Obrázek 7: InformSys systém



Zdroj: vlastní tvorba

Popis obrázku: 1 – závodní okruh, 2 – hlavní smyčka, 3 – záložní smyčka, 4 – dekodér, 5 – PC s programem InformSys, 6 – AccessPoint, 7 – vysílací směrová anténa, 8 – přijímací směrová anténa, 9 – AccessPoint, 10 – AccessPoint, 11 – všesměrová vysílací anténa, 12 – PDA s programem InformSys for PDA version

Použité komponenty

Komponenty jsou opět použity od firmy AMB. Pro přenos výsledků do PDA/PC/Notebook se využívá WiFi bezdrátová síť.

Hardware požadavky pro PC/Notebook:

- PC s procesorem Intel Pentium III (min. 600 Mhz),
- Windows XP Pro (preferovaný), XP Home, Windows 2000,
- 128 MB vnitřní pamět,
- přibližně 5 MB místa na disku,
- přítomnost .NET Framework, MySQL server.

Hardware požadavky pro PDA:

- operační systém PocketPC 2003 a vyšší,
- přítomnost Compact .NET Framework,
- wiFi karta.

Hodnocení systému:

Výhody:

- stejná funkčnost při zobrazování online výsledků jako Orbits,
- vynahrazuje všechny nedostatky R-monitoru.

Nevýhody:

Při vysoké frekvenci průjezdů v jeden okamžik dochází k zahlcení programu. Důsledkem je, že někteří závodníci sice projedou cílem, ale systém je nezachytí.

1.4.5 Ostatní měřicí systémy

Existuje mnoho jiných systémů pro měření okruhových závodů. Tyto systémy jsou z velké části založeny na měření pomocí optických čidel⁴. Nepoužívá se transpondérů ani dekodérů. Měření běží na principu přerušení optické závoru. Optické čidlo je napojeno na hodiny, které každé přerušení závoru detekují jako průjezd. Tento průjezd doplní o systémový čas daného průjezdu a vytisknou na speciální tiskárně, která je součástí hodin.

Závodníci tedy jezdí po okruhu a každý jejich průjezd je zaznamenán na hodinách a vytisknut na nekonečnou pásku. Jeden z časoměřičů musí stále sledovat závodní dráhu a průjezdy jednotlivých závodníků, neboť se může stát, že dva závodníci projedou optickou závorou těsně vedle sebe a to má za následek pouze detekci jednoho průjezdu. Takového situace se řeší manuálním doplněním dalšího průjezdu závodníka. Další časoměřič při každém průjezdu závodníka musí přečíst jeho jedinečné startovní číslo a tuto informaci zapsat na speciální formulář. Ostatní časoměřiči buď data z hodin a ze speciálního formuláře s průjezdy a startovními čísly definují do vlastního programu nebo je píší do dalšího formuláře pro manuální počítání časů. Toto počítání se provádí prostým odčítáním časů závodníka v jednotlivých průjezdech.

⁴ Fotobuňka

Použité komponenty

Nejčastěji se používají optická čidla a hodiny od firmy TAG Heuer.

Hardware požadavky

Pokud se nepoužívá program pro počítání časů závodníků, tak žádné požadavky nejsou. Naopak při použití jsou požadavky v závislosti na nárocích konkrétních programů časoměřičů.

Hodnocení systému

Bohužel žádná výhoda neexistuje. Tento systém se masivně používal dříve. Ze stylu měření se nedá objektivně hodnotit systém v konkurenci ostatních produktů.

1.4.6 Systémy používané v České republice

V České republice se do minulých let používaly dva základní přístupy k měření času. Jedním přístupem je klasické použití optické závory a pro zpracování dat se využívá PC techniky. Druhým přístupem je použití systému Orbits a všech komponent od firmy AMB. Poslední rok se začalo využívat i mého programu InformSys for PDA, který slouží pro online zobrazování výsledků na PDA do depa.

V současnosti se masivně přechází na měření druhým přístupem.

2 MODELOVÁNÍ V UML

2.1 Co je to UML

Jazyk UML⁵ je univerzální jazyk pro vizuální modelování systémů. Přestože je nejčastěji spojován s modelováním objektově orientovaných softwarových systémů, má mnohem širší využití, což vyplývá z jeho zabudovaných rozšiřovacích mechanismů.

Jazyk UML byl navržen proto, aby spojil nejlepší existující postupy modelovacích technik a softwarového inženýrství. Jako takový je explicitně navržen takovým způsobem, aby jej mohly implementovat všechny nástroje CASE⁶. Zmíněná koncepce vychází z pochopení skutečnosti, že se rozsáhlé softwarové systémy obvykle bez podpory nástrojů CASE neobejdou. Diagramy vytvořené v jazyku UML jsou srozumitelné pro lidi, ale navíc je mohou snadno interpretovat i programy CASE.

Je nesmírně důležité abychom si uvědomili, že jazyk UML nenabízí žádný druh metodiky modelování. Přirozeně, určité aspekty metodiky můžeme najít v každém z elementů, z nichž se model UML skládá. Samotný jazyk UML však poskytuje pouze vizuální syntaxi, kterou můžeme využít při sestavování svých modelů.

Unified Process (zkráceně UP) již ovšem metodikou je. Sděluje nám, jaké pracovníky musíme využít, jaké činnosti vykonat a jaké produkty vyrobit, aby se nám podařilo sestavit model funkčního softwarového systému.

Jazyk UML není vázán na žádnou specifickou metodiku nebo životní cyklus. Lze jej použít společně se všemi existujícími metodami. UP využívá jazyk UML jako vlastní syntaxi pro vizuální modelování. Z tohoto pohledu lze metodiku UP považovat za upřednostňovanou metodu užití jazyka UML, protože je pro tento jazyk nejlépe adaptována. Jazyk UML však může poskytovat podporu vizuálního modelování i pro jiné metody.

2.2 Struktura jazyka UML

Funkci jazyka UML jako jazyka vizuálního porozumíme nejlépe, podíváme-li se na jeho strukturu, která obsahuje tyto části:

⁵ Unified Modeling Language, unifikovaný modelovací jazyk

⁶ Computer-aided software engineering

- **stavební bloky.** Bloky jsou základní prvky modelu, relace a diagramy,
- **společné mechanismy.** Obecné způsoby, jimiž v jazyku UML dosáhneme specifických cílů,
- **architektura.** Pohled v jazyku UML na architekturu navrhovaného systému.

2.2.1 Stavební bloky jazyka UML

Jazyk UML je sestaven pouze ze tří stavebních bloků:

- **předmětů,** což jsou samotné elementy modelu,
- **vztahů,** jež jsou pojátkem mezi předměty,
- **diagramů,** což jsou pohledy na modely UML, ukazující kolekce předmětů, které vyprávějí příběh o softwarovém systému a jsou naším způsobem vizualizace toho, co systém bude dělat, a toho, jak to bude dělat.

2.2.2 Obecná mechanika jazyka UML

Jazyk UML obsahuje čtyři společné mechanismy používané v celém jazyku konzistentně. Popisují čtyři strategické cesty k modelování objektů, jež jsou opakovaně používány v různých kontextech v celém jazyce UML:

- specifikace,
- ozdoby,
- podskupiny,
- mechanismy rozšiřitelnosti.

2.2.3 Architektura

Architektura je zachycením strategických aspektů vyšší struktury systému. Abychom byli schopni zachytit všechny podstatné aspekty architektury daného systému, definuje jazyk UML čtyři různé pohledy na systém:

- **logický pohled.** Zachycuje slovník oblasti problému jako množinu tříd a objektů. Důraz je přitom kladen především na zobrazení způsobu, jakým objekty a třídy tvoří základ systému implementující jeho chování,
- **pohled procesů.** Modeluje spustitelná vlákna a procesy jako aktivní třídy. Je to procesově orientovaná varianta logického pohledu, která obsahuje stejné artefakty,

- **pohled implementace.** Modeluje soubory a komponenty, které utvářejí hotový kód systému. Slouží jednak ke znázornění závislostí mezi jednotlivými komponentami, jednak toho, jak spravovat konfiguraci množin vytvořených z těchto komponent. Umožňuje definici verze systému,
- **pohled nasazení.** Modeluje fyzické nasazení komponent na množinu fyzických výpočetních uzlů. Umožňuje modelování komponent na příslušné uzly distribuovaného systému,
- **pohled případů užití.** Všechny jiné pohledy jsou odvozeny od pohledu případů užití. Tento pohled zachycuje základní požadavky kladené na příslušný systém jako na množinu případů užití a tvoří tak základ tvorby všech dalších pohledů.

2.3 Unified Process

Proces vývoje software SDP⁷, známý rovněž jako metoda tvorby softwarového vybavení SEP⁸, definuje při vývoji software otázky kdo, co, kdy a jak.

Metodika USDP⁹ je průmyslovým standardem SEP pocházejícím od autorů jazyka UML. Tento je běžně označován jako UP.

Projekt UML vznikl z potřeby nabídnout jak vizuální jazyk, tak proces tvorby softwarového vybavení. To, co známe dnes pod pojmem UML, je jazykovou částí projektu, UP je částí procesní.

2.3.1 Konkrétní aplikace metodiky UP v novém projektu

Metodika UP je obecnou metodikou tvorby software. Pro každou organizaci, stejně jako potom pro každý jednotlivý projekt, je tedy třeba vytvořit její novou instanci. Tím se uznává, že každý projekt se od ostatních liší a že model „tato košile padne všem“ zde rozhodně neplatí.

⁷ Software development process

⁸ Software engineering process

⁹ Unified Software Development Process

2.3.2 Axiomy metodiky UP

Metodika UP obsahuje tři základní axiomy. Jsou to:

- zásada řízení případem užití a rizikem,
- zásada soustředění se na architekturu,
- zásada iterace a přírůstku.

2.3.3 Metodika UP je založena na iterativním a přírůstkovém procesu

Ke správnému porozumění metodiky UP je třeba porozumět iteracím. Základní myšlenka je velice prostá – historie ukazuje, že člověk, se obecně vzato, řeší lépe menší problémy než větší. Snažíme se tedy o rozložení velkého softwarového projektu na řadu menších monoprojektů. Výsledkem je snazší správa a úspěšné dokončení. Klíčovým východiskem je skutečnost, že iterace obsahuje všechny prvky normálního softwarového projektu:

- plánování,
- analýza a návrh,
- tvorba,
- integraci a testování,
- interní nebo externí uvedení.

2.3.4 Pracovní postupy iterace

V každé iteraci existuje pět základních pracovních postupů jež určují, co je třeba udělat a způsob, jakým toho dosáhnout. Kromě pěti základních pracovních postupů obsahuje obvykle každá iterace ještě další pracovní postupy jako jsou plánování, odhad a vše, co je pro danou iteraci specifické. Tyto pracovní postupy ovšem v metodice UP zajištěny nejsou. Základní pracovní postupy jsou:

- **požadavky.** Zachycují to, co by měl systém dělat,
- **analýza.** Vybroušení požadavků a jejich strukturování,
- **návrh.** Realizace požadavků v architektuře systému,
- **implementace.** Tvorba softwaru,
- **testování.** Ověření, zda implementace funguje tak, jak se od ní očekává.

2.4 Využití UML v modelování informačního systému okruhových závodů

V této diplomové práci se snažíme využít vizuálního vyjadřování za pomoci prostředků jazyka UML. Naší snahou je zaměřit se na postup vývoje informačního systému pro měření okruhových závodů.

Pro tvorbu a popis programu jsme použili metodiku Unified Process. Vývoj software dle dané metodiky se dělí do určitých dílčích celků, které však nemohou být chápány jako chronologický postup, kde ukončení jednoho dílčího celku znamená začátek následného. Tyto dílčí celky se vzájemně prolínají a částečně paralelně existují. Je zde však zaručena základní myšlenka tvorby systému od přijetí a zpracování požadavků, až po zavedení a testování.

Každý dílčí celek bude popisován ve dvou úrovních. První částí každého kroku vývoje bude jeho teoretický popis a v druhé následující části se zaměříme na konkrétní realizaci a aplikaci informačního systému okruhových závodů.

Postup vývoje software:

- požadavky,
- analýza,
- návrh,
- implementace,
- zavedení,
- testování.

3 POŽADAVKY

3.1 Pracovní postup požadavků

Ještě předtím, než se vůbec pustíme do objektově orientované analýzy či návrhu, musíme mít alespoň rámcový přehled o tom, čeho vlastně chceme dosáhnout a jaký je smysl požadavků a jejich specifikace. Musíme jednak zjistit, co má vlastně systém dělat, jednak v tomto bodu dosáhnout shody. Vše by mělo být popsáno v terminologii, kterou používají uživatelé budoucího systému. Tvoříme vyšší specifikaci toho, co má systém dělat – což je označováno jako inženýrství požadavků¹⁰.

V podstatě každý projekt může mít více typů uživatelů, techniků údržby, pomocného personálu, prodavačů, manažerů apod. Inženýrství požadavků spočívá ve stanovení služeb, které by měl vyvíjený systém poskytovat, a omezení, za nichž musí pracovat. Spočívá rovněž v získání požadavků, jaké na nový systém mají jeho uživatelé a v sestavování jejich priority. Je to proces vyjednávání, neboť obvykle je třeba vyřešit a vyvážit i mnoho protichůdných požadavků.

3.2 Informační systém okruhových závodů

3.2.1 Obecný popis systému:

System se používá pro zpracovávání časů závodníků při okruhových závodech.

Jádrem celého systému je program, jehož primárním úkolem je načítat data z dekodéru. Dekodér je zařízení, které je napojeno na smyčku zabudovanou do závodní dráhy. Každý závodník má na svém stroji přístroj zvaný transpondér, který při průjezdu přes cílovou smyčku je touto smyčkou detekován a vyšle prostřednictvím ní signál do dekodéru. Transpondér má své unikátní číslo a tímto číslem se pro systém identifikuje. Dekodér dále signál zpracuje a předá informaci do systému. Tato informace obsahuje čas průjezdu a identifikační číslo transpondéru.

Program získaná data načítá z dekodéru a dále zpracovává. Dále dokáže definovat závodníka a jednotlivé transpondéry. Každý závodník má definováno svoje číslo transpondéru. Tyto informace se v programu definují ještě před průběhem závodu.

¹⁰ Requirements engineering

Další částí systému je program běžící na mobilním zařízení typu PDA. Tento program se připojí k databázi, do které jsou ukládána data o průjezdech závodníků (zajišťováno programem výše popsaným), a data zobrazuje uživateli. Tato data jsou v definovaných cyklech aktualizována.

3.2.2 Požadavky na informační systém

Hlavní program

Požadavky:

- komunikovat s dekodérem, načítat data z dekodéru,
- komunikovat s databází, načítat a ukládat data,
- data načíst z dekodéru, dále je zpracovat a uložit do databáze,
- vytvářet, rušit a editovat závodníky,
- vytvářet, rušit a editovat transpondéry.

Program na PDA

Požadavky:

- připojit se k databázi,
- sledovat připojení k databázi,
- načítat data z databáze a zobrazovat uživateli.

Vstupy systému:

- informace z dekodéru,
- přihlášky závodníků s informací o závodníkovi a přiděleném transpondéru.

Výstupy systému

Data zobrazená uživateli v tabulkové podobě. Data jsou seříděna podle definovaného klíče.

Uložení dat

Data budou uložena v databázi

Požadavky na implementaci systému:

- jednoduchá ovladatelnost a uživatelsky příjemné a intuitivní prostředí,
- komunikace s dekodérem,
- data ukládat do databáze,
- vytvořit počítačovou síť.

Hlavní cíl systému

System načítá data z dekodéru. Dále je zpracovává a posílá přes vybudovanou síť na zařízení typu PDA, kde jsou uživateli zpřístupněna uživatelsky příjemným způsobem.

4 ANALÝZA

4.1 Pracovní postup analýzy

4.1.1 Modelování případů užití

Modelování případů užití je jednou z forem „inženýrství požadavků“. Je jedním ze způsobů získávání a dokumentování požadavků. Modelování se skládá z následujících aktivit:

- nalezení hranic systému,
- vyhledání účastníků,
- nalezení případů užití,
- specifikace případů užití,
- tvorba scénářů.

Výstupem uvedených aktivit je model případu užití. Tento model obsahuje čtyři komponenty:

- **účastníci.** Jsou to role, přidělené osobám nebo předmětům používajícím systém,
- **případy užití.** Činnosti, které mohou účastníci se systémem vykonávat,
- **relace.** Smysluplné vztahy mezi účastníky a případy užití,
- **hranice systému.** Ohraničení zobrazené kolem případů užití, jež je vyznačením území nebo hranic modelovaného systému.

Hranice systému

První věcí, kterou musíme udělat, přemýšlíme-li o tvorbě softwarového systému, je stanovení jeho hranic. Jinými slovy to znamená, že musíme určit, co je součástí systému a co naopak není jeho součástí.

Účastníci

Účastník specifikuje roli, kterou určitá externí entita přijímá v okamžiku, kdy začíná daný systém bezprostředně používat. Může vyjadřovat roli uživatele, roli dalšího systému, který se dotýká hranic našeho systému.

Případ užití (Use Case)

Projděme seznam účastníků a zvažme způsob, jímž bude každý z nich systém používat. Pomocí této strategie můžeme vytvořit seznam případných případů užití. Název každého případu užití musí být slovesnou vazbou.

Při určování případů můžeme leckdy najít nové účastníky.

Modelování případů užití je iterativní proces a postupuje vpřed postupným upřesňováním. Nejprve začneme s pouhým názvem případů užití. Později začneme k názvu připojovat další podrobnosti. Zmiňované podrobnosti se skládají z počátečních krátkých popisů, které nakonec upřesníme do úplné specifikace.

4.1.2 Slovníček pojmů

Slovníček pojmů daného projektu může být jedním z jeho nejdůležitějších artefaktů. Každé odvětví má vlastní žargon, jazyk, terminologii. Hlavním smyslem inženýrství požadavků a analýzy spočívá v pochopení a zachycení tohoto jazyka. Slovníček pojmů poskytuje lexikon klíčových obchodních termínů a definicí.

4.1.3 Sekvenční diagramy

Sekvenční diagramy slouží k zobrazení interakcí ve vztahu k času. Jsou izomorfní vzhledem k diagramům spolupráce a kromě elementů sdílených s diagramy spolupráce obsahují ještě další dva elementy – čáru života objektu a aktivaci. V diagramech spolupráce označuje vnoření pořadových čísel aktivaci. V sekvenčních diagramech lze ovšem aktivaci zobrazit mnohem zřetelněji a explicitněji.

Sekvenční diagramy slouží v objektově orientované analýze k poněkud jiným účelům než je tomu v případě diagramů spolupráce. Diagramy spolupráce jsou velmi dobrým prostředkem k zobrazení skutečných objektů a jejich strukturálních relací. Jsou však slabší v chronologickém zobrazení interakcí jako časově uspořádané posloupnosti událostí. Právě v tomto směru mají sekvenční diagramy velkou výhodu. Sekvenční diagramy se také vyskytují v obecné i konkrétní podobě. Většinou se používá diagram konkrétních sekvencí, právě proto se na něj zaměřím.

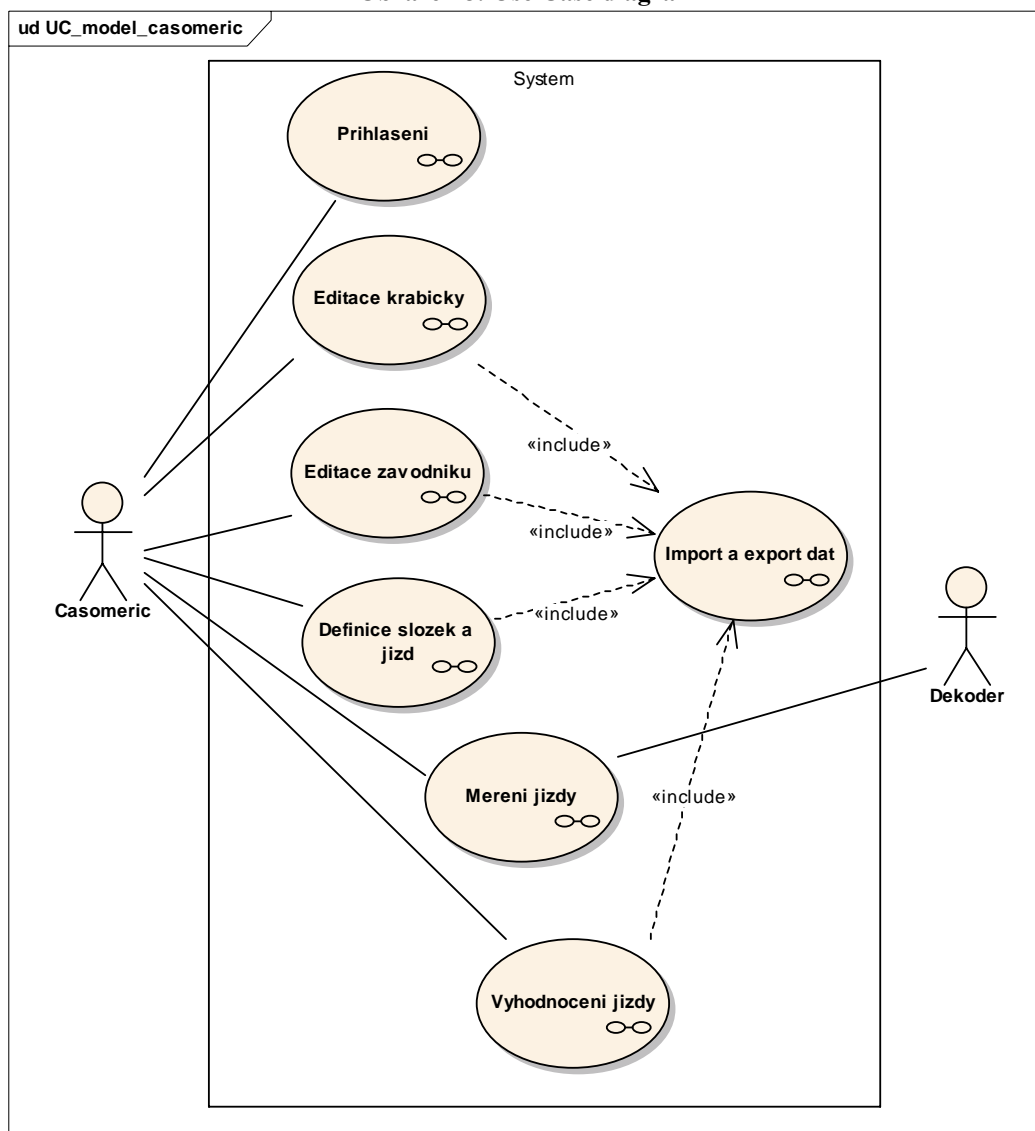
Modelování obvykle začínáme náčrtem realizace případů užití pomocí diagramu spolupráce. V něm je totiž velice snadné nejen rozmístění objektů, ale i jejich vzájemné propojení. Chceme-li se však zaměřit na chronologické uspořádání událostí ve vztahu k času, je vždy lepší použít sekvenční diagram. Vzhledem k tomu, že jsou oba typy diagramů pouze jinými pohledy na tentýž model, mohou je nástroje CASE velmi snadno převádět z jednoho na druhý.

4.1.4 Analýza v UML

Veškeré případy užití a sekvenční diagramy, které jsou v této kapitole vyobrazeny, musíme chápat abstraktně. Elementy diagramů jsou pouze rozšířením a zpracováním požadavků od zákazníka. Tyto požadavky jsou zpracovány do podoby případů užití a sekvenčních diagramů, aby byly dobře chápány zákazníkem a bylo možné se zákazníkem o problémové doméně debatovat.

4.2 Případy užití systému

Obrázek 8: Use Case diagram



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Diagram zobrazuje případy užití systému pro měření času okruhových závodů. Na obrázku můžeme vidět, že externím uživatelem systému je na jedné straně časoměřič, který je hlavním a jediným uživatelem, který zasahuje a ovlivňuje chod systému a na druhé straně je

to dekodér, který, jakožto externí zařízení, hraje důležitou roli při načítání naměřených časů do systému.

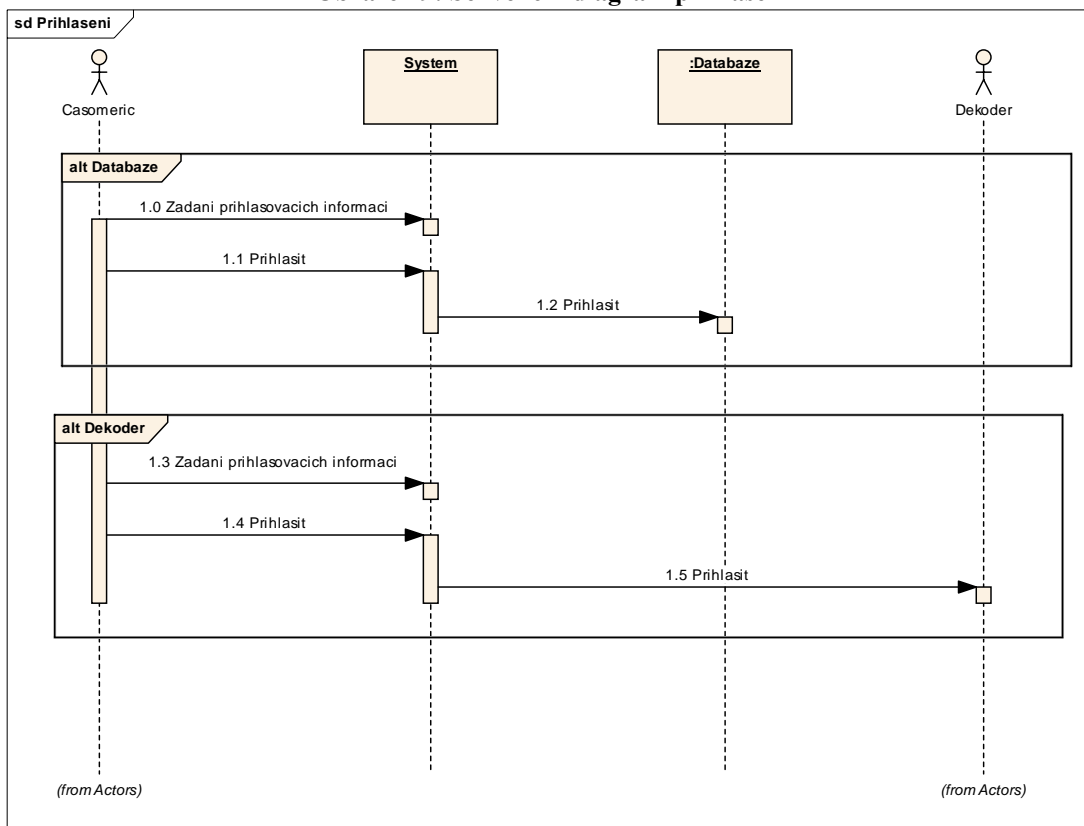
Pokud se podíváme blíže na obrázek č. 8, můžeme vidět zbarvené elipsy, které představují jednotlivé případy užití. Každý z případů užití představuje určitý požadavek uživatele. Obdélník představuje hranice systému a elementy, které stojí mimo systém jsou externí uživatelé, kteří k systému přistupují a ovládají systém způsobem, který je zastoupen jednotlivými případy užití. Každé spojení uživatele a případu užití je vztah, ve kterém externí uživatel klade požadavek na systém. Vztahy případů užití v rámci systému jsou situace, kdy jeden případ užití v sobě obsahuje jiný případ užití.

Příkladem můžeme uvést, vztah uživatele „časoměřič“ s případem užití „editace krabičky“. Jde o požadavek časoměřiče na možnost editovat krabičky. Tento případ užití v sobě obsahuje možnost časoměřiče editovat, přidávat a odstraňovat krabičky. Dále tento případ užití v sobě obsahuje případ užití import a export dat, neboť uživatel má i možnost importovat a exportovat seznam krabiček ze systému případně do systému.

V následujících podkapitolách budou jednotlivé případy blíže specifikovány a zobrazeny v podobě sekvenčních diagramů.

4.2.1 Přihlášení

Obrázek 9: Sekvenční diagram přihlášení



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Popis obrázku: sd – sekvenční diagram, alt – alternativa

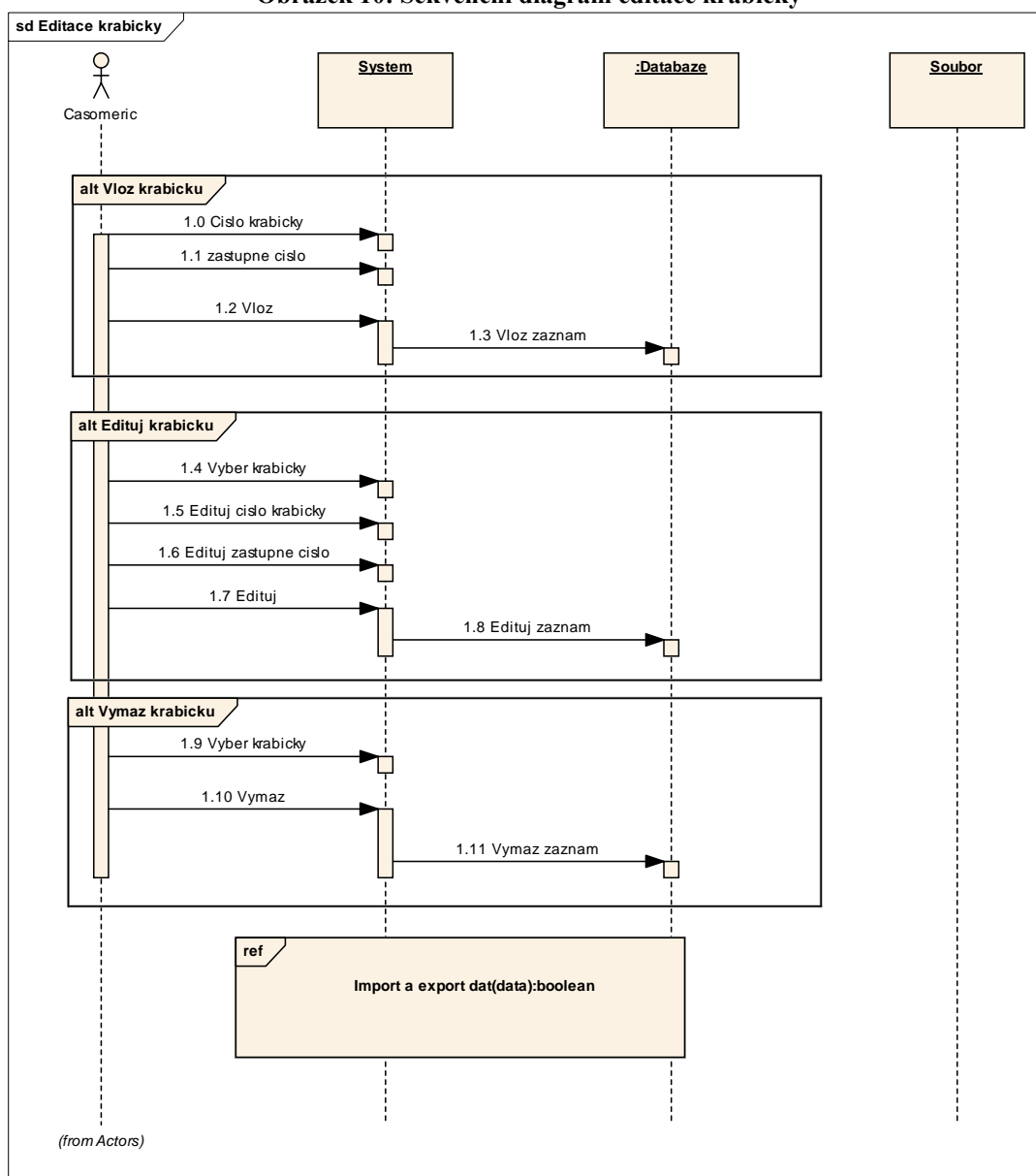
Přihlášení je zde uváděno ne ve smyslu přihlášení do systému, ale jako navázání komunikace s databází a s dekodérem. V požadavcích na systém je uvedeno, že data aplikace mají být uložena v databázi. Dále je v požadavcích uvedeno, že aplikace musí navázat komunikaci s dekodérem a komunikovat s ním při získávání dat o průjezdech jednotlivých závodníků, přesněji jejich transpondérů.

V tomto smyslu je zde uveden tento případ užití, kdy uživatel naváže komunikaci jednak s databází pro uložení dat, a dále s dekodérem pro možnost načítání informací o průjezdech závodníků.

V obou případech je vidět, že navazování probíhá velice podobným způsobem. Nejprve uživatel zadá přihlašovací informace a následně dá požadavek na přihlášení. V další fázi systém provede podle zadaných informací konkrétní přihlášení.

4.2.2 Editace krabičky

Obrázek 10: Sekvenční diagram editace krabičky



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Dalším základním požadavkem na systém je editace krabiček neboli transpondérů. Systém musí být schopen uchovávat záznamy o krabičkách, editovat tyto záznamy a odstraňovat.

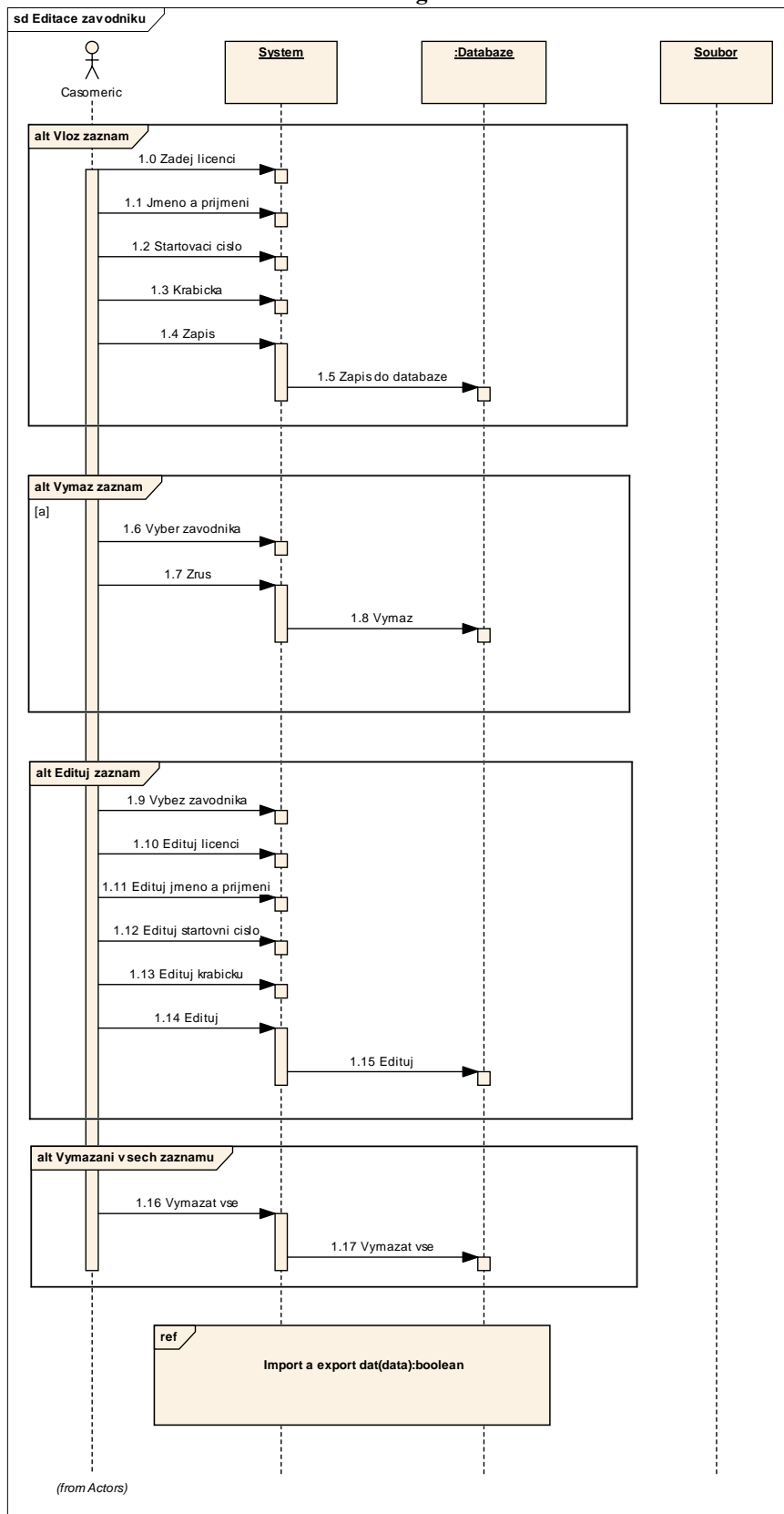
Krabička je pro systém měření času nejdůležitějším zařízením. Pokud závodní stroj tuto krabičku nemá nebo není funkční, je velice obtížné takovýto průjezd zaznamenat.

Mezi základní operace, které systém umožňuje ve vztahu k evidování krabiček, patří:

- **vložení krabičky.** Uživatel zadá požadované informace a systém vloží nový záznam do databáze,
- **editace krabičky.** Uživatel nejprve vybere jednu z evidovaných krabiček. Edituje informace o krabičce a následně provede uložení,
- **odstranění krabičky.** Uživatel nejprve vybere jednu z evidovaných krabiček, kterou následně z evidence odstraní,
- **import a export krabiček do a ze souboru.** Importem a exportem se budeme zabývat ve zvláštním případě užití viz. kapitola „Import a export dat“.

4.2.3 Editace závodníků

Obrázek 11: Sekvenční diagram editace závodníků



Zdroj: vlastní tvorba v prostředí Enterprise Architect

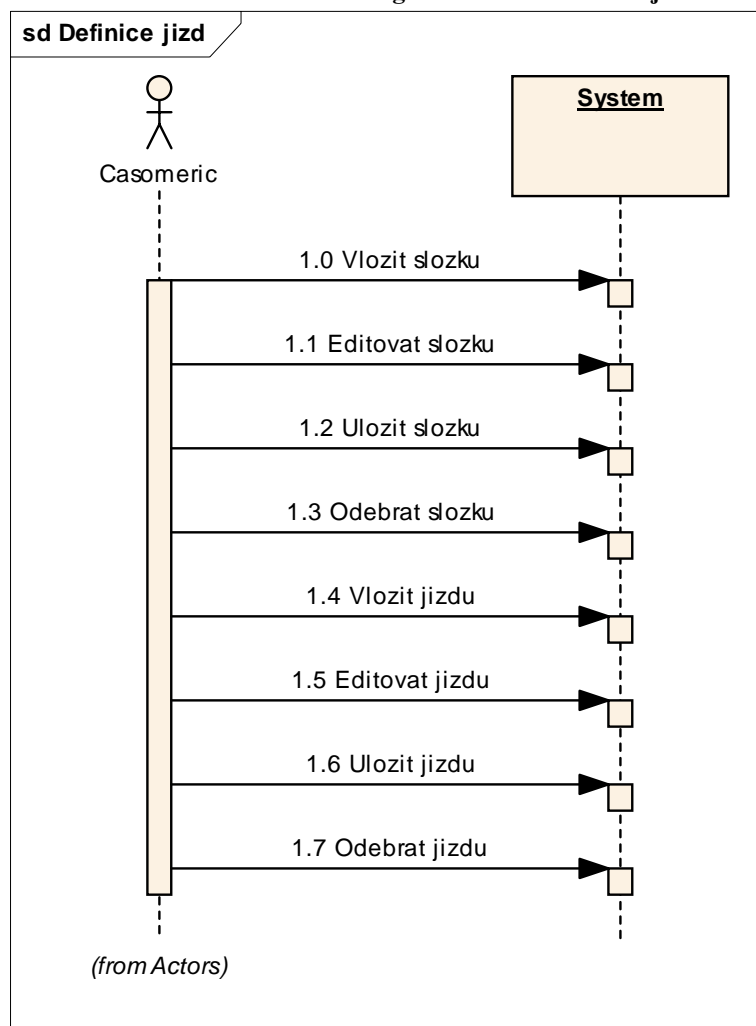
Společně s vedením evidence o krabičkách je i vedení záznamů o jezdcích další z velice důležitých úkolů systému.

Zatím je pro systém každý průjezd jen informace o čísle krabičky, která přes cílovou čáru projela. Právě udržování informací o závodnících a jejich propojení s číslem krabičky dává přesnou a uživatelsky úplnou zprávu o tom, kdo a kdy projel a byl systémem zaznamenán.

Operace spojené s evidencí závodníků jsou rámcově totožné s evidencí krabiček, a proto se jimi již v této kapitole nebudu zabývat.

4.2.4 Definice složek a jízd

Obrázek 12: Sekvenční diagram definice složek a jízd



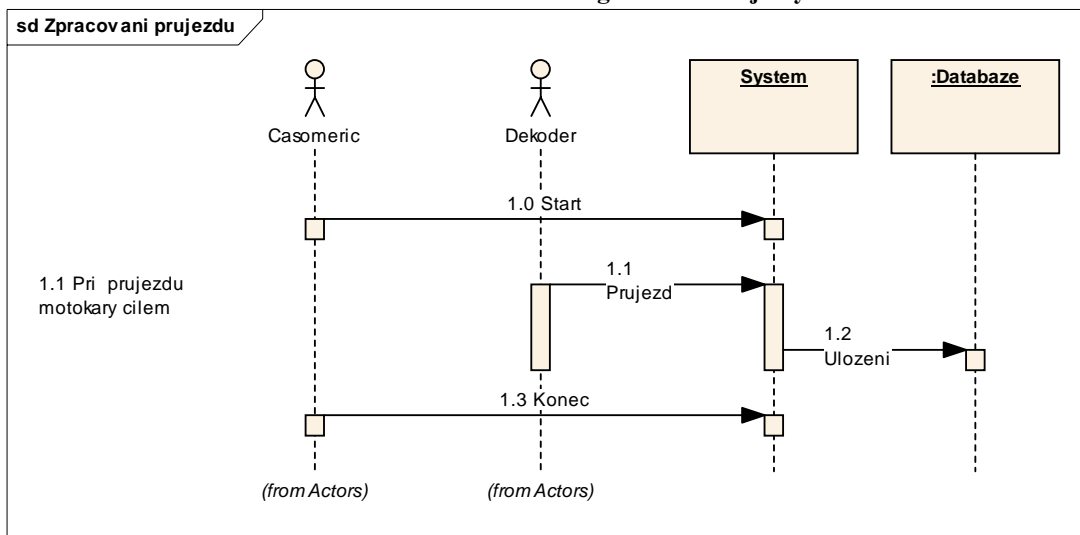
Zdroj: vlastní tvorba v prostředí Enterprise Architect

Dalším případem užití je definice složek a jízd. Z důvodů rozdělení startovního pole do několika kategorií, bylo vhodné a potřebné přizpůsobit systém pro tuto situaci. Uživatel

má možnost si v programu vytvářet, editovat a mazat složky pro jednotlivé kategorie, do kterých může vkládat, editovat a odstraňovat jednotlivé jízdy.

4.2.5 Měření jízdy

Obrázek 13: Sekvenční diagram měření jízdy



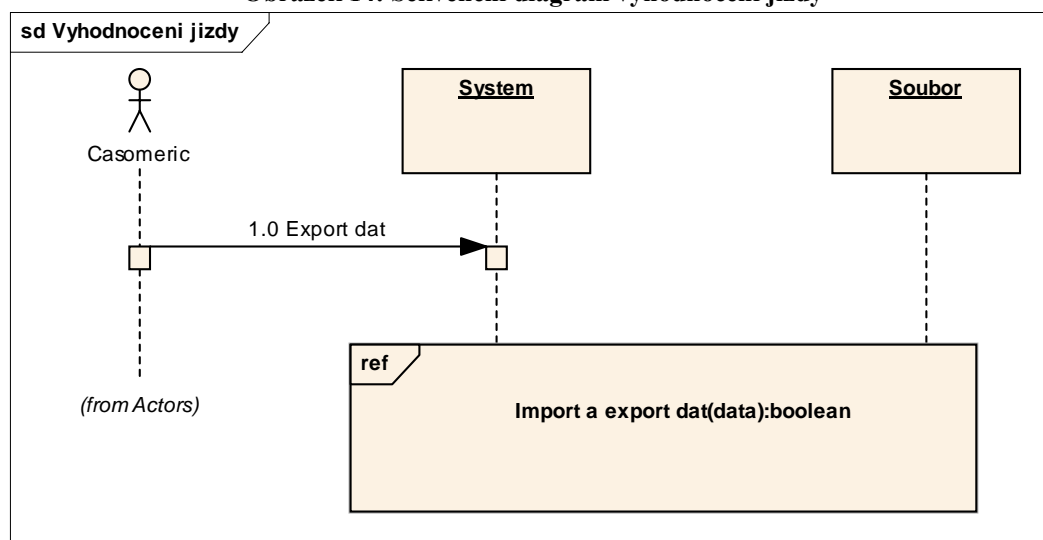
Zdroj: vlastní tvorba v prostředí Enterprise Architect

Tento případ užití je jádrem celého systému. Jedná se o situaci, kdy uživatel spouští měření času.

System přijímá informace z dekodéru, dále je zpracovává a následně ukládá do databáze. Tento proces se opakuje do doby, kdy uživatel načítání času neskončí.

4.2.6 Vyhodnocení jízdy

Obrázek 14: Sekvenční diagram vyhodnocení jízdy

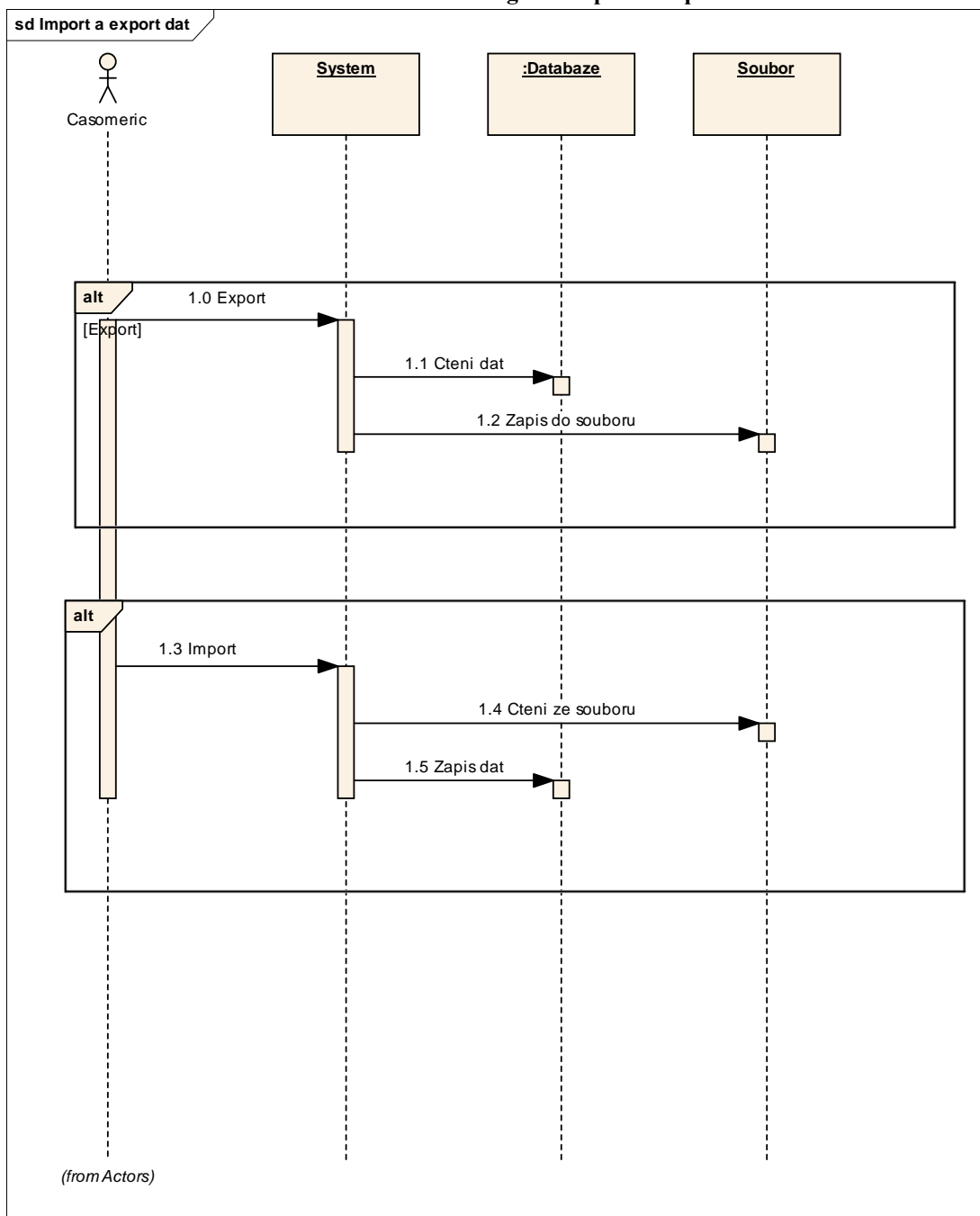


Zdroj: vlastní tvorba v prostředí Enterprise Architect

Po skončení jízdy má možnost uživatel systému exportovat report dané uplynulé jízdy. S importem a exportem se podrobněji seznámíme v případě užití viz kapitola „Import a export dat“.

4.2.7 Import a export dat

Obrázek 15: Sekvenční diagram import a export dat



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Tento případ užití se stará o obsluhu požadavků na import a export dat. Při importu jsou data načítána z externího souboru do systému a naopak při exportu jsou ze systému vkládána do externího souboru.

V systému se data podle zvolené operace zapisují do databáze nebo čtou z databáze.

4.3 Slovníček pojmů

- **Transpondér.** Krabice, která je připevněná na stroji závodníka.
- **Dekodér.** Zařízení, které zaznamenává průjezdy závodníků přes cílovou pásku a data posílá do systému.
- **Časoměřič.** Osoba, která je oprávněná a kvalifikovaná k obsluze systému.
- **Závodník.** Licencovaný účastník závodu, který byl řádně prezentován a jeho závodní stroj přejet.

4.4 Zaměření pracovního postupu analýzy

V této fázi se zaměřujeme na upřesnění požadavků na systém a převedení těchto informací do jazyka UML a jeho vizuální podoby.

5 NÁVRH

5.1 Pracovní postup návrhu

Hlavní důraz prvních iterací klademe na požadavky a na analýzu. Během postupného upřesňování analýzy se modelování stále více zaměřuje na návrh. Aktivity analýzy a návrhu mohou do určité míry probíhat paralelně. Je však velmi důležité, aby bylo správně rozlišeno mezi artefakty obou aktivit.

Analýza je zaměřena hlavně na tvorbu logického modelu připravovaného systému, který zachycuje funkce, jež tento systém musí poskytovat, aby uspokojil požadavky uživatelů. Smyslem návrhu je přesná specifikace způsobu, jak takové funkce implementovat. Na tuto otázku se lze dívat z pohledu problémové domény¹¹, ale i z pohledu domény řešení¹². Požadavky přicházejí z problémové domény. Analýza je vlastně zkoumáním této domény z pohledu uživatelů systému a dalších zainteresovaných osob. Návrh spočívá ve sloučení technických řešení z domény řešení za účelem vytvoření modelu systému, který skutečně lze implementovat.

Během návrhu rozhodují návrháři objektově orientovaného systému o strategických otázkách, např. o perzistenci objektů, o distribuci objektů a o tvorbě příslušného návrhového modelu. Vedoucí projektu a inženýr projektu by měli vytvořit zásady, na jejichž základě bude možné se vypořádat se všemi taktickými otázkami týkajícími se návrhu.

5.1.1 Návrhové třídy

Návrhové třídy jsou třídy, jejichž specifikace je na takovém stupni, že je lze implementovat.

Během analýzy je zdrojem tříd problémová doména. Je to množina požadavků, která popisuje problém, jež se snažíme vyřešit. Zdrojem analytických tříd mohou být případy užití, specifikace doprovodných požadavků, slovníčky pojmů a jakékoli další související informace.

¹¹ Obchodní požadavky

¹² Podrobné úvahy na téma návrhu

Návrhové třídy lze získat ze dvou zdrojů:

- z **problémové domény** prostřednictvím analytických tříd. Součástí upřesňování je rovněž doplňování implementačních detailů,
- z **domény řešení**, která představuje řešení. Je sférou knihoven uživatelských tříd a znovupoužitelných komponent.

5.1.2 Diagram aktivit

Diagramy aktivit jsou objektově orientovanými diagramy toků. Díky nim můžeme modelovat proces jako kolekci aktivit a přechodů mezi nimi. Diagramy aktivit jsou ve skutečnosti obdobou stavových diagramů, v nichž stavy reprezentují vykonávání aktivit a přechody jsou vyvolány ukončením aktivity.

Diagram aktivit lze připojit k libovolnému modelovanému elementu, umožní modelovat jeho chování.

Diagramy aktivit lze s velkým úspěchem použít rovněž k modelování obchodních procesů a pracovních postupů.

Přestože jsou diagramy aktivit určeny především k řazení aktivit, může být skutečný zdrojový kód operace jejím nejlepším a nejpregnantnějším vyjádřením. Vždy je třeba rozhodovat podle podstaty problému.

5.1.3 Stavový diagram

Diagramy aktivit jsou v podstatě zvláštním případem stavových diagramů, v nichž jsou stavy vyjádřeny jako akce nebo stavy vnořených aktivit a v nichž jsou přechody spouštěny automaticky po ukončení předchozích akcí nebo aktivit. Diagramy aktivit obvykle využívají pouze malou podmnožinu bohaté syntaxe stavových diagramů UML.

Přestože dynamické chování systému modelujeme prostřednictvím diagramů aktivit a stavových diagramů, liší se zmiňované typy diagramů svým účelem. Diagramy aktivit jsou používány k modelování obchodních procesů, jichž se účastní několik objektů. Stavové diagramy se naproti tomu používají k modelování životního cyklu jednoho reaktivního objektu.

Reaktivní objekt je objekt, který poskytuje kontext pro stavový diagram. Reaktivní objekty:

- reagují na vnější události,
- jejich životní cyklus je modelován jako řada stavů, přechodů a událostí,
- jejich chování je důsledkem předchozího chování.

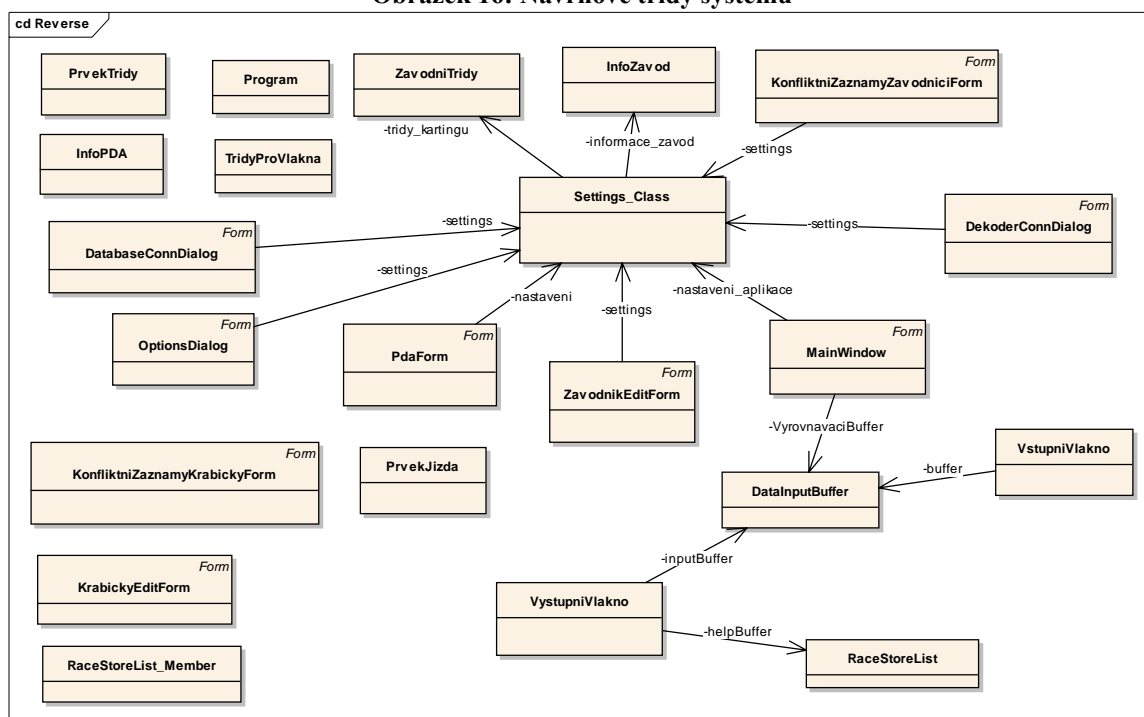
5.1.4 Datový model

Datový model je specifický tím, že:

- odděluje data, která jsou chápána jako relace, od jejich implementace,
- přístup k datům je symetrický, tj. při manipulaci se nezajímáme o přístupové mechanismy k datům,
- pro omezení redundance dat v relační databázi je zde normalizace dat.

5.2 Návrhové třídy

Obrázek 16: Návrhové třídy systému



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Na výše obrázku č. 16 je kompletní výčet tříd, které jsou v systému pro měření času použity. Na první pohled si můžeme všimnout, že v systému je propracované nastavení systému představováno třídou „Settings_Class“, které zasahuje téměř do všech dalších návrhových tříd. Jedná se o nastavení aplikace, které je nutností pro pohodlné ovládání

systemu. Uživatel informace do systému zadá pouze jednou a následně jsou mu stále nabízené, neboť tyto informace se ukládají do speciálního souboru, který je při dalších spuštěních programu načten a použit.

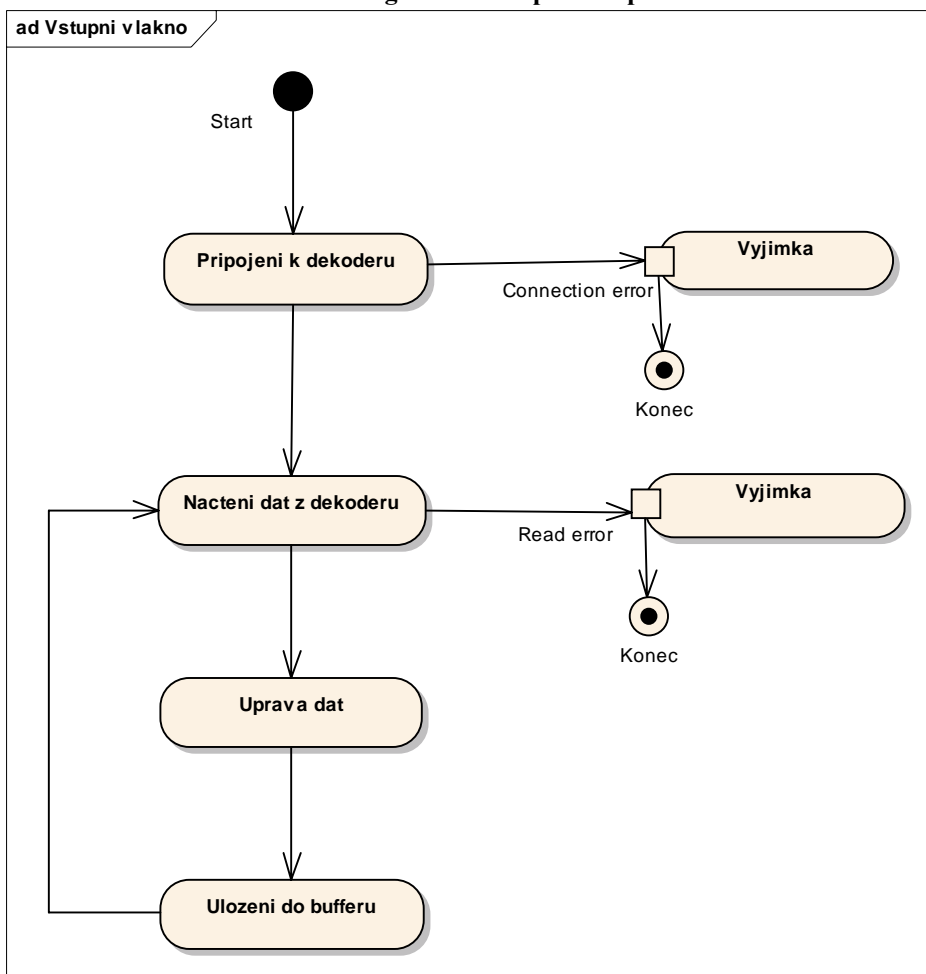
Není nezbytné se zabývat každou návrhovou třídou samostatně. V dalších kapitolách jsme se spíše zaměřili na logický pohled na systém, jeho aktivity a stavy. Dále konkrétněji na dvě hlavní vlákna, která jsou v systému použita. Tato vlákna představují jádro aplikace a slouží pro načítání dat z dekodéru, zpracovávání těchto dat a následné ukládání do databáze.

5.3 Diagramy aktivit

V diagramech aktivit jsme se zaměřili na jádro systému, které je představováno dvojicí vláken, která se starají o načítání informací z dekodéru, další zpracování těchto informací a následné uložení do databáze.

Vstupní vlákno

Obrázek 17: Diagram aktivit pro vstupní vlákno



Zdroj: vlastní tvorba v prostředí Enterprise Architect

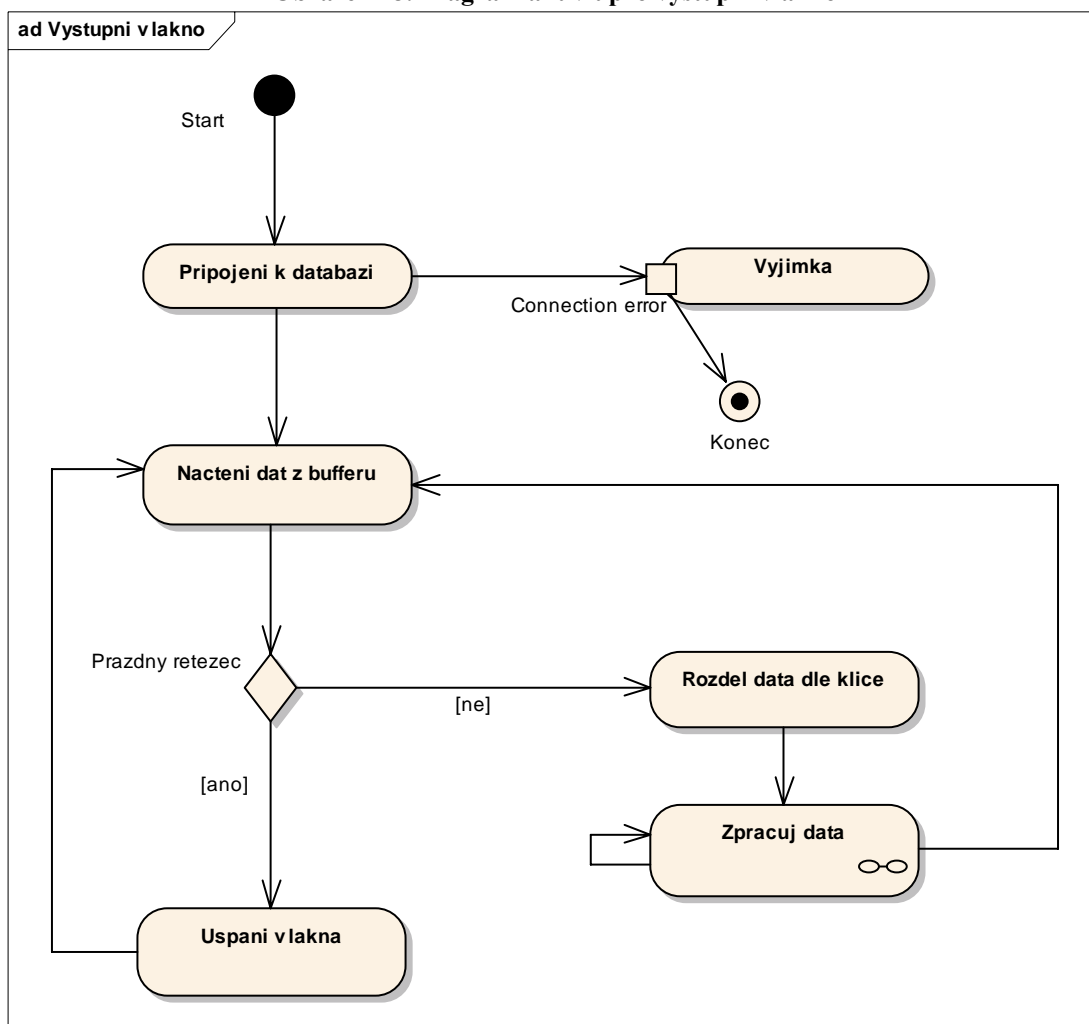
V první fázi dojde k připojení k dekodéru. Pokud připojení selže, vyvolá se výjimka a vlákno se ukončí. Pokud však dojde ke správnému připojení, čeká se na informaci, kterou do systému zasílá dekodér. Po příchodu informace dochází k úpravě a uložení informace do bufferu. Následně se opět čeká na další informaci, kterou zašle dekodér do systému.

Toto vlákno je pro svoji jednoduchost velice rychlé a jediná jeho činnost tkví v načtení informace z dekodéru a uložení do bufferu.

Tento buffer stojí mezi vstupním a výstupním vláknem. Vstupní vlákno do tohoto bufferu zapisuje informace a výstupní vlákno z tohoto bufferu informace načítá a dále zpracovává. Každý jeden záznam v bufferu představuje informaci, kterou zaslal dekodér do systému.

Výstupní vlákno

Obrázek 18: Diagram aktivit pro výstupní vlákno



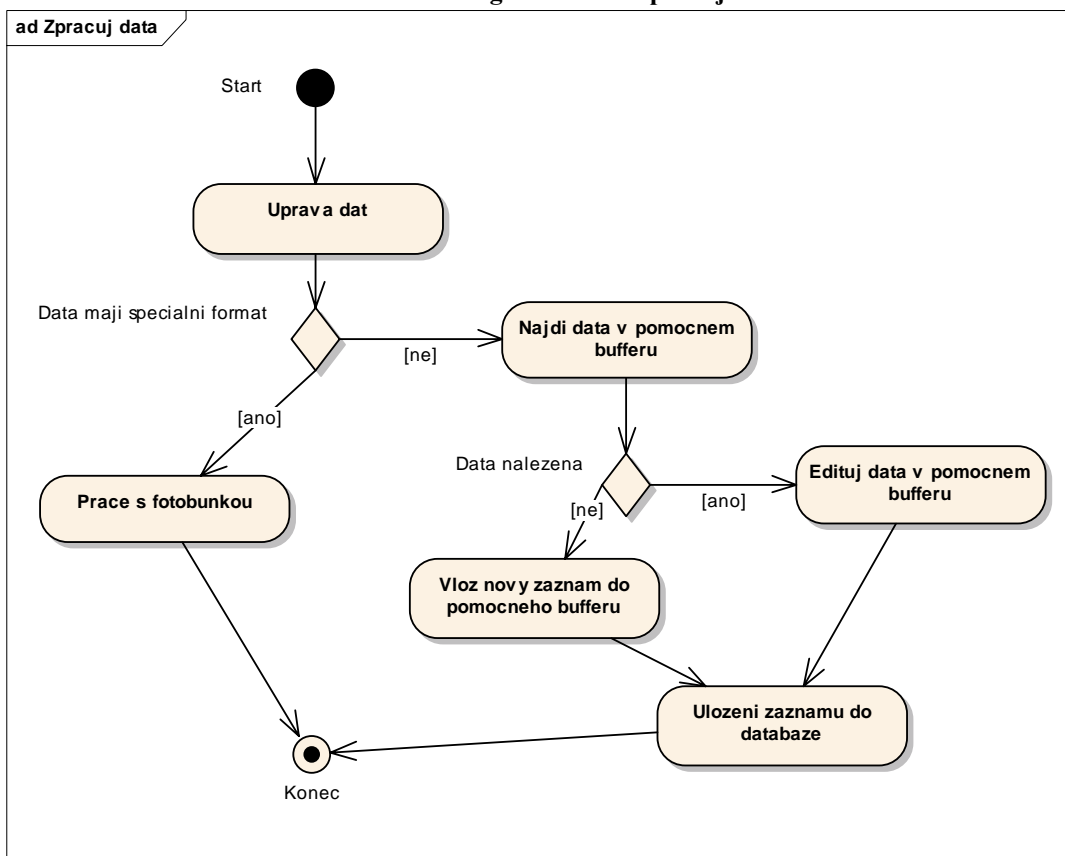
Zdroj: vlastní tvorba v prostředí Enterprise Architect

Toto vlákno se nejprve připojí k databázi. Tato operace může vyvolat výjimku a ukončit vlákno. Pokud však dojde ke správnému připojení k databázi, pokusí se vlákno načíst záznam z bufferu. V případě, že se záznam nenačte, vlákno se uspí na stanovenou dobu a následně se pokusí záznam opětovně načíst. Jestliže se záznam z bufferu načte, pokusí se vlákno rozdělit záznam dle definovaného klíče.

Pokus o rozdělení záznamu se provádí z důvodu, kdy projede cílovou čarou v jednu chvíli větší množství závodníků se svými transpondéry. Dekodér do jedné informace, kterou zasílá do systému vloží více jak jeden průjezd závodníka. Tedy jedna informace obsahuje více informací, které jsou za běžného standardního provozu na závodní dráze zasílány jako samostatné informace.

Veškeré záznamy jsou následně zpracovány.

Obrázek 19: Diagram aktivit zpracuj data



Zdroj: vlastní tvorba v prostředí Enterprise Architect

V rámci zpracování dat se nejprve provede úprava záznamu. Následně se kontroluje zda záznam obsahuje speciální formát.

Záznamy z dekodéru se dají rozdělit na dva základní typy. Jedním typem je záznam o průjezdu závodníka s číslem jeho transpondéru a druhým typem je informace z fotobuňky. Dalo by se říci, že každý průjezd je zaznamenán dvojmo. Jednou za pomoci smyčky a transpondéru a poté protnutím optické závory fotobuňky. Z tohoto důvodu se testuje, zda záznam je ve speciálním formátu, který představuje informaci získanou primárně z fotobuňky a ne ze smyčky.

Pokud tedy záznam obsahuje speciální formát uplatňuje se zde čas, který je nastaven uživatelem a představuje mezičas mezi časem, který detekuje dekodér ze smyčky a časem, který je detekován dekodérem jako průjezd fotobuňkou. Následně se systém pokusí dohledat nějaký průjezd smyčkou, který se do tohoto mezičasu vejde. Pokud takový průjezd je nalezen další zpracování průjezdu se neprovádí. Pokud však tento čas neexistuje, představuje to situaci, kdy například transpondér nefunguje a smyčka ho nedetekovala. V tu chvíli se vytvoří průjezd, který není definován, ale je zaznamenán a uživatel musí následně dodefinovat, který transpondér cílovou čáru v daném čase protnul.

Pokud záznam speciální formát neobsahuje, jde o informaci získanou ze smyčky.

V tomto vlákně se pro urychlení používá lokální buffer, který slouží pro uložení záznamů a nezatěžování databáze. Tento buffer obsahuje informace o jednotlivých transpondérech a jejich posledních průjezdech a časech těchto průjezdů.

Z tohoto důvodu se nejprve systém snaží najít informaci o transpondéru v tomto lokálním bufferu. Pokud tuto informaci nalezne, jedná se o transpondér, který již dříve cílovou čárou projel a lze získat hodnoty:

- celkový čas závodníka,
- nejlepší kolo závodníka,
- čas posledního kola závodníka,
- počet kol závodníka a kolo, ve kterém byl dosažen nejrychlejší čas,
- zlepšení případně zhoršení času posledního kola s ohledem na kolo předcházející,
- další časové údaje o daném transpondéru a jeho průjezdech.

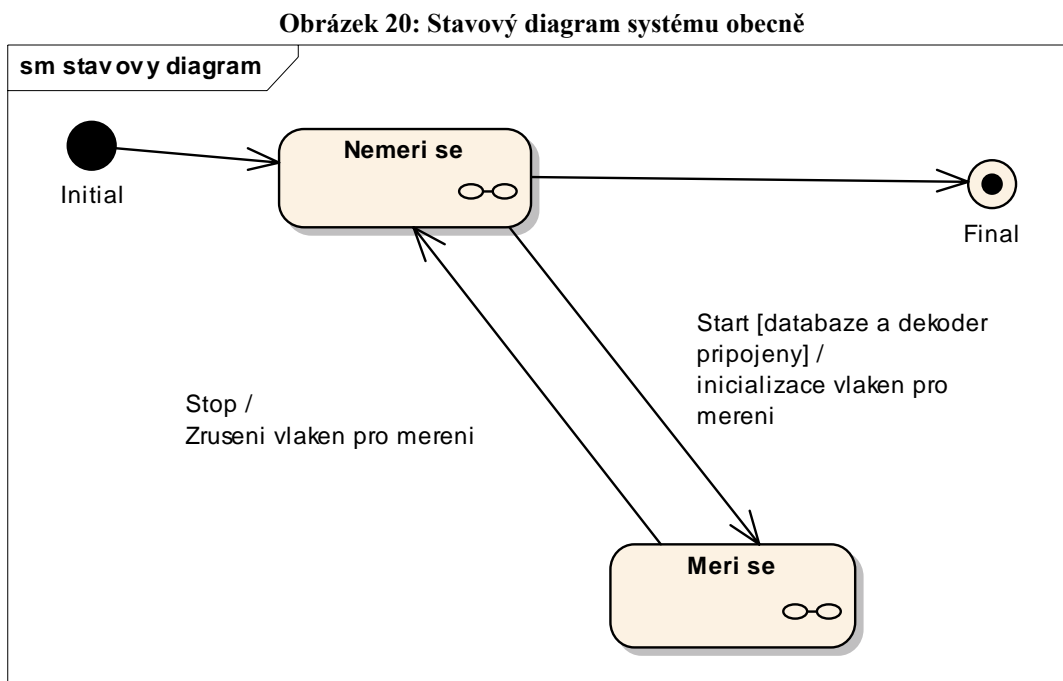
Následně dojde k uložení informací o časech průjezdu do databáze a aktualizaci informace v lokálním bufferu.

Pokud informace v lokálním bufferu není nalezena, jedná se o transpondér, který projel přes cílovou čáru poprvé a v tomto případě nelze získat žádné časové údaje tohoto transpondéru. Proto je vložena nová informace o časech prvního průjezdu do lokálního bufferu a následně dochází k uložení informace o prvním průjezdu do databáze.

5.4 Stavové diagramy

Tyto diagramy představují základní typy stavů, ve kterých se systém pro měření času může nacházet.

5.4.1 Základní stavy systému



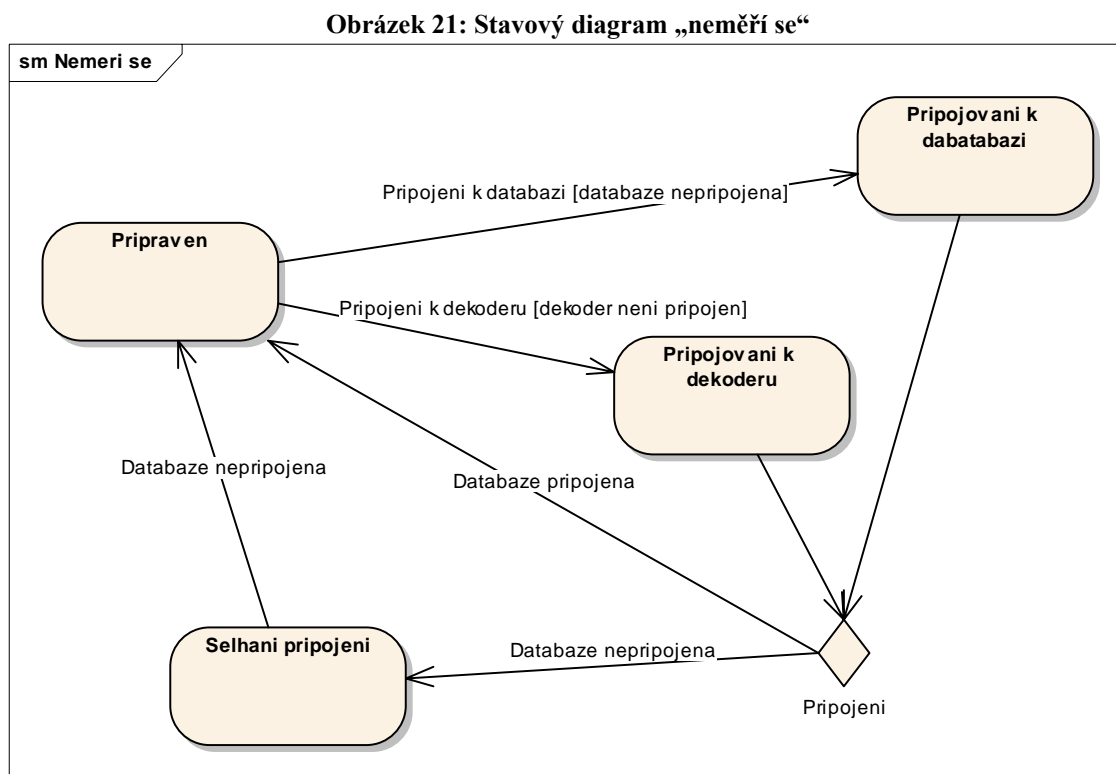
Zdroj: vlastní tvorba v prostředí Enterprise Architect

Obrázek č. 20 představuje obecný pohled na systém za pomoci stavového diagramu. Z diagramu můžeme dobře rozpoznat, že systém se může nacházet ve dvou základních stavech. Po inicializaci systému se systém dostává do prvního základního stavu „neměří se“. Tento stav v sobě na konkrétnější úrovni obsahuje další podstavy, které se dají zahrnout do stavu „neměří se“. Podrobněji se těmto stavům budeme věnovat dále v textu.

Druhým důležitým stavem je stav „měří se“, který je aktivován po události „start“. Tento přechod ze stavu „neměří se“ je podmíněn přítomností připojené databáze a dekodéru. Důsledkem tohoto přechodu je spuštění vláken, která jsou pro načítání času důležitá a starají se o načtení informací z dekodéru, dalšího zpracování a v poslední fázi uložení těchto informací do databáze.

Přechod ze stavu „měří se“ do stavu „neměří se“ je aktivován po události „stop měření“. Tento přechod ze stavu „měří se“ nemá žádnou podmínku a důsledkem tohoto přechodu je ukončení činnosti vláken, která se starala o načítání a zpracování informací z dekodéru.

5.4.2 Stavový diagram „neměří se“



Zdroj: vlastní tvorba v prostředí Enterprise Architect

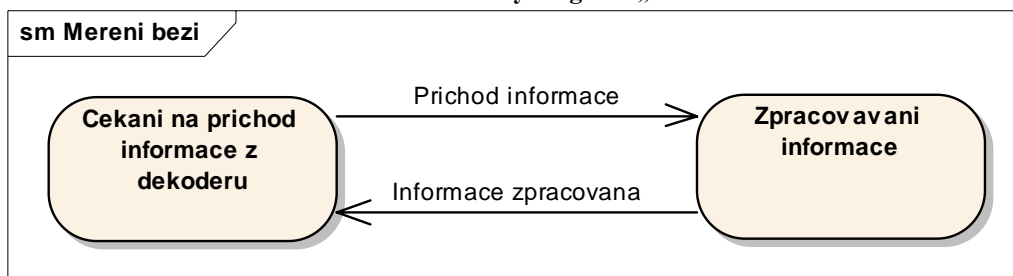
V rámci stavu systému „neměří se“ můžeme pozorovat další podstavy, do kterých se systém může dostat. Základním stavem je „připraven“, kdy systém se nachází v „klidovém“ stavu, který je představován čekáním na podněty uživatele systému.

Do stavu „připojování k databázi“ se systém může přemístit po události žádající připojení k databázi. Tento přechod je podmíněn nepřipojeností databáze. Pokud je v této fázi již databáze připojena, k tomuto přechodu nedojde. Z tohoto stavu systému je možnost přejít do dvou následujících stavů podle toho zda se systém dokáže k databázi připojit. Pokud je připojení k databázi úspěšné, přejde systém do stavu „připraven“. Pokud však nedojde k řádnému připojení k databázi, přechází systém do stavu „selhání připojení“ a následně do stavu „připraven“. Pro tento účel je v modelu naznačen rozhodovací blok pojmenovaný „připojení“.

Stejný princip přechodů je i u stavů „připraven“ → „připojování k dekodéru“ → „selhání připojení“.

5.4.3 Stavový diagram „měří se“

Obrázek 22: Stavový diagram „měří se“

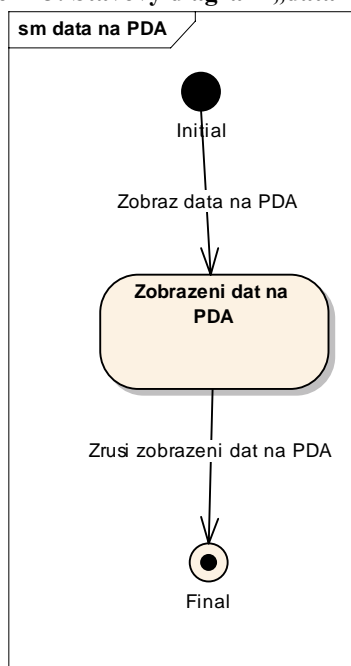


Zdroj: vlastní tvorba v prostředí Enterprise Architect

V rámci stavu systému „měří se“ můžeme obecně uvažovat o dvou stavech systému. Prvním stavem systému je „Čekání na příchod informace z dekodéru“. Při tomto stavu systém naslouchá dekodéru a čeká na informaci, kterou mu dekodér zašle. Při příchodu informace se systém přemístí do dalšího stavu, který je na obrázku č. 22 nazván „zpracování informace“. V tomto stavu systém zpracovává příchozí informaci a ukládá tuto informaci do databáze. Po vykonání operací spojených se zpracováním dat se systém opět navrací do stavu „čekání na příchod informace z dekodéru“.

5.4.4 Stavový diagram „data na PDA“

Obrázek 23: Stavový diagram „data na PDA“



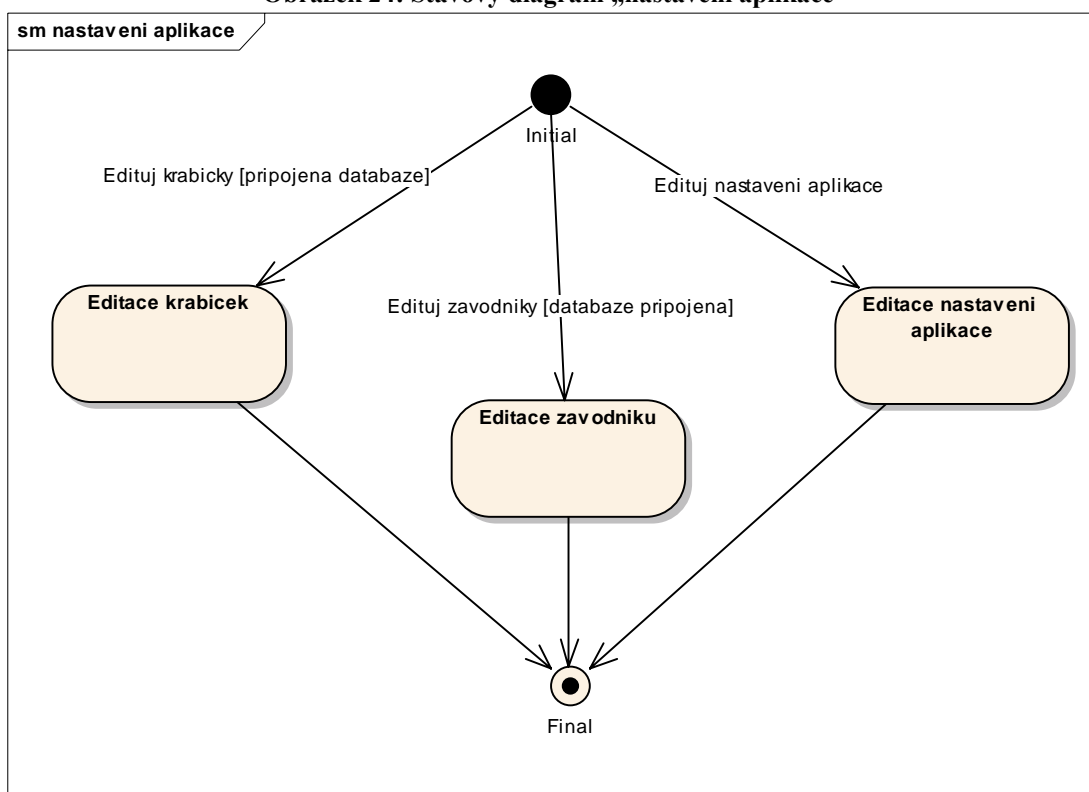
Zdroj: vlastní tvorba v prostředí Enterprise Architect

V rámci stavů, které byly dosud popsány, se systém paralelně může nacházet v dalším stavu, kterým je „zobrazování dat na PDA“. Tato možnost vychází z použití více vláken.

Do tohoto stavu se systém dostává po požadavku na zobrazení dat, která jsou zasílána na PDA. Opuštění tohoto stavu je ve chvíli ukončení požadavku na zobrazení dat. Tento stav představuje možnost uživatele zobrazit si, nezávisle na měření a načítání informací z dekodéru, data, která jsou uživateli zasílána na mobilní zařízení typu PocketPC. Uživatel má možnost kontrolovat správnost naměřených dat a jejich shodování se s daty, která vidí koncový uživatel. Koncový uživatel je uživatel, který neovládá systém. Tento uživatel si pouze zobrazuje naměřená data systémem.

5.4.5 Stavový diagram „nastavení aplikace“

Obrázek 24: Stavový diagram „nastavení aplikace“



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Dalšími paralelními stavy jsou „editace krabiček“, „editace závodníků“ a „editace nastavení aplikace“. Pro přechod do stavů „editace krabiček“ a „editace závodníků“ je podmínkou připojení k databázi.

„Editace krabiček“ je stav, ve kterém může uživatel přidávat, odebírat a editovat krabičky nadefinované v systému. Další možností uživatele je importování a exportování seznamu krabiček do a ze systému.

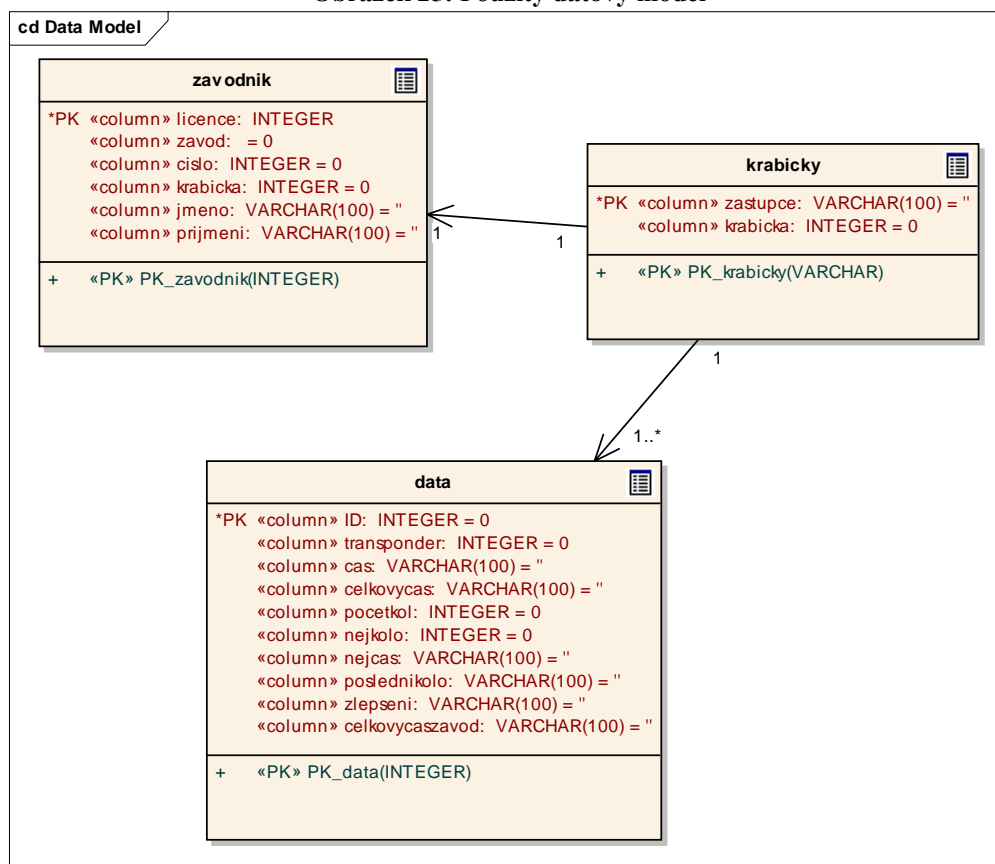
„Editace závodníků“ je stav, ve kterém může uživatel přidávat, odebírat a editovat závodníky nadefinované v systému. Další možností je, stejně jako v případě „editace krabiček“, importovat a exportovat seznam závodníků do a ze systému.

Při „editace nastavení aplikace“ má uživatel možnost nadefinovat údaje, které se budou ukládat do externího souboru a budou k dispozici při dalším spuštění aplikace. Mezi údaje, které se ukládají, patří:

- informace spojené s připojením k databázi,
- informace spojené s připojením k dekodéru,
- třídy, ve kterých závodníci jezdí,
- údaje o pořádaném závodě,
- informace spojené s nastavením fotobuňky.

5.5 Datový model

Obrázek 25: Použitý datový model



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Na obrázku č. 25 můžeme vidět datový model, který je v aplikaci použit.

Tabulka „závodník“ v sobě udržuje důležité informace týkající se závodníka. Primárním klíčem v této tabulce je licence závodníka, která je jedinečná. Každý licencovaný závodník má přiděleno své vlastní číslo licence, podle které je v celém poháru závodů motokár jedinečný. V této tabulce je cizím klíčem atribut „krabicka“, který je propojen s primárním klíčem tabulky „krabicky“.

Tabulka „krabicky“ se zdá být na první pohled trochu nadbytečnou. Ale při bližším pohledu je důležitou a pro uživatele systému zjednodušující při evidenci a práci s krabíčkami (transpondéry). Každý transpondér má své unikátní číslo, které je obvykle představováno 6-ti místným číslem. Z tohoto důvodu se v systému nepracuje s takto dlouhým číslem, ale nahrazuje se zástupným číslem nebo i číslem a znakem či znaky, která jsou kratší a se kterými je jednodušší manipulace.

Primární klíč tabulky „krabičky“ je dále používán jako cizí klíč v poslední tabulce, přesněji v tabulce „data“. Tato tabulka je primární pro ukládání zpracovaných časů získaných z dekodéru.

Data, která jsou zasílána na zařízení typu PocketPC, jsou z těchto tabulek selektována.

6 IMPLEMENTACE

6.1 Pracovní postup implementace

Implementace spočívá v převodu návrhového modelu do spustitelného kódu. Z pohledu analytika nebo návrháře je smyslem implementace tvorba požadovaného implementačního modelu. Tento model zahrnuje rozdělení návrhových tříd do komponent. Způsob, jímž je to provedeno, obvykle do velké míry závisí na volbě programovacího jazyka.

Pracovní postup implementace je zaměřen hlavně na tvorbu spustitelného kódu. Vedlejším produktem této aktivity může být implementační model, přestože tento model není výsledkem explicitní modelovací aktivity. Mnoho nástrojů CASE ve skutečnosti umožňuje provádět zpětnou analýzu implementačního modelu na základě zdrojového kódu. Proto je modelování implementace ponecháno zcela na pohledu programátorů.

6.2 Implementace systému

Celý systém je naprogramován v jazyce C# (CSharp) pomocí vývojového prostředí Visual Studio 2005 Professional Edition. K tomuto rozhodnutí vedlo hned několik pohnutek.

Jednak část informačního systému musí být schopna běžet na zařízení typu PocketPC, tedy zařízení s operačním systémem Windows Mobile 2003 a novější a také to byla snaha vytvořit systém v inovovaném programovacím jazyce na platformě .NET.

Neopomenutelnou pohnutkou je i podpora vývojového nástroje od společnosti Microsoft. Velice rozsáhlá nápověda a dostupné informace na stránkách společnosti byly neocenitelnou pomocí při vývoji systému.

Jak jsme již uvedli, celý systém byl naprogramován v prostředí .NET Framework, s ohledem na část pro mobilní zařízení, tedy i v prostředí Compact .NET Framework. Rozdíl obou prostředí je pouze v tom, že prostředí Compact .NET Framework je „chudší“ upravenou verzí prostředí .NET Framework.

Pro připojení k databázi jsme použili ovladače MySQL .NET connector 1.0.7 a MySQLDirect. Bohužel nebylo možné použít stejný ovladač. MySQL .NET connector 1.0.7 nepodporuje práci v prostředí Compact .NET Framework. Pro komunikace s dekodérem byla

využita příručka k danému zařízení, která definuje formáty, ve kterých dekodér zasílá informace a také, na které adrese a kterém portu se dekodér hlásí. Formát komunikace je uveden v příloze č. 2 „Specifikace protokolů k dekodéru“.

Celý systém se skládá ze dvou hlavních programů a dvou vedlejších podpůrných programů.

- Hlavní programy
 - TimeKeeper
 - TimeKeeper pro PDA
- Podpůrné programy
 - Ping status
 - Simulátor dekodéru

6.2.1 TimeKeeper

Lze říci, že tento program je jádrem celého systému. Uživatel se přihlásí k dekodéru a databázi a následně má možnost editovat krabíčky, závodníky, jízdy a v neposlední řadě také začínat a končit měření jízdy.

Jeho funkce jsou popsány v příloze č. 3 „Uživatelská příručka“.

6.2.2 TimeKeeper pro PDA

Tento program je nainstalován na uživatelském mobilním zařízení typu PocketPC. Má funkci zobrazování online výsledků jízdy v uživatelsky přívětivé podobě.

Jeho funkce jsou popsány v příloze č. 3 „Uživatelská příručka“.

6.2.3 Ping status

Tento program je podporou pro zjišťování stavu sítě. Dobrý stav a funkčnost sítě je v tomto informačním systému prioritní. Pokud by došlo k výpadku nějakého síťového uzlu, nebudou si moci uživatelé zobrazit aktuální výsledky jízdy.

Pro tento případ je tu tato malá utilita, do které se nastaví sledované IP adresy a ona v definovaných cyklech kontroluje dostupnost těchto definovaných IP adres. Při špatné funkci by se zobrazila zpráva a obsluha systému by mohla aktuální situaci řešit.

6.2.4 Simulátor dekodéru

Při vytváření systému jsme narazili na velice důležitý a zásadní problém. Dekodér je velice specifické zařízení. Toto zařízení je velice drahé a z tohoto důvodu bylo Autoklubem České republiky zakoupeno jen pár kusů. Z příčiny malého počtu dekodérů a velkého množství závodů, které se za sezónu pořádají, je velice složité toto zařízení půjčit a použít pro testování systému. Z tohoto důvodu jsme vytvořili utilitu, která dokáže tento dekodér zastoupit.

Tomuto programu lze nastavit data, která má vysílat dekodér. Následně ho aktivovat a simulátor se chová pro systém jako opravdový dekodér.

Využití této utility je jednak při testování, ale také při rekonstrukci nějaké jízdy a řešení případných problémů.

7 NASAZENÍ

7.1 Pracovní postup nasazení

7.1.1 Komponenta

Komponenta je „fyzickou nahraditelnou součástí systému, která obaluje implementaci a poskytuje realizaci množiny specifikovaných rozhraní“. Komponenty reprezentují hmatatelné fyzické věci.

Komponenta je jednotkou znovupoužitelného kódu a má velmi širokou definici. Každá komponenta může obsahovat mnoho tříd a realizovat velké množství rozhraní.

7.1.2 Diagram nasazení

Diagram nasazení ukazuje nejen fyzický hardware, na němž bude softwarový systém spuštěn, ale i způsob, jímž je software na tomto hardwaru nasazen

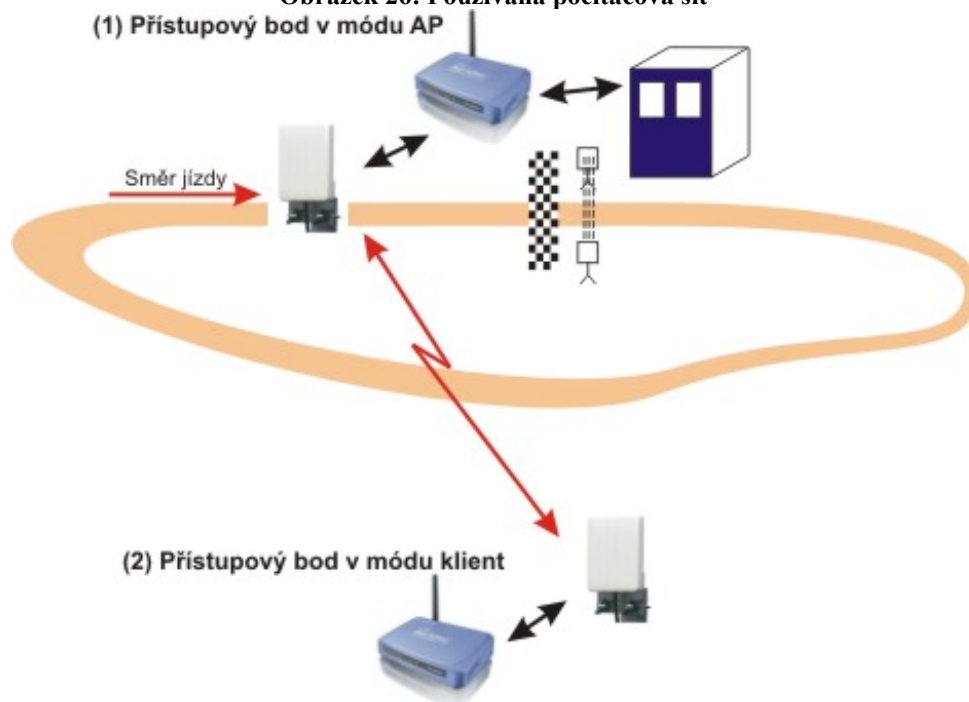
Existují dvě verze diagramu nasazení:

- **diagram nasazení** – obsahuje komponenty, uzly a vztahy mezi jednotlivými uzly. Uzel reprezentuje typ hardwaru. Komponenta zastupuje typ softwaru,
- **diagram konkrétního nasazení** – obsahuje instance uzlů, relace mezi nimi a instance komponent. Instance uzlu zastupuje specifickou identifikovatelnou část hardwaru. Instance komponenty zastupuje specifickou část softwaru.

7.2 Hardware

Pro naplnění požadavků pro informační systém bylo nutné vybudovat počítačovou síť. Tato počítačová síť musí splňovat požadavek univerzálnosti. Závody se pořádají na různých okruzích po celé České republice a nelze vybudovat síť „šitou na míru“ pouze jednomu závodisti. Dalším důležitým kritériem je mobilita. Zařízení musí být skladné a jednoduše přepravitelné. Systém není na každém závodisti pevně umístěn. Veškerá zařízení po závodech cestují.

Obrázek 26: Používaná počítačová síť



Zdroj: vlastní tvorba

V první fázi myšlenek o vybudování sítě jsme uvažovali o využití stávajících prostředků a zařízení, které se do té doby používaly. Na obrázku č.26 můžeme vidět stávající stav sítě, která se používala.

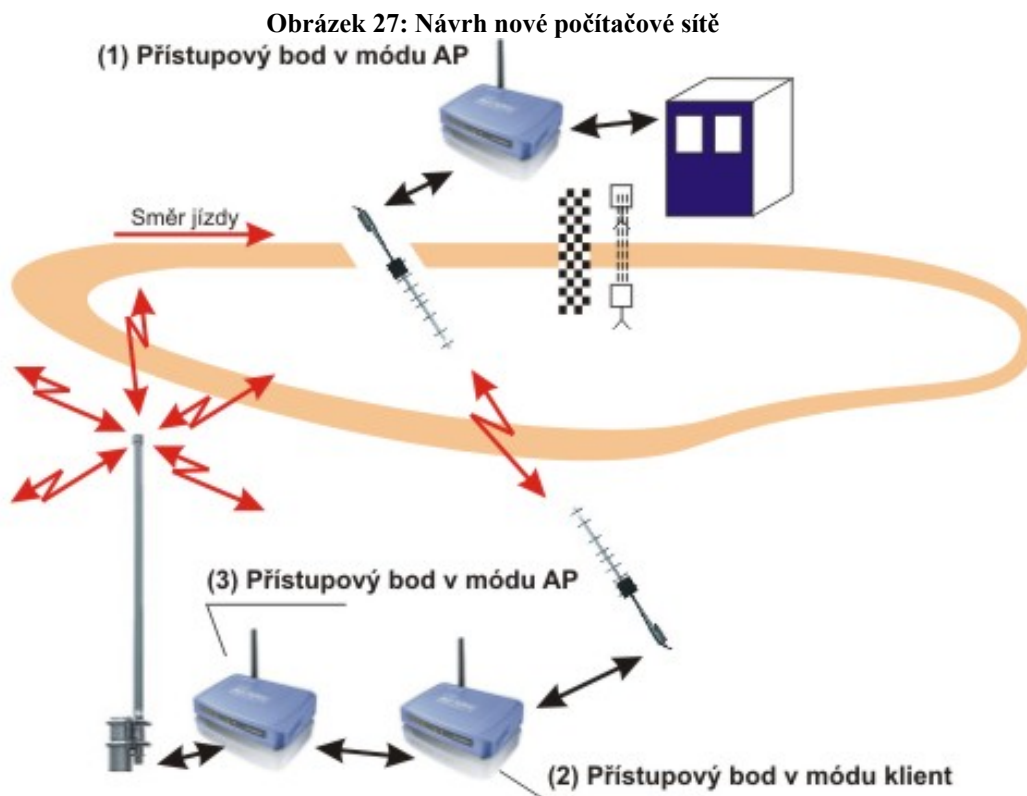
V rámci budovy časoměřičů se používá síť typu hvězda s centrálním prvkem inteligentního přepínače (switch), ke kterému jsou připojeny počítače a dekodér za pomoci kabeláže. Z daného přepínače je dalším kabelem připojen přístupový bod (Access Point), který je nastaven v módu AP¹³ s možností dynamického přidělování adres a připojen k panelové anténě. Tato anténa je nasměrována na další panelovou anténu, která je spojena s druhým přístupovým bodem, který je v módu klient¹⁴.

Již z obrázku č. 26 je tedy dostatečně zřetelné, že síť pro účely informačního systému nebyla dostatečná. Bohužel se nedalo uvažovat ani o rozšíření stávajícího stavu. Na obrázku č. 26 vidíme, že byly použity panelové antény a ty již při stávajícím stavu nebyly vyhovující. Dále je patrné, že síť nebyla připravena pro použití širokou uživatelskou veřejností. Jednalo se pouze o bezdrátové propojení jednoho PC, případně několika PC propojených kabeláží, s věží časoměřičů. Dalším kritériem je výkonnost používaných antén. Síť je provozována nad závodní dráhou a stroje jezdící po okruhu byly velice často zdrojem rušení.

¹³ Access Point, tedy mód pro vysílání a hlášení se jako přístupový bod

¹⁴ Klient, tedy mód pro přijímání signálu a chování se jako klientské zařízení

Z těchto důvodů jsme zahájili zcela nový návrh sítě. Tato síť bude jednak sloužit pro informační systém okruhových závodů, ale je zde i možnost na rozšíření pro jiné informační systémy. Na obrázku č. 27 můžeme vidět návrh nové počítačové sítě.



Zdroj: vlastní tvorba

V budově centra časoměřičů je umístěn první přístupový bod, který je v módu AP. Tento prvek je kabelem připojen do sítě v budově časoměřičů. Jeho název je TimeKeeper časomíra a nemá zapnuté přidělování IP adres. Slouží pouze jako zprostředkovatel spojení k dalšímu přístupovému bodu, který je nazván „TimeKeeper přijímač“. K oběma zařízením je připojena směrová anténa s výkonem 12 dB. Druhý přístupový bod je nastaven v módu klient a je napojen na první přístupový bod, se kterým komunikuje. Na všech závodištích je dodržena přímá viditelnost obou zařízením a vzdálenost nepřesahuje hodnotu 500 metrů. Při zavedení nebylo do současné doby pozorováno zhoršení signálu zapříčiněné závodními stroji jezdců.

K přístupovému bodu „TimeKeeper přijímač“ je následně kabelem připojen třetí přístupový bod nazvaný jako „TimeKeeper vysílač“, který za pomoci připojené všesměrové 9,5 dB antény poskytuje kvalitní pokrytí celého areálu závodiště. Při testování se zařízeními typu PocketPC nebyl zjištěn žádný závažný problém s nedostatkem signálu a to i přes velice malý výkon přijímací antény na zmiňovaném zařízení.

Poslední zmiňovaný přístupový bod nazvaný příhodně „TimeKeeper vysílač“ je nastaven do módu AP a má zapnuté přidělování IP adres v rozmezí 192.168.1.150 – 200. Tyto adresy jsou vyhrazeny pro zařízení z řad závodníků a uživatelů tratě.

Dalším vedlejším produktem bylo umožnění větší mobility v rámci pracovního prostoru časoměřičů. Výkonná směrová anténa umístěná u prvního přístupového bodu na věži časoměřičů poskytuje dostatečný výkon i v ostatních směrech a časoměřiči se mohou na tento přístupový bod přihlásit a využívat síť za pomoci bezdrátové technologie.

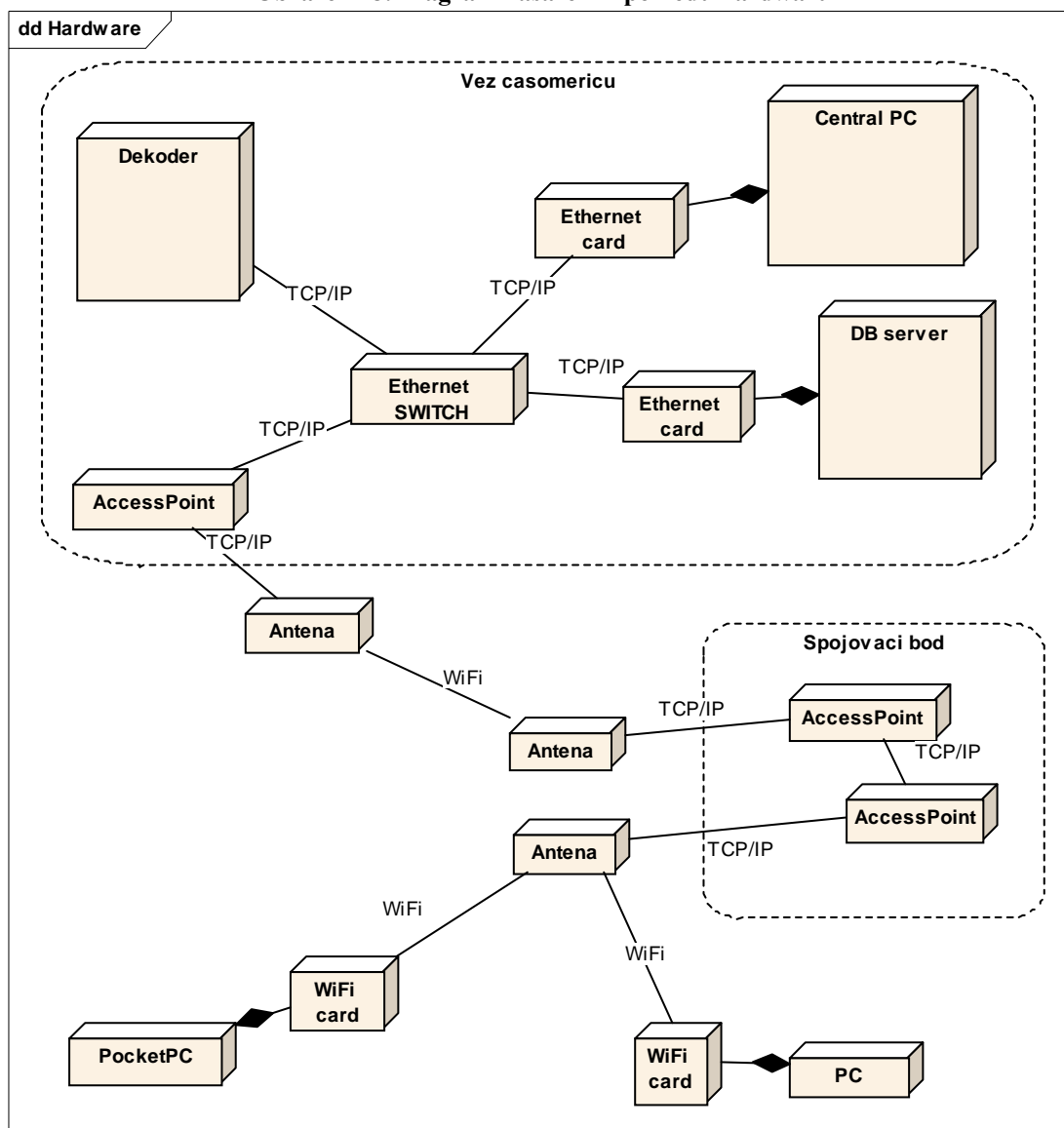
Dalším důležitým prvkem v návrhu počítačové sítě bylo zajištění ochrany zařízení před vlivem počasí. Jelikož je první přístupový bod umístěn na věži časoměřičů nebyla potřeba žádných zásadních opatření. Jediným kritickým bodem byla spojka kabelu, který vede od směrové antény k přístupovému bodu. Zde byla jednoduše použita elektrikářská páska pro izolování od vlhkosti.

Důležité místo vhodné pro ochranu zařízení se stal spojovací bod s druhým a třetím přístupovým bodem. Zde se z důvodů ochrany použila standardní elektrikářská rozvodná skříň, která byla k účelům tohoto síťového bodu upravena. Na spodní straně skříně byly vyvrtány otvory pro přivedení kabelu obou antén a napájecího kabelu. Tyto otvory byly utěsněny. Posledním otvorem, který byl na spodní straně vytvořen, se stala větrací mřížka, která se stará o zamezení držení se vlhkosti uvnitř skříně.

Posledním bodem vybudování sítě je její zabezpečení. V současnosti je zabezpečení stále ve fázi návrhu. Je nutné zabezpečit počítačovou síť časoměřičů od zbytku uživatelů, kteří se do sítě přihlásí. Počítače časoměřičů však v současnosti přímo žádná data nesdílejí. Jejich komunikace probíhá za pomoci FTP serverů, kde je přístup omezen na uživatele znající přístupové jméno a heslo. Veškerá komunikace je zabezpečena ověřením přístupových informací. Tato situace není v žádném případě optimální a navrhuje se systém nového zabezpečení sítě.

7.2.1 Diagram nasazení z pohledu hardware

Obrázek 28: Diagram nasazení z pohledu hardwaru



Zdroj: vlastní tvorba v prostředí Enterprise Architect

Z obrázku č. 28 můžeme vidět rozdělení sítě na část uvnitř věže časoměřičů a následně na část spojovacího uzlu, který obsahuje dva přístupové body.

Centrem celé sítě je aktivní přepínač, který se stará o komunikaci mezi všemi zařízeními připojenými do sítě. Celá síť je navržena jako hvězda, tedy výpadek jednoho zařízení v síti nijak neohrozí chod ostatních článků sítě.

7.3 Software

V celém systému jsou využívány produkty od společnosti Microsoft. Počínaje operačním systémem Windows Professional a konče Windows Mobile 2003. Dále celý

system běží v prostředí NET Framework od jmenované společnosti. Využita byla databáze MySQL.

7.3.1 Komponenty

V této kapitole bychom se velice rádi zaměřili na nejdůležitější uzly celého systému a jejich používané komponenty. U každého uzlu bude uvedena nejprve jeho hardwarová specifikace a následně použité komponenty.

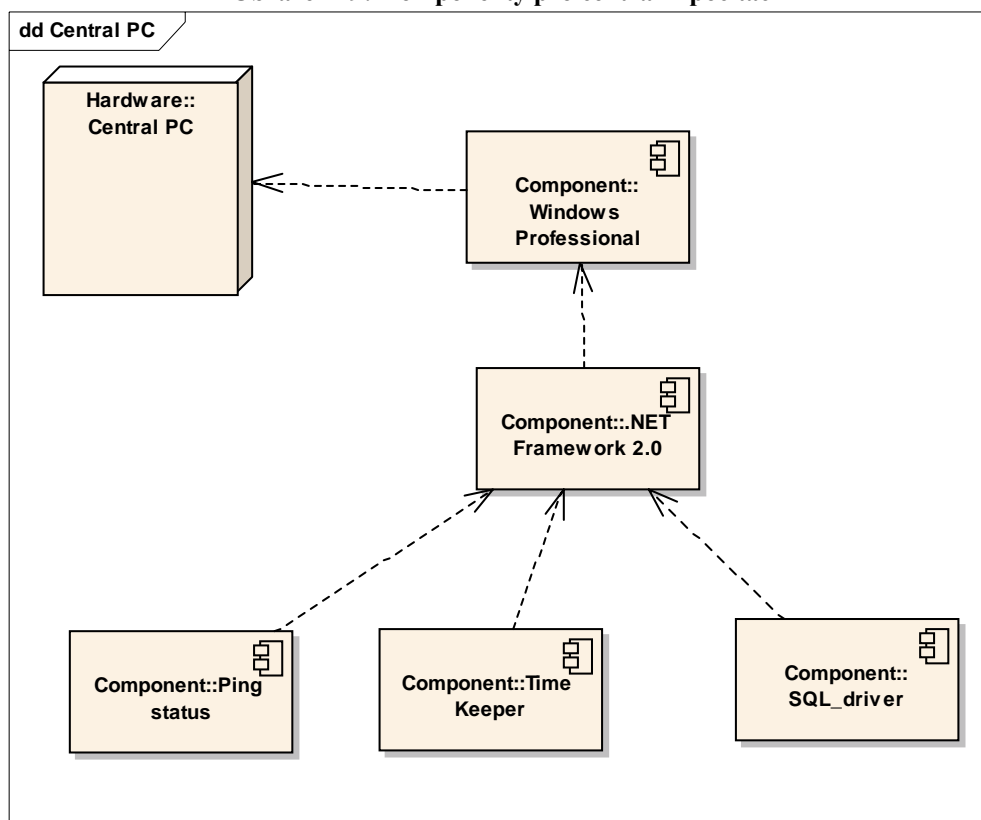
Centrální počítač

Hardware

CPU	2,4 GHz AMD Athlon XP mobile
Paměť	512 MB RAM
HDD	40 GB

Komponenty

Obrázek 29: Komponenty pro centrální počítač



Zdroj: vlastní tvorba v prostředí Enterprise Architect

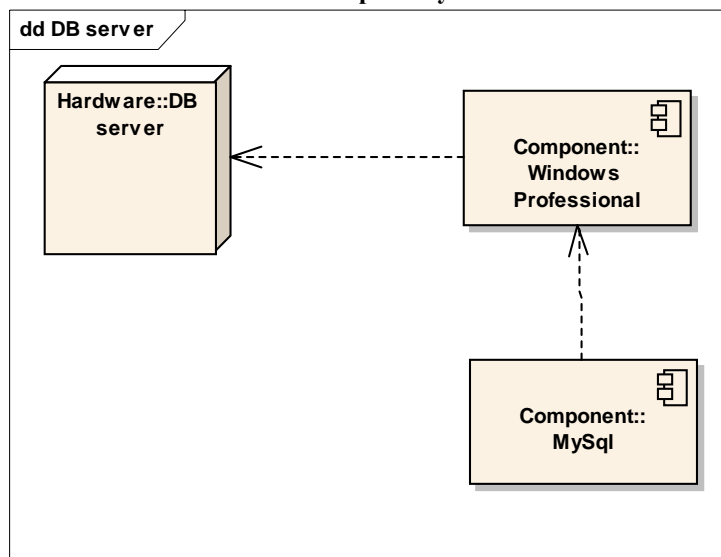
Databázový server

Hardware

CPU	2,4 GHz AMD Athlon XP mobile
Paměť	512 MB RAM
HDD	40 GB

Komponenty

Obrázek 30: Komponenty DB serveru



Zdroj: vlastní tvorba v prostředí Enterprise Architekt

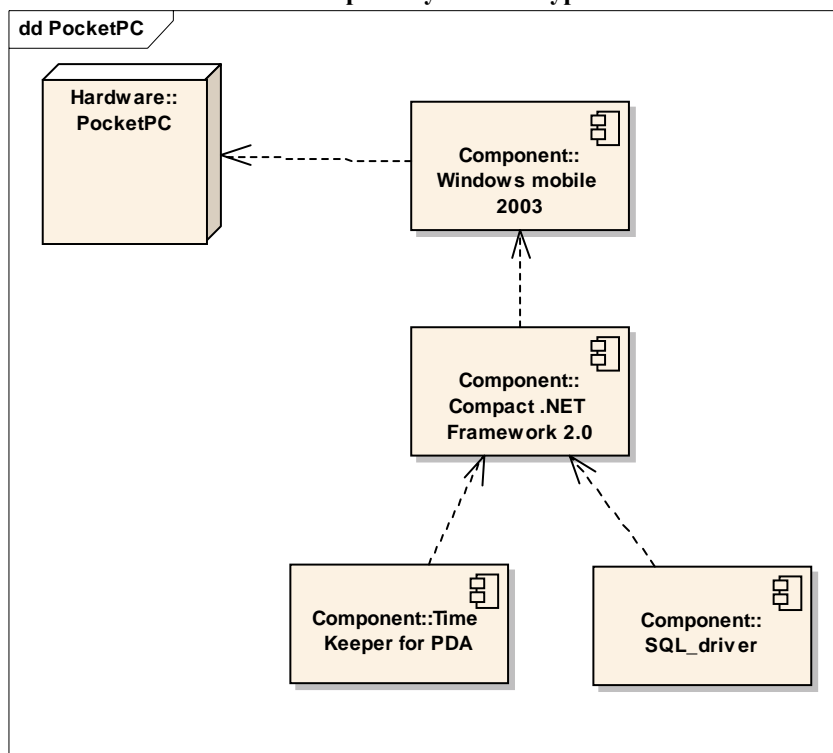
Zařízení PocketPC

Hardware

CPU	312 MHz PXA272
Paměť	64 ROM, 64 RAM, 1 GB externí SD karta

Komponenty

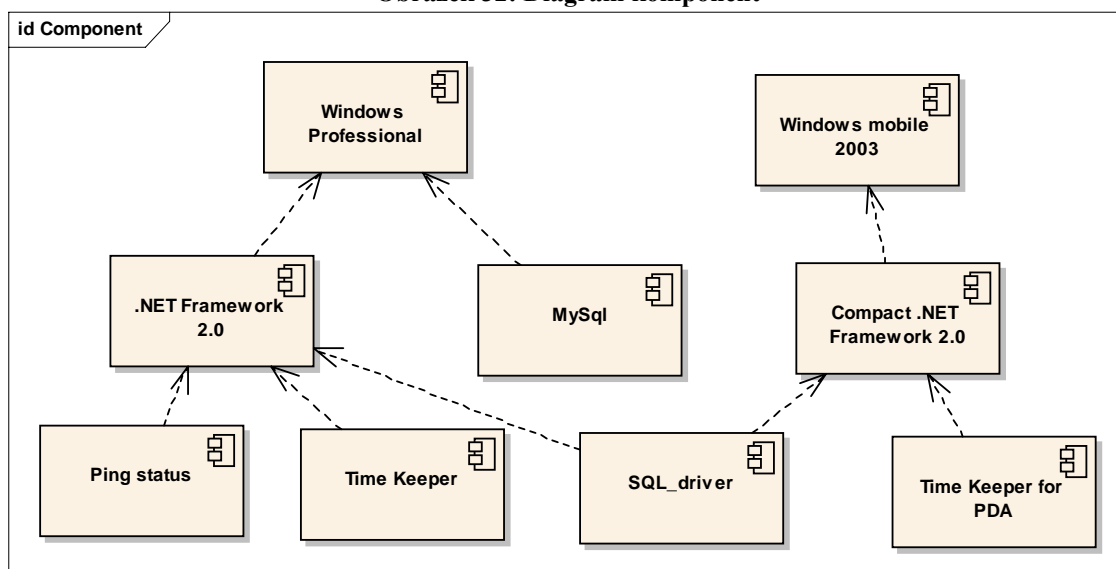
Obrázek 31: Komponenty zařízení typu PocketPC



Zdroj: vlastní tvorba v prostředí Enterprise Architekt

7.3.2 Diagram komponent

Obrázek 32: Diagram komponent



Zdroj: vlastní tvorba v prostředí Enterprise Architekt

8 TESTOVÁNÍ

8.1 Pracovní postup testování

Vytvořený systém byl podroben praktickým testům a zkouškám. Byl prakticky uplatněn na dvou závodech. Data byla porovnávána se systémem Orbits, který se v České republice prioritně používá pro měření času okruhových závodů motokár.

Systém byl použit na Mistrovství České republiky v České Lípě a na závodech Hobby Cup ve Vysokém Mýtě.

8.1.1 Mistrovství České republiky Česká Lípa

Informace o závodě

- *Místo*
 - Autodrom Sosnová u České Lípy
- *Datum*
 - 29. 4 – 30. 4. 2006
- *Stav počasí*
 - První den závodu bylo počasí střídavě oblačné a zamračené. V odpoledních hodinách přišly přeháňky. Druhý den bylo v ranních hodinách zamračeno, ale po polední se obloha vyjasnila a bylo krásné slunné odpoledne. Teploty o víkendu se v sobotu pohybovaly kolem 17 °C a v neděli stouply k 20 °C.
- *Počet uživatelů využívajících systém*
 - 6

Kategorie, počty jezdců a počty jízd

<i>Kategorie</i>	<i>Počet jezdců</i>	<i>Počet jízd celkem</i>
Kadet	14	6
Bambiny	11	6
NJ a NA	11	6
ROK CZ	19	6
ROK junior	15	6
ICC 125	19	6

Výsledky testování

Tabulka 1: Výsledky testování MČR Česká Lípa

Kategorie	Jízda 1.	Jízda 2.	Jízda 3.	Jízda 4.	Jízda 5.	Jízda 6.
Kadet	Ok	Ok	Ok	Ok	Ok	Ok
Bambiny	Ok	Ok	Ok	Ok	Ok	Ok
NJ a NA	Ok	Ok	Ok	Ok	Error	Ok
ROK CZ	Ok	Ok	Ok	Error	Ok	Ok
ROK junior	Ok	Ok	Ok	Ok	Ok	Ok
ICC 125	Ok	Ok	Ok	Ok	Ok	Ok

Zdroj: Vlastní tvorba

Popis tabulky: Ok – při této jízdě systém fungoval bez problému, Error – při této jízdě došlo k potížím ve funkci systému

Nároky systému na hardware zařízení

- *Měření času neběží*
 - CPU 0 %
- *Měření času běží*
 - CPU 90 %
 - Využití sítě 0,06 %

Shrnutí

Systém se při prvním nasazení velice dobře zaběhl. I ve chvílích, kdy časoměřič je nucen provádět a sledovat několik činností najednou, není uživatelské ovládání programu nijak zatěžující.

Po konzultacích s uživateli, kteří tuto první verzi vyzkoušeli, je možné konstatovat, že se shledala s převážně kladným ohlasem.

Zjištěné problémy a návrh na jejich odstranění:

- **neošetřená výjimka MySQL exception.** Chyba byla zjištěna přímo při průběhu závodu a byla dodatečně odstraněna,
- **vytížení procesoru.** Chyba byla lokalizována a dodatečně odstraněna.

8.1.2 Hobby Cup Vysoké Mýto

Informace o závodě

- *Místo*
 - Vysoké Mýto
- *Datum*
 - 6. 5. – 7. 5. 2006
- *Stav počasí*
 - Celý víkend bylo krásné počasí.
- *Počet uživatelů využívajících systém*
 - 10

Kategorie, počty jezdců a počty jízd

<i>Kategorie</i>	<i>Počet jezdců</i>	<i>Počet jízd celkem</i>
Mladí	15	6
Kadet a Comer	10	6
CZ a WF	10	6
FA 100	16	6
Junior	18	6
ROK	15	6
FC senior	13	6
FC junior	18	6
Honda 390	10	6

Výsledky testování

Tabulka 2: Výsledky testování Hobby Cup Vysoké Mýto

Kategorie	Jízda 1.	Jízda 2.	Jízda 3.	Jízda 4.	Jízda 5.	Jízda 6.
Mladí	Ok	Ok	Ok	Ok	Ok	Ok
Kadet a Comer	Ok	Ok	Ok	Ok	Ok	Ok
CZ a WF	Ok	Ok	Ok	Ok	Ok	Ok
FA 100	Ok	Ok	Ok	Ok	Ok	Ok
Junior	Ok	Ok	Ok	Ok	Ok	Ok
ROK	Ok	Ok	Ok	Ok	Ok	Ok
FC senior	Ok	Ok	Ok	Ok	Ok	Ok
FC junior	Ok	Ok	Ok	Ok	Ok	Ok
Honda	Ok	Ok	Ok	Ok	Ok	Ok

Zdroj: Vlastní tvorba

Popis tabulky: Ok – při této jízdě systém fungoval bez problému

Nároky systému na hardware zařízení

- *Měření času neběží*
 - CPU 0 %
- *Měření času běží*
 - CPU 4 %
 - Využití sítě 0,06 %

Shrnutí

Po odstranění problémů zjištěných při prvním použití systému v praxi, se u druhého nasazení neprojevily žádné problémy. Celý závod proběhl bez zjištěných nedostatků

Zjištěné problémy a návrh jejich odstranění

Nebyly zjištěny.

8.2 Shrnutí pracovního postupu testování

Při celkovém hodnocení testování lze dle výsledků dvou závodů usoudit, že systém se jeví stabilním a dobře ovladatelným. V příloze č. 4 „Mistrovství České republiky Česká Lípa“ jsem uvedl příklad srovnání naměřených výsledků informačního systému a programu Orbits. Dále v příloze č. 5 „Výstup z programu TimeKeeper“ je uveden export jízdy systému.

Toto srovnání je pro další použití programu velice důležité. Program Orbits je považován za velice výkonný nástroj pro měření času a je oficiálně používán Autoklubem České republiky.

Dle získaných porovnání za dva závody se dá říci, že informační systém zpracovává a generuje výsledky totožné s výstupy programu Orbits.

Závěr

Tato diplomová práce je zaměřena na tvorbu informačního systému pro okruhové závody. Pod pojmem okruhový závod se rozumí takový závod, kde start a cíl závodu se nachází v jednom místě závodní dráhy. V České republice se pořádá mnoho druhů okruhových závodů. Mezi ty nejznámější patří závody motocyklové, automobilové a v neposlední řadě také závody motokár.

Cílem diplomové práce se stalo uspokojení požadavků týmů závodů motokár na vytvoření nového informačního systému, který by doplnil stávající skladbu softwarových prostředků pro měření časů těchto závodů. Do současné doby neexistoval systém, který by dokázal informovat týmy o probíhající jízdě na zařízení typu PocketPC. Uživatelským primárním požadavkem je software, který zpřístupní online výsledky probíhajícího závodu. Tedy přesněji informace o všech jezdcích, jejich nejlepším kole, počtu kol, čase posledního kola a další důležité informace.

V první fázi jsem se v kapitole nazvané „Požadavky“ zaměřil na zpracování požadavků uživatelů a správné pochopení problémové domény měření časů závodů motokár. Tato činnost v sobě obsahovala komunikaci s uživateli a praktické seznámení s měřením času.

V návrhové části jsem zpracoval informace týkající se požadavků budoucích uživatelů tohoto systému. V této fázi jsem obecné požadavky rozpracoval do konkrétní podoby jednotlivých tříd a vztahů těchto tříd. Výsledkem této činnosti je takový návrh tříd, podle kterých je možno programově vytvořit základní kostru informačního systému s jednotlivými třídami, atributy a metodami těchto tříd.

V předposlední fázi tvorby systému, která je popsána v kapitolách „implementace“ a „nasazení“ je prakticky naprogramován celý informační systém okruhových závodů.

V poslední fázi této diplomové práce jsem se zaměřil na testování celého systému v praktických podmínkách dvou závodů motokár. Tato činnost je popsána v kapitole „testování“. Při praktickém použití bylo zjištěno několik nedostatků, které byly následně odstraněny. Tyto nedostatky bylo možno odhalit až při fungování systému v rámci praktického průběhu daného závodu. Po této fázi se dá konstatovat, že systém funguje stabilně a uživateli je hodnocen kladně.

Cíl této diplomové práce, tedy tvorba informačního systému, byl splněn.

Tento systém je dle mého názoru prakticky použitelný pro online informování uživatelů o probíhající jízdě. Je možno ho zavést při závodech motokár a obecně při každých závodech, kde je měření založeno na transpondérech a dekodéru. Jelikož se toto měření v současné době používá téměř při všech okruhových závodech je možno tento informační systém využít na každém takovém závodě.

Tento systém je možno dále rozšiřovat. Jelikož bylo naprogramováno jádro pro měření času, je možno vytvořit další rozšiřující moduly, které do systému zahrnou další činnosti vykonávané při závodech.

Seznam literatury

- [1] ROBINSON, S.; ALLEN, K.S.; CORNES, C.; GLYNN, J.; GREENVOSS, Z.; HARVEY, B.; NAGEL, CH.; SKINNER, M.; WATSON, K. *C# Programujeme profesionálně*. Brno: Computer Press, 2003. ISBN 80-251-0085-5.
- [2] ARLOW, J.; NEUSTADT, I. *UML a unifikovaný proces vývoje aplikací*. Brno: Computer Press, 2003. ISBN 80-7226-947-X.
- [3] *Nápověda k vývojovému prostředí Visual Studio 2005*. Dostupný na WWW: <<http://msdn.microsoft.com>>.
- [4] *Ovladač pro MySQL databázi*. Dostupný na WWW: <<http://www.crlab.com>>.
- [5] *Nápověda k MySQL databázi*. Dostupný na WWW: <<http://dev.mysql.com>>.
- [6] *Ročenka kartingu*. Dostupný na WWW: <<http://www.autoklub.cz>>.
- [7] *Hardware a software komponenty od společnosti AMB*. Dostupný na WWW: <<http://www.amb-it.com>>.
- [8] *Kart systém*. Dostupný na WWW: <<http://www.kart-data.com>>.
- [9] *PID systém*. Dostupný na WWW: <<http://www.pid-system.com>>.
- [10] *Alfano*. Dostupný na WWW: <<http://www.alfano.be>>.
- [11] *TAG Heuer photocells*. Dostupný na WWW: <<http://www.tagheuer-timing.com>>.
- [12] *Motokárový svět*. Dostupný na WWW: <<http://www.motokary.cz>>.

Seznam tabulek

Tabulka 1: Výsledky testování MČR Česká Lípa	72
Tabulka 2: Výsledky testování Hobby Cup Vysoké Mýto.....	73

Seznam obrázků

Obrázek 1: Okruh	12
Obrázek 2: Umístění transpondéru na závodním stroji	13
Obrázek 3: Umístění komponent pro měření času	15
Obrázek 4: Orbits systém	18
Obrázek 5: Alfano systém	20
Obrázek 6: Kart data systém.....	21
Obrázek 7: TimeKeeper systém	23
Obrázek 8: Use Case diagram	36
Obrázek 9: Sekvenční diagram přihlášení	38
Obrázek 10: Sekvenční diagram editace krabičky	39
Obrázek 11: Sekvenční diagram editace závodníků.....	41
Obrázek 12: Sekvenční diagram definice složek a jízd.....	42
Obrázek 13: Sekvenční diagram měření jízdy.....	43
Obrázek 14: Sekvenční diagram vyhodnocení jízdy	43
Obrázek 15: Sekvenční diagram import a export dat	44
Obrázek 16: Návrhové třídy systému	48
Obrázek 17: Diagram aktivit pro vstupní vlákno	49
Obrázek 18: Diagram aktivit pro výstupní vlákno	50
Obrázek 19: Diagram aktivit zpracuj data.....	51
Obrázek 20: Stavový diagram systému obecně.....	53
Obrázek 21: Stavový diagram „neměří se“	54
Obrázek 22: Stavový diagram „měří se“	55
Obrázek 23: Stavový diagram „data na PDA“	55
Obrázek 24: Stavový diagram „nastavení aplikace“	56
Obrázek 25: Použitý datový model	58
Obrázek 26: Používaná počítačová síť	64
Obrázek 27: Návrh nové počítačové sítě.....	65
Obrázek 28: Diagram nasazení z pohledu hardwaru	67
Obrázek 29: Komponenty pro centrální počítač.....	68
Obrázek 30: Komponenty DB serveru	69
Obrázek 31: Komponenty zařízení typu PocketPC	70
Obrázek 32: Diagram komponent	70

Seznam příloh

Příloha č. 1 – Komponenty pro měření času a jejich technická specifikace

Příloha č. 2 – Specifikace protokolů k dekodéru

Příloha č. 3 – Uživatelská příručka

Příloha č. 4 – Mistrovství České republiky Česká Lípa

Příloha č. 5 – Výstup z programu TimeKeeper

Příloha č. 6 – Používané vlastní utility

Komponenty pro měření času a jejich technická specifikace

- **Hlavní cílová smyčka**



Smyčka od firmy AMB

Šířka závodní dráhy: max 20m (66 ft)

Délka koaxiálního kabelu: max 100 m (330 ft)

- **Záložní cílová smyčka**

Jsou dvě alternativy použití záložní cílové smyčky. Jednou možností je použití stejné smyčky jako u hlavní cílové smyčky. Druhou možností je použití optické závory.



Fotobuňka a hodiny od firmy TAG heuer

Fotobuňka:

Napájení: 6 – 12 V

Šířka závodní dráhy: max 20 m.

Technologie: IrDA

Hodiny:

Provoz: • -20° C až +50° C

Napájení: 12 V z adaptéru nebo baterií

Paměť: 8 000 průjezdů



- **Alfano smyčka**



Smyčka od firmy Alfano

Šířka závodní dráhy: max 20m (66 ft)

Délka koaxiálního kabelu: max 100 m (330 ft)

- **Dekódovací zařízení**



Paměť: 25000 průjezdů

Přesnost: 0.001s

Provoz: -10 – 60 °C (14 – 140 °F)

Napájení: 12 VDC via 110v/230v AC adapter

Výstup: RS232, 10/100 Base T (TCP/IP)

- **Systém pro zpracování dat**

Data jsou zpracovávána na standardním stolním počítači nebo na notebooku.

- **Prezentace online výsledků jízdy**

Podle systému měření se data posílají na standardní stolní počítač, notebook, PDA.

- **Transpondér**



Max. rychlost: 160 km/h / 100 mph

Pozice: max. výška 30 cm / 1 ft

Váha: 95 g

Výdrž: min. 4 dny

Nabíjení: 16 hodin pro plné nabití

- **AccesPoint**



Zařízení je možné konfigurovat webovým prohlížečem nebo SNMP managementem. K samořejmým funkcím patří podpora módů klient, bridge a repeater.

- **Směrová anténa**

Směrová anténa 12 dBi, konektor N female



- **Všesměrová anténa**

Všesměrová anténa pro pásmo 2,4 GHz s vertikální polarizací. Anténa je určena k montáži na stožár. Zisk 12 dBi, konektor N-female



Specifikace protokolů k dekodéru

Úvod

The AMB i.t. decoder has a serial communication port (RS232) and an Ethernet port (10Base-T).

Serial communication port (RS232)

The serial port uses the following RS232 output format : 9600 baud, 8 databits, 1 stopbit, no parity, no handshake. The enhanced protocol is used.

Ethernet port (10Base-T)

The Ethernet port supports the well known TCP/IP protocol. The decoder acts as a server and will listen for TCP connections at port 5100. The maximum number of connections that will be served is 4. The TCP connections use the enhanced mode protocol records. The enhanced protocol is described in the following paragraphs.

The simplest way to connect to a decoder is using a 'telnet' client, which is usually a terminal emulator like 'Hyperterminal'. When using Hyperterminal create a New Connection with options TCP/IP (Winsock), supply the IP host-address of the decoder and 5100 as port number. This assumes the IP address and Subnet Mask are set by using the RS232 port and the appropriate commands.

Some of the enhanced protocol state information like reference time and stored passings indexes is kept unique per TCP connection. Actually for the RS232 link this state information is kept unique also. This means that various settings can be done on a connection not influencing other ports. This allows various clients like timekeeper applications and loggers to be served independently. Be aware, however, that some commands like 'Reset' have impact on all ports (including the RS232 port).

GPS

A Global Positioning System (GPS) receiver can be attached to the decoder on its auxiliary port. When the GPS receiver has calculated an accurate time and position, the decoder will be synchronized to that time and the status record will output the position.

Základní struktura protokolu

The following pages describe the general structure of the enhanced protocol used by the AMB i.t. decoder.

<SOR><Command><Sep><Param1><Sep><Param2>...<sep><ParamN><sep><CRC><EOR>

where:

Item	Name	Range	Value
<SOR>	Start of Record		0x01
<Command>	Command	0x21..0x7e	
<Sep>	Separator	[; or , or 0x09]	
<ParamX>	Parameter X	(Notes 5, 6, 7)	
<CRC>	Cyclic Redundancy Check	[0 .. 2 ¹⁶ - 1]	
<EOR>	End of Record		0x0D 0x0A

Notes:

- <SOR> is the ASCII control character 0x01 (^A).
- <Command> is a single (printable) character.
- <Sep> is either a semicolon, a comma or a tab. A decoder will always use a tab.
- <EOR> are the two characters CR & LF (0x0D 0x0A).
- Parameters can either be decimal or hexadecimal integer values or decimal floating-point values.
- Hexadecimal values are preceded by a 'x' or 'X' and are unsigned. Floating-point values have a decimal point after the unit digit. Examples:

-12345	decimal negative number
xE35DA	hexadecimal unsigned number
12.34	positive floating point number
- The first two parameters will be used for decoder ID and record sequence numbering.
- The CRC is calculated using the ITU CRC16 function over all characters following the <SOR> up and until the last separator (the underlined part). See Appendix B.
- CRC can be missing (<EOR> follows last <sep> directly).
- The decoder will always send a CRC. Commands to the decoder do not need a CRC; however, they need the last separator otherwise the last parameter will be regarded as CRC. If a CRC is not supplied a command will be accepted as is; corruption of the command could result in unwanted results.
- <Command>s used are: '@' for a passing, '#' for a status record, '?' for commands to the decoder and '\$' for any other records originating from the decoder. Generally a decoder will echo a '\$' record as a response to a '?' record with the same parameter structure.
- 'Decoder ID' and 'Record seq. num.' need not be supplied for a '?' command to the decoder. Separators need still be given to indicate skipped values.

Záznam o průjezdu

AMBrc	AMB 130	TranX-2	TranXPro	Elite	PeqaSys
-------	---------	---------	----------	-------	---------

The passing record is sent after a transponder has been received and the passing time has been calculated.

The passing record uses:

Item	Type	Range	Usage
<Command>	<char>	'@'	Passing
<Param1>	<integer>	[1...254]	Decoder ID
<Param2>	<integer>	[0...4294967295]	Record seq. num.
<Param3>	<integer>	[1...4294967295]	Transponder number
<Param4>	<float>	[0...28147497671]	Time in seconds
<Param5>	<integer>	[2...65535]	Number of hits
<Param6>	<integer>	[0...255]	Signal strength
<Param7>	<integer>	[0...255]	Passing status
<Param8>	<string>	[CCC-00000 ... ZZZ-99999]	Transponder code

Notes:

- Time is represented by a floating-point format with seconds as unit. Decimal point and decimals are added as necessary according to actual time value (e.g. 12345.67); any leading or trailing zeroes are suppressed. The range extends over many years.
- Time is kept separate per connection. The effect is that each client can and will have its own time base.
- Time is represented in seconds since January 1, 1970 (UTC) once the decoder is synchronized to the GPS.
- Decoder ID is a fixed decoder setting to identify multiple decoders in a timing system with intermediate point timing lines. 254 different decoder ID's are possible.
- Transponder numbers:
 - 9991 = Photocell 1
 - 9992 = Photocell 2 / External Start
 - 9993 = Sync input
 Hits and strength are meaningless here and can be missing.
- Number of hits (successful receptions) and signal strength are an indication of the performance of the combination of transponder power and transponder and loop placement related signal weakening.
- The low-battery value is the least significant bit of Passing status field and is normally 0 and will be 1 with a passing of a transponder with a low battery charge left.
- The Loop Number is coded in the 3 bits left to the low-battery bit. The loop number can be 1 or 3 (Loop inputs 1 and 2 are combined on extended loop decoders).

Future use	Loop number	Low Battery
b7..b4	b3..b1	b0
0000	001..111	0 or 1

- Transponder code for the newer generations AMB transponders with alphanumeric code printed on the transponder. For other transponders this parameter will be omitted. **First digit is optional and may not be always available.**
To convert transponder number to a transponder code, please refer to appendix D.

Záznam o stavu zařízení

The status record is issued about every 5 seconds.

Status record uses:

AMBrc	AMB 130	TranX-2	TranXPro	Elite	PegaSys
-------	---------	---------	----------	-------	---------

Item	Type	Range	Usage
<Command>	<char>	`#`	Status
<Param1>	<integer>	[1...254]	Decoder ID
<Param2>	<integer>	[0...4294967295]	Record seq. num.
<Param3>	<integer>	[0...255]	Noise value *
<Param4>	<integer>	[0...255]	Status *
<Param5>	<float>	[-180.0000...+179.9999]	Latitude
<Param6>	<float>	[-180.0000...+179.9999]	Longitude

AMBrc	AMB 130	TranX-2	TranXPro	Elite	PegaSys
-------	---------	---------	----------	-------	---------

Item	Type	Range	Usage
<Command>	<char>	`#`	Status
<Param1>	<integer>	[1...254]	Decoder ID
<Param2>	<integer>	[0...4294967295]	Record seq. num.
<Param3>	<integer>	[0...255]	Noise value *
<Param4>	<integer>	[0...255]	Not used
<Param5>	<integer>	[0...100]	Temperature in C
<Param6>	<float>	[0.00...19.99]	Battery voltage
<Param7>	<integer>	[0...255]	Status *
<Param8>	<float>	[-180.0000...+179.9999]	Latitude
<Param9>	<float>	[-180.0000...+179.9999]	Longitude

Notes:

- The noise value gives an indication of the received noise level. The noise level should be related to the strength recorded in passings.
- Grayed params will only be send when GPS receiver is detected and the decoder is synchronized to the GPS time.
- The GPS Status contains flag to indicate the status of the GPS receiver.

b0	AC powered
b1	Charging
b2	GPS synchronized
b3...b7	Future use
- The GPS receiver is typically used when multiple decoders need to be synchronized to the exact same time.

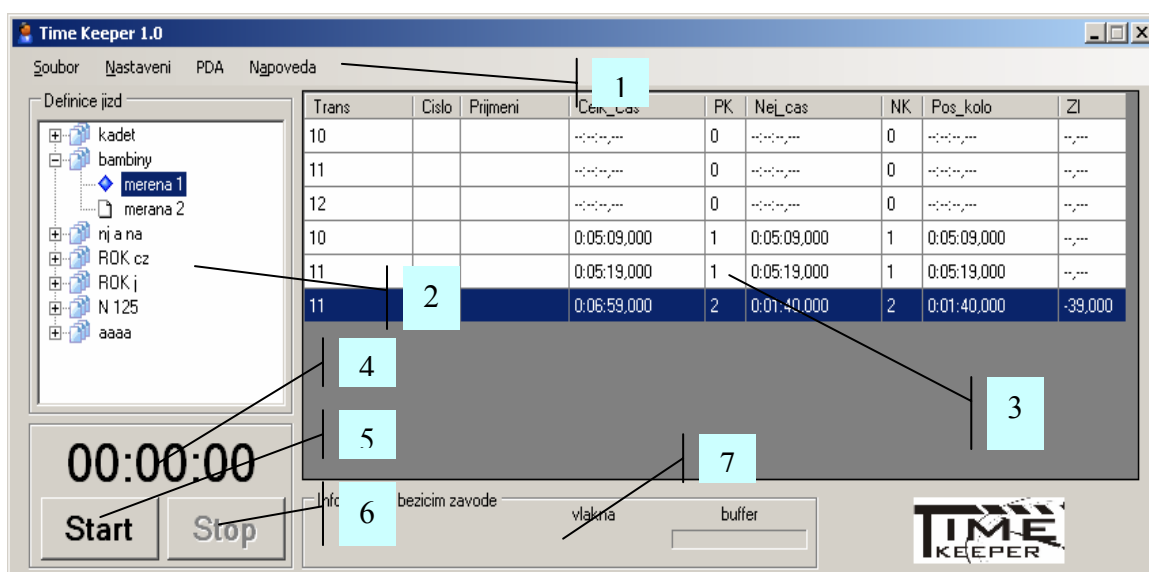
Uživatelská příručka

TimeKeeper

Instalace

- V první fázi musí být na uživatelský počítač nainstalováno prostředí .NET Framework 2.0, které je nutné pro spuštění a provozování celé aplikace.
- V druhé fázi si uživatel na svém počítači vytvoří složku. Umístění této složky je na uvážení daného uživatele. Do této složky je nutné zkopírovat:
 - Aplikaci TimeKeeper – soubor TimeKeeper.exe
 - Knihovnu MySQL.Data.dll

Popis prostředí programu



Legenda:

1. Hlavní menu program
Hlavní rozcestník programu. Jednotlivé položky viz výklad dále.
2. Definice jízdy
Strom definovaných kategorií a jednotlivých dílčích jízd. Kliknutím na symbol + resp. – se rozbalí resp. sbalí příslušná kategorie. Kliknutím pravým tlačítkem myši na příslušnou položku se zobrazí místní nabídka, která nabízí vložení/odebrání kategorie resp. jízdy.
3. Tabulka, kde se zobrazují načítaná data
V této tabulce se zobrazují jednotlivé zaznamenané průjezdy. První záznam v tabulce představuje první průjezd a poslední záznam poslední průjezd.
4. Časomíra
Pro kontrolu délky trvání dané jízdy je zde vyobrazen čas. Daný čas běží od začátku jízdy až po její konec, kdy se nuluje.
5. Tlačítko Start
Tlačítko start je aktivní pouze při vybrání nějaké konkrétní jízdy a funguje za předpokladu, že program je připojen k databázi a dekodéru. Toto tlačítko startuje načítání jednotlivých průjezdů, jejich zpracování a vyhodnocování.

6. Tlačítko Stop

Tlačítko stop ukončuje aktuálně běžící jízdu. Uživateli je nabídnut export právě ukončené jízdy pro zpětné nahlédnutí a další práci s daty.

7. Informace o probíhajícím závodě.

Tyto informace se aktivují po stisknutí tlačítka Start a deaktivují při ukončení jízdy tlačítkem Stop. Na tomto panelu se zobrazují dva druhy informací:

- i. Status vlákna
- ii. Stav vyrovnávací paměti (bufferu)

Hlavní menu programu



Struktura hlavního menu

- Soubor
 - Připojení
 - Databaze
 - Dekoderu
 - Konec
- Nastavení
 - Databaze
 - Krabicky
 - Zavodnici
 - Možnosti
- PDA
 - Mereny trenink
 - Bodovana jizda
- Napoveda

Funkce nabídek hlavního menu

- Soubor→Připojení→Databaze

Možnost připojení k databázi. Je nutné nadefinovat IP adresu serveru, uživatelské jméno a heslo pro připojení k databázovému serveru. Implicitně se jedná o připojení k MySQL databázi.

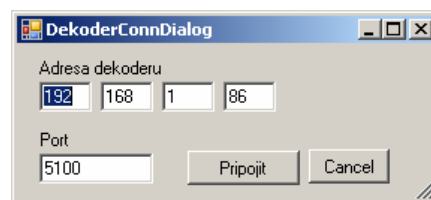
Tato operace je potvrzena zprávou pro uživatele a v hlavním menu je tato položka zaškrtnuta.



- Soubor→Připojení→Dekoderu

Možnost připojení k dekodéru. Je nutné zadat IP adresu dekodéru a port na který se uživatel chce připojit. Implicitně se jedná o dekodér od firmy AMB.

Tato operace je potvrzena zprávou pro uživatele a v hlavním menu je tato položka zaškrtnuta.



- Soubor→Konec

Ukončí se běh programu. Pro zabezpečení neúmyslného ukončení programu je uživatel na tuto operaci upozorněn a dotázán zda opravdu tuto operaci chce vykonat.

- Nastavení→Databaze→Krabicky

Po kliknutí na tuto nabídku je uživateli otevřeno editační okno pro možnost definování, editování a odebírání krabiček (transpondérů). Vše je doplněno o možnost importu a exportu seznamu krabiček.

Zastupne cislo	Cislo krabicky
1111	1111
1	12343
3	32554
4	43234

Editace dat

Zastupny nazev: 1111

Cislo transponderu: 1111

Buttons: Edituj zaznam, Cancel

- Nastavení→Databaze→Zavodnici

Po kliknutí na tuto nabídku je uživateli otevřeno editační okno pro možnost definování, editování a odebírání závodníků. Vše je doplněno o možnost importu a exportu seznamu závodníků

Licence	Cislo	Krabicka	Jmeno	Prijmeni
2	12	11	Paclt	Radek
3	33	12	Neterda	Tomas
5	44	10	Novak	Petr
6	4	433	Deda	Jiri

Editace dat

Cislo licence: 5, Prijmeni: Petr, Jmeno: Novak

Cislo: 44, Kategorie: prvni kategorie

Krabicka: 10

Buttons: Edituj zaznam, Cancel

- Nastavení→Moznosti

Propracovaný systém nastavení aplikace. Uživatel může definovat nastavení pro dekodér, databázi, kategorie závodníků, informace o závodě a mnoho dalších parametrů, které se do aplikace ukládají. Při případném dalším spuštění se načítají a uživatel již není nucen všechny údaje vyplňovat opětovně.

Níže jsou vyobrazeny jednotlivé druhy nastavení aplikace.

Nastavení programu

Databaze Dekoder Kategorie Info zavod Fotobunka

Adresa serveru

Přihlasovací jméno

Heslo

Automaticky přihlasovat po spuštění?

Uložit změny Zavřít

Nastavení programu

Databaze Dekoder Kategorie Info zavod Fotobunka

Adresa dekoderu

Číslo portu

Automaticky přihlasovat po spuštění?

Uložit změny Zavřít

Nastavení programu

Databaze Dekoder Kategorie Info zavod Fotobunka

Kategorie	Číslo
Kadet	1
ROK CZ	2
ROK junior	3
ICC 125	4

Uložit změny Zavřít

Nastavení programu

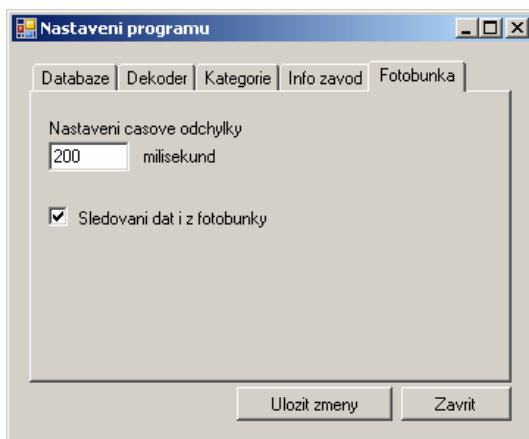
Databaze Dekoder Kategorie Info zavod Fotobunka

Název pořádaného závodu

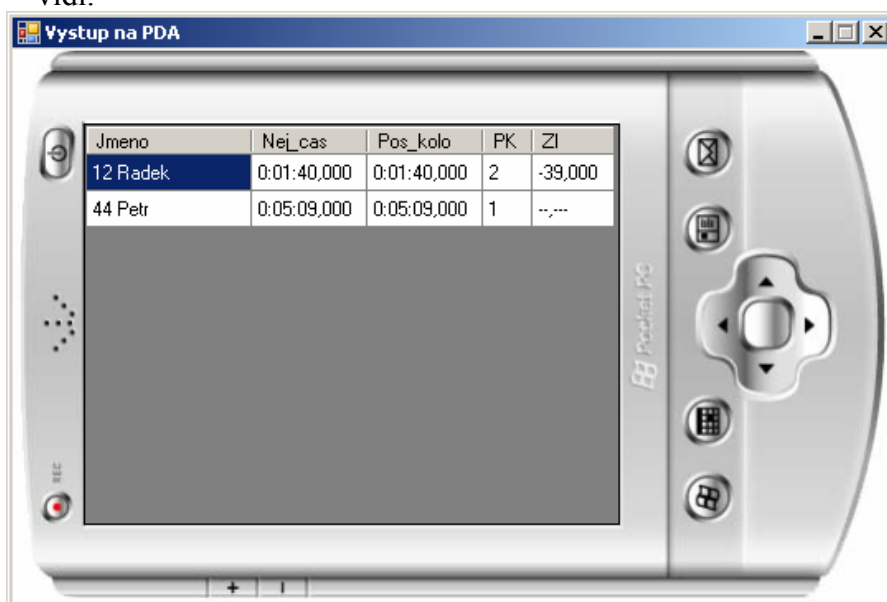
Místo pořádání

Datum konání

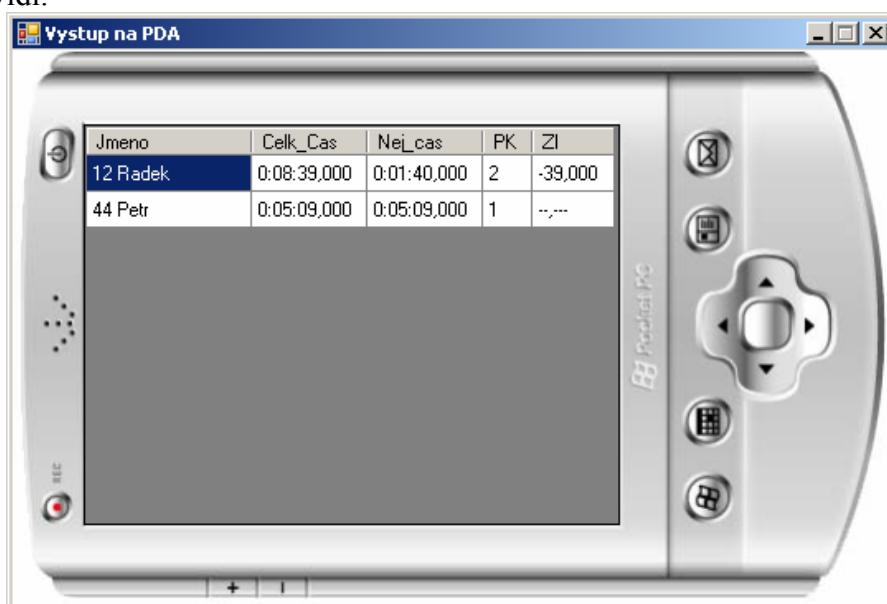
Uložit změny Zavřít



- PDA→Měření trénink
 Jde o náhled informací, které jsou zasílány na PDA a které uživatel v PDA vidí.



- PDA→Bodovaná jízda
 Jde o náhled informací, které jsou zasílány na PDA a které uživatel v PDA vidí.



- Napoveda
Programová nápověda

TimeKeeper pro PDA (PocketPC zařízení)

Instalace

- V první fázi musí být na uživatelské zařízení nainstalováno prostředí Compact .NET Framework 2.0, které je nutné pro spuštění a provozování celé aplikace.
- V druhé fázi si uživatel na svém zařízení vytvoří složku, umístění této složky je na uvážení daného uživatele. Do této složky je nutné zkopírovat:
 - Aplikaci TimeKeeper pro PDA – TimeKeeper_pro_PDA.exe
 - Knihovnu MySQL.Data.dll

Hlavní menu programu

Struktura hlavního menu

- Soubor
 - Připojení
 - Odpojení
 - Jízda
 - Merena
 - Bodovana
 - Konec

Funkce nabídek hlavního menu

- Soubor→Připojení
Uživatel se připojí k databázi. Pro připojení je nutné nastavit:
 - IP adresu serveru
 - Uživatelské jméno
 - Heslo
- Soubor→Odpojení
Odpojení od databáze.
- Soubor→Jízda→Merena
Při zvolení druhu jízdy se upraví formát stahovaných dat.
- Soubor→Jízda→Bodovana
Při zvolení druhu jízdy se upraví formát stahovaných dat.
- Soubor→Konec
Ukončí se chod aplikace. U zařízení typu PDA je nutné upozornit na možnost, že aplikace stále běží a není vypnuta. Bližší informace viz. uživatelská příručka PDA.

Mistrovství České republiky Česká Lípa

Výsledky prvního měřeného tréninku kategorie ROK junior

Výstup z programu Orbits

<i>Startovní číslo</i>	<i>Nejlepší čas</i>
52	56,634
99	57,219
72	57,580
100	57,879
...	...

Výstup z programu TimeKeeper

<i>Startovní číslo</i>	<i>Nejlepší čas</i>
52	56,634
99	57,219
72	57,580
100	57,879
...	...

Poznámka

Informace z obou výstupů jsou upraveny a naformátovány pro snadné a rychlé posouzení podobnosti dat.

Výstup z programu TimeKeeper

Výstup z prvního měřeného tréninku ROK junior

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
166826	66	a	--:--:--	0	--:--:--	0	--:--:--	--:--
9991			--:--:--	0	--:~:~:~	0	--:~:~:~	--:~:~
1558576	51	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:01,3	1	00:01,3	1	00:01,3	--:~:~
1691303	99	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:05,1	2	00:01,3	1	00:03,8	2,548
941606	72	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:29,1	3	00:01,3	1	00:24,0	22,707
1266400	92	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
105283	52	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:32,0	4	00:01,3	1	00:02,9	1,609
241601	67	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:34,1	5	00:01,3	1	00:02,1	0,825
432936	55	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:37,0	6	00:01,3	1	00:02,8	1,561
1113482	91	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:44,4	7	00:01,3	1	00:07,4	6,16
1248877	94	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			00:54,6	8	00:01,3	1	00:10,1	8,86
166826	66	a	01:11,0	1	01:11,0	1	01:11,0	--:~:~
9991			00:58,6	9	00:01,3	1	00:04,1	2,763
536130	88	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			01:04,7	10	00:01,3	1	00:06,1	4,766
32563	100	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			01:09,4	11	00:01,3	1	00:04,7	3,397
941606	72	a	01:04,2	1	01:04,2	1	01:04,2	--:~:~
9991			01:09,8	12	00:00,4	12	00:00,4	-0,868
1691303	99	a	01:08,5	1	01:08,5	1	01:08,5	--:~:~
9991			01:10,4	13	00:00,4	12	00:00,6	0,22
1558576	51	a	01:10,4	1	01:10,4	1	01:10,4	--:~:~
9991			01:29,2	14	00:00,4	12	00:18,8	18,34
105283	52	a	01:00,0	1	01:00,0	1	01:00,0	--:~:~
9991			01:34,6	15	00:00,4	12	00:05,4	5,014
1266400	92	a	01:05,5	1	01:05,5	1	01:05,5	--:~:~
9991			01:37,8	16	00:00,4	12	00:03,2	2,746
241601	67	a	01:05,8	1	01:05,8	1	01:05,8	--:~:~
9991			01:38,4	17	00:00,4	12	00:00,6	0,163
432936	55	a	01:04,2	1	01:04,2	1	01:04,2	--:~:~
9991			01:39,9	18	00:00,4	12	00:01,6	1,141
1113482	91	a	01:02,9	1	01:02,9	1	01:02,9	--:~:~
9991			01:50,1	19	00:00,4	12	00:10,1	9,713
258440	68	a	--:~:~:~	0	--:~:~:~	0	--:~:~:~	--:~:~
9991			01:52,2	20	00:00,4	12	00:02,1	1,721
1248877	94	a	01:07,8	1	01:07,8	1	01:07,8	--:~:~

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
9991			01:54,6	21	00:00,4	12	00:02,4	1,977
312042	50	a	--:--:--	0	--:--:--	0	--:--:--	--:--
9991			01:59,0	22	00:00,4	12	00:04,4	4,016
166826	66	a	02:15,4	2	01:04,4	2	01:04,4	-6,547
9991			02:02,0	23	00:00,4	12	00:02,9	2,52
536130	88	a	01:03,3	1	01:03,3	1	01:03,3	--:--
9991			02:06,7	24	00:00,4	12	00:04,8	4,351
32563	100	a	01:02,1	1	01:02,1	1	01:02,1	--:--
9991			02:08,4	25	00:00,4	12	00:01,7	1,265
9991			02:08,8	26	00:00,4	26	00:00,4	-0,038
941606	72	a	02:03,3	2	00:59,1	2	00:59,1	-5,163
1691303	99	a	02:07,5	2	00:59,0	2	00:59,0	-9,444
9991			02:12,4	27	00:00,4	26	00:03,6	3,19
1558576	51	a	02:12,4	2	01:01,9	2	01:01,9	-8,466
9991			02:26,9	28	00:00,4	26	00:14,5	14,142
105283	52	a	01:57,7	2	00:57,7	2	00:57,7	-2,292
9991			02:36,4	29	00:00,4	26	00:09,5	9,152
1266400	92	a	02:07,3	2	01:01,8	2	01:01,8	-3,677
9991			02:40,6	30	00:00,4	26	00:04,2	3,796
1113482	91	a	02:03,6	2	01:00,7	2	01:00,7	-2,231
9991			02:41,5	31	00:00,4	26	00:00,9	0,487
432936	55	a	02:07,3	2	01:03,1	2	01:03,1	-1,09
9991			02:41,9	32	00:00,4	26	00:00,5	0,07
241601	67	a	02:09,9	2	01:04,2	2	01:04,2	-1,603
9991			02:50,9	33	00:00,4	26	00:09,0	8,597
258440	68	a	01:00,9	1	01:00,9	1	01:00,9	--:--
9991			02:54,1	34	00:00,4	26	00:03,1	2,757
1248877	94	a	02:09,6	2	01:01,8	2	01:01,8	-5,932
9991			02:58,1	35	00:00,4	26	00:04,1	3,674
312042	50	a	01:03,5	1	01:03,5	1	01:03,5	--:--
9991			02:59,1	36	00:00,4	26	00:00,9	0,559
166826	66	a	03:15,5	3	01:00,0	3	01:00,0	-4,422
9991			03:01,3	37	00:00,4	26	00:02,2	1,849
536130	88	a	02:02,7	2	00:59,3	2	00:59,3	-4,031
9991			03:06,9	38	00:00,4	26	00:05,6	5,252
9991			03:07,3	39	00:00,4	39	00:00,4	-0,019
941606	72	a	03:01,8	3	00:58,5	3	00:58,5	-0,582
1691303	99	a	03:05,9	3	00:58,5	3	00:58,5	-0,551
9991			03:08,2	40	00:00,4	39	00:00,9	0,521
32563	100	a	02:03,5	2	01:01,4	2	01:01,4	-0,652
9991			03:11,6	41	00:00,4	39	00:03,4	3,083
1558576	51	a	03:11,6	3	00:59,2	3	00:59,2	-2,718
9991			03:14,9	42	00:00,4	39	00:03,3	2,887
1802633	77	a	--:--:--	0	--:--:--	0	--:--:--	--:--
9991			03:24,1	43	00:00,4	39	00:09,3	8,887
105283	52	a	02:54,9	3	00:57,2	3	00:57,2	-0,52
9991			03:37,3	44	00:00,4	39	00:13,2	12,859
1266400	92	a	03:08,2	3	01:00,9	3	01:00,9	-0,925
9991			03:39,6	45	00:00,4	39	00:02,2	1,882
1113482	91	a	03:02,6	3	00:59,0	3	00:59,0	-1,729
9991			03:41,6	46	00:00,4	39	00:02,1	1,691

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
432936	55	a	03:07,5	3	01:00,1	3	01:00,1	-2,976
9991			03:42,8	47	00:00,4	39	00:01,2	0,826
241601	67	a	03:10,8	3	01:00,9	3	01:00,9	-3,275
9991			03:49,8	48	00:00,4	39	00:07,0	6,646
258440	68	a	01:59,8	2	00:58,9	2	00:58,9	-1,944
9991			03:59,0	49	00:00,4	39	00:09,2	8,821
166826	66	a	04:15,4	4	01:00,0	4	01:00,0	-0,058
9991			04:00,5	50	00:00,4	39	00:01,5	1,096
9991			04:00,6	51	00:00,1	51	00:00,1	-0,225
9991			04:00,8	52	00:00,1	51	00:00,1	0,004
1248877	94	a	03:16,0	3	01:01,8	2	01:06,4	4,593
312042	50	a	02:06,0	2	01:02,5	2	01:02,5	-1,022
536130	88	a	03:02,1	3	00:59,3	2	00:59,5	0,151
9991			04:05,4	53	00:00,1	51	00:04,6	4,474
941606	72	a	04:00,2	4	00:58,5	4	00:58,5	-0,03
9991			04:07,1	54	00:00,1	51	00:01,8	1,633
32563	100	a	03:02,4	3	00:59,0	3	00:59,0	-2,438
9991			04:08,2	55	00:00,1	51	00:01,1	0,924
1691303	99	a	04:06,9	4	00:58,5	3	01:00,9	2,45
9991			04:10,0	56	00:00,1	51	00:01,8	1,666
1558576	51	a	04:10,0	4	00:58,4	4	00:58,4	-0,851
9991			04:17,6	57	00:00,1	51	00:07,6	7,428
1802633	77	a	01:02,7	1	01:02,7	1	01:02,7	---
9991			04:20,8	58	00:00,1	51	00:03,2	3,042
105283	52	a	03:51,6	4	00:56,6	4	00:56,6	-0,571
9991			04:37,1	59	00:00,1	51	00:16,4	16,252
1266400	92	a	04:08,0	4	00:59,8	4	00:59,8	-1,091
9991			04:38,2	60	00:00,1	51	00:01,1	0,949
1113482	91	a	04:01,2	4	00:58,6	4	00:58,6	-0,322
9991			04:48,9	61	00:00,1	51	00:10,7	10,541
258440	68	a	02:58,8	3	00:58,9	2	00:59,1	0,16
9991			04:50,4	62	00:00,1	51	00:01,5	1,365
432936	55	a	04:16,3	4	01:00,1	3	01:08,8	8,631
9991			04:59,3	63	00:00,1	51	00:08,9	8,727
9991			04:59,7	64	00:00,1	51	00:00,4	0,282
536130	88	a	04:00,6	4	00:58,5	4	00:58,5	-0,79
166826	66	a	05:16,1	5	01:00,0	4	01:00,7	0,718
9991			05:01,1	65	00:00,1	51	00:01,4	1,234
9991			05:01,3	66	00:00,1	51	00:00,3	0,144
312042	50	a	03:06,5	3	01:00,5	3	01:00,5	-2,046
1248877	94	a	04:16,9	4	01:00,9	4	01:00,9	-0,95
9991			05:03,4	67	00:00,1	51	00:02,1	1,917
941606	72	a	04:58,3	5	00:58,0	5	00:58,0	-0,424
9991			05:05,0	68	00:00,1	51	00:01,6	1,476
32563	100	a	04:00,3	4	00:57,9	4	00:57,9	-1,092
9991			05:05,8	69	00:00,1	51	00:00,7	0,597
1691303	99	a	05:04,4	5	00:57,6	5	00:57,6	-0,913
9991			05:08,2	70	00:00,1	51	00:02,4	2,291
1558576	51	a	05:08,1	5	00:58,2	5	00:58,2	-0,203
9991			05:18,5	71	00:00,1	51	00:10,3	10,207
105283	52	a	04:49,4	5	00:56,6	4	00:57,8	1,146

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
9991			05:24,8	72	00:00,1	51	00:06,2	6,109
1802633	77	a	02:09,9	2	01:02,7	1	01:07,2	4,526
9991			05:36,3	73	00:00,1	51	00:11,5	11,387
1266400	92	a	05:07,2	5	00:59,2	5	00:59,2	-0,644
9991			05:36,7	74	00:00,1	51	00:00,4	0,283
1113482	91	a	04:59,7	5	00:58,5	5	00:58,5	-0,152
9991			05:47,7	75	00:00,1	51	00:11,0	10,866
258440	68	a	03:57,7	4	00:58,8	4	00:58,8	-0,095
9991			05:56,1	76	00:00,1	51	00:08,4	8,274
432936	55	a	05:22,0	5	01:00,1	3	01:05,7	5,58
9991			05:58,0	77	00:00,1	51	00:01,8	1,686
536130	88	a	04:59,3	5	00:58,5	4	00:58,7	0,162
9991			05:59,4	78	00:00,1	51	00:01,5	1,318
166826	66	a	06:15,8	6	00:59,7	6	00:59,7	-0,244
9991			06:00,5	79	00:00,1	51	00:01,1	0,963
312042	50	a	04:05,9	4	00:59,5	4	00:59,5	-1
9991			06:01,1	80	00:00,1	51	00:00,6	0,417
9991			06:01,2	81	00:00,1	51	00:00,2	0,016
1248877	94	a	05:16,6	5	00:59,7	5	00:59,7	-1,16
941606	72	a	05:56,1	6	00:57,8	6	00:57,8	-0,206
9991			06:02,9	82	00:00,1	51	00:01,7	1,583
32563	100	a	04:58,2	5	00:57,9	4	00:57,9	0,047
1691303	99	a	06:01,6	6	00:57,2	6	00:57,2	-0,335
9991			06:06,7	83	00:00,1	51	00:03,8	3,621
1558576	51	a	06:06,7	6	00:58,2	5	00:58,5	0,358
9991			06:15,6	84	00:00,1	51	00:08,8	8,707
105283	52	a	05:46,4	6	00:56,6	4	00:57,0	0,387
9991			06:26,8	85	00:00,1	51	00:11,2	11,081
1802633	77	a	03:11,9	3	01:02,0	3	01:02,0	-0,733
9991			06:35,1	86	00:00,1	51	00:08,4	8,216
1113482	91	a	05:58,1	6	00:58,4	6	00:58,4	-0,089
9991			06:35,9	87	00:00,1	51	00:00,8	0,626
1266400	92	a	06:06,8	6	00:59,2	5	00:59,6	0,425
9991			06:46,5	88	00:00,1	51	00:10,6	10,494
258440	68	a	04:56,5	5	00:58,8	5	00:58,8	-0,022
9991			06:56,1	89	00:00,1	51	00:09,6	9,415
9991			06:56,2	90	00:00,1	51	00:00,2	0,028
432936	55	a	06:21,9	6	00:59,9	6	00:59,9	-0,205
536130	88	a	05:57,6	6	00:58,3	6	00:58,3	-0,249
9991			06:58,8	91	00:00,1	51	00:02,5	2,401
166826	66	a	07:15,2	7	00:59,4	7	00:59,4	-0,357
9991			06:59,3	92	00:00,1	51	00:00,5	0,401
312042	50	a	05:04,7	5	00:58,8	5	00:58,8	-0,653
9991			07:00,0	93	00:00,1	51	00:00,6	0,499
9991			07:00,1	94	00:00,1	51	00:00,2	0,022
941606	72	a	06:54,8	7	00:57,8	6	00:58,7	0,909
1248877	94	a	06:15,7	6	00:59,0	6	00:59,0	-0,682
9991			07:00,8	95	00:00,1	51	00:00,7	0,516
1691303	99	a	06:59,4	7	00:57,2	6	00:57,8	0,576
9991			07:01,6	96	00:00,1	51	00:00,8	0,644
32563	100	a	05:56,8	6	00:57,9	4	00:58,6	0,724

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
9991			07:05,1	97	00:00,1	51	00:03,5	3,395
1558576	51	a	07:05,1	7	00:58,2	5	00:58,4	0,204
9991			07:12,4	98	00:00,1	51	00:07,3	7,175
105283	52	a	06:43,2	7	00:56,6	4	00:56,8	0,213
9991			07:27,2	99	00:00,1	51	00:14,8	14,638
1802633	77	a	04:12,3	4	01:00,4	4	01:00,4	-1,571
9991			07:33,5	100	00:00,1	51	00:06,3	6,162
1113482	91	a	06:56,5	7	00:58,3	7	00:58,3	-0,058
9991			07:34,9	101	00:00,1	51	00:01,4	1,268
1266400	92	a	07:05,8	7	00:59,0	7	00:59,0	-0,168
9991			07:45,4	102	00:00,1	51	00:10,5	10,411
258440	68	a	05:55,4	6	00:58,8	5	00:58,9	0,109
9991			07:54,5	103	00:00,1	51	00:09,1	8,958
536130	88	a	06:55,9	7	00:58,3	6	00:58,3	0,02
9991			07:55,8	104	00:00,1	51	00:01,3	1,188
432936	55	a	07:21,7	7	00:59,8	7	00:59,8	-0,166
9991			07:58,3	105	00:00,1	51	00:02,4	2,277
9991			07:58,6	106	00:00,1	51	00:00,4	0,225
941606	72	a	07:53,1	8	00:57,8	6	00:58,3	0,485
312042	50	a	06:04,0	6	00:58,8	5	00:59,3	0,518
9991			07:59,9	107	00:00,1	51	00:01,3	1,125
9991			08:00,3	108	00:00,1	51	00:00,4	0,224
1248877	94	a	07:15,5	7	00:59,0	6	00:59,8	0,74
32563	100	a	06:55,6	7	00:57,9	4	00:58,7	0,833
9991			08:02,2	109	00:00,1	51	00:02,0	1,853
1691303	99	a	08:01,0	8	00:57,2	6	01:01,6	4,374
9991			08:03,3	110	00:00,1	51	00:01,0	0,9
1558576	51	a	08:03,2	8	00:58,2	5	00:58,2	0,01
9991			08:09,2	111	00:00,1	51	00:05,9	5,768
105283	52	a	07:40,0	8	00:56,6	4	00:56,8	0,156
9991			08:27,8	112	00:00,1	51	00:18,6	18,463
1802633	77	a	05:12,9	5	01:00,4	4	01:00,6	0,214
9991			08:31,8	113	00:00,1	51	00:04,0	3,861
1113482	91	a	07:54,8	8	00:58,3	8	00:58,3	-0,034
9991			08:43,6	114	00:00,1	51	00:11,8	11,643
258440	68	a	06:53,5	7	00:58,1	7	00:58,1	-0,655
9991			08:50,9	115	00:00,1	51	00:07,4	7,232
1266400	92	a	08:21,8	8	00:59,0	7	01:16,1	17,067
9991			08:53,3	116	00:00,1	51	00:02,4	2,232
536130	88	a	07:54,7	8	00:58,3	6	00:58,8	0,507
9991			08:55,2	117	00:00,1	51	00:01,9	1,793
432936	55	a	08:21,1	8	00:59,4	8	00:59,4	-0,383
9991			08:55,9	118	00:00,1	51	00:00,6	0,476
941606	72	a	08:50,7	9	00:57,6	9	00:57,6	-0,238
9991			08:57,5	119	00:00,1	51	00:01,7	1,531
312042	50	a	07:02,9	7	00:58,8	5	00:58,9	0,089
9991			08:58,6	120	00:00,1	51	00:01,1	0,966
9991			08:58,9	121	00:00,1	51	00:00,3	0,175
32563	100	a	07:53,9	8	00:57,9	4	00:58,4	0,489
1248877	94	a	08:14,5	8	00:59,0	6	00:59,0	0,001
9991			09:01,4	122	00:00,1	51	00:02,4	2,306

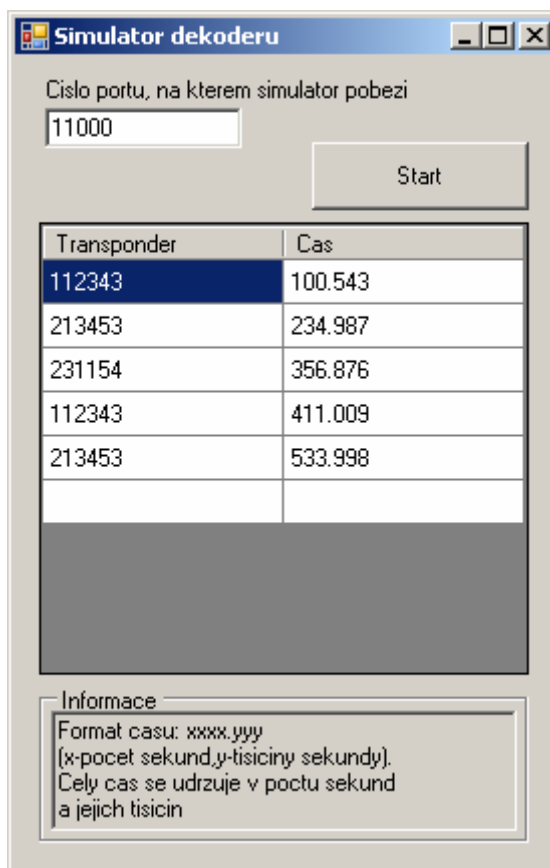
Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
1558576	51	a	09:01,4	9	00:58,1	9	00:58,1	-0,063
9991			09:18,6	123	00:00,1	51	00:17,2	17,096
105283	52	a	08:49,4	9	00:56,6	4	01:09,4	12,78
9991			09:23,5	124	00:00,1	51	00:04,9	4,738
1691303	99	a	09:22,2	9	00:57,2	6	01:21,1	23,915
9991			09:27,9	125	00:00,1	51	00:04,4	4,25
1802633	77	a	06:13,0	6	01:00,1	6	01:00,1	-0,305
9991			09:30,0	126	00:00,1	51	00:02,2	2,017
1113482	91	a	08:53,0	9	00:58,3	9	00:58,3	-0,062
9991			09:42,2	127	00:00,1	51	00:12,2	12,014
258440	68	a	07:52,1	8	00:58,1	7	00:58,6	0,482
9991			09:51,0	128	00:00,1	51	00:08,8	8,694
1266400	92	a	09:21,9	9	00:59,0	7	01:00,1	1,093
9991			09:52,2	129	00:00,1	51	00:01,1	1,01
536130	88	a	08:53,5	9	00:58,3	6	00:58,9	0,589
9991			09:54,0	130	00:00,1	51	00:01,9	1,724
941606	72	a	09:48,9	10	00:57,6	9	00:58,2	0,593
9991			09:54,5	131	00:00,1	51	00:00,5	0,318
432936	55	a	09:20,3	9	00:59,2	9	00:59,2	-0,144
9991			09:55,7	132	00:00,1	51	00:01,3	1,121
312042	50	a	08:01,1	8	00:58,2	8	00:58,2	-0,574
9991			09:56,7	133	00:00,1	51	00:00,9	0,766
32563	100	a	08:51,9	9	00:57,9	4	00:58,0	0,14
9991			09:57,1	134	00:00,1	51	00:00,5	0,321
1248877	94	a	09:12,7	9	00:58,2	9	00:58,2	-0,876
9991			09:59,9	135	00:00,1	51	00:02,8	2,658
1558576	51	a	09:59,9	10	00:58,1	9	00:58,5	0,408
9991			10:21,5	136	00:00,1	51	00:21,6	21,464
1691303	99	a	10:20,2	10	00:57,2	6	00:58,0	0,793
9991			10:27,8	137	00:00,1	51	00:06,3	6,113
1802633	77	a	07:12,9	7	00:59,9	7	00:59,9	-0,217
9991			10:28,3	138	00:00,1	51	00:00,6	0,425
1113482	91	a	09:51,3	10	00:58,3	9	00:58,3	0,031
9991			10:40,7	139	00:00,1	51	00:12,4	12,239
258440	68	a	08:50,6	9	00:58,1	7	00:58,5	0,367
9991			10:50,7	140	00:00,1	51	00:10,0	9,876
9991			10:50,9	141	00:00,1	51	00:00,2	0,073
536130	88	a	09:52,1	10	00:58,3	6	00:58,5	0,265
1266400	92	a	10:21,8	10	00:59,0	7	00:59,9	0,911
9991			10:52,1	142	00:00,1	51	00:01,2	1,031
941606	72	a	10:47,0	11	00:57,6	9	00:58,1	0,48
9991			10:53,1	143	00:00,1	51	00:01,0	0,823
432936	55	a	10:18,9	10	00:58,6	10	00:58,6	-0,68
9991			10:54,1	144	00:00,1	51	00:01,1	0,946
312042	50	a	08:59,5	9	00:58,2	8	00:58,4	0,168
9991			10:54,8	145	00:00,1	51	00:00,7	0,527
9991			10:55,1	146	00:00,1	51	00:00,3	0,134
32563	100	a	09:50,1	10	00:57,9	4	00:58,1	0,266
1248877	94	a	10:10,6	10	00:58,0	10	00:58,0	-0,2
9991			10:58,5	147	00:00,1	51	00:03,4	3,281
1558576	51	a	10:58,5	11	00:58,1	9	00:58,6	0,477

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
9991			11:19,2	148	00:00,1	51	00:20,7	20,549
1691303	99	a	11:17,8	11	00:57,2	6	00:57,7	0,45
9991			11:27,4	149	00:00,1	51	00:08,2	8,072
1113482	91	a	10:50,4	11	00:58,3	9	00:59,1	0,817
9991			11:28,0	150	00:00,1	51	00:00,6	0,437
1802633	77	a	08:13,1	8	00:59,9	7	01:00,2	0,326
9991			11:39,1	151	00:00,1	51	00:11,2	11,024
258440	68	a	09:49,1	10	00:58,1	7	00:58,4	0,288
9991			11:49,3	152	00:00,1	51	00:10,2	10,015
536130	88	a	10:50,6	11	00:58,3	6	00:58,6	0,296
9991			11:50,8	153	00:00,1	51	00:01,5	1,348
9991			11:51,1	154	00:00,1	51	00:00,3	0,158
941606	72	a	11:45,6	12	00:57,6	9	00:58,7	1,09
1266400	92	a	11:21,9	11	00:59,0	7	01:00,1	1,143
9991			11:51,7	155	00:00,1	51	00:00,7	0,518
432936	55	a	11:17,6	11	00:58,6	10	00:58,7	0,097
9991			11:52,4	156	00:00,1	51	00:00,7	0,57
9991			11:52,8	157	00:00,1	51	00:00,4	0,275
312042	50	a	09:57,8	10	00:58,2	8	00:58,3	0,068
9991			11:53,2	158	00:00,1	51	00:00,4	0,237
32563	100	a	10:48,1	11	00:57,9	4	00:58,0	0,164
1248877	94	a	11:08,8	11	00:58,0	10	00:58,1	0,176
9991			11:57,3	159	00:00,1	51	00:04,1	3,948
1558576	51	a	11:57,3	12	00:58,1	9	00:58,8	0,693
9991			12:16,9	160	00:00,1	51	00:19,6	19,46
1691303	99	a	12:15,6	12	00:57,2	6	00:57,7	0,501
9991			12:25,8	161	00:00,1	51	00:08,9	8,743
1113482	91	a	11:48,8	12	00:58,3	9	00:58,4	0,141
9991			12:27,4	162	00:00,1	51	00:01,7	1,518
1802633	77	a	09:12,6	9	00:59,5	9	00:59,5	-0,408
9991			12:37,7	163	00:00,1	51	00:10,3	10,116
258440	68	a	10:47,6	11	00:58,1	7	00:58,6	0,421
9991			12:48,2	164	00:00,1	51	00:10,5	10,355
9991			12:48,4	165	00:00,1	51	00:00,2	0,033
536130	88	a	11:49,5	12	00:58,3	6	00:58,9	0,621
941606	72	a	12:43,2	13	00:57,6	13	00:57,6	-0,006
9991			12:50,3	166	00:00,1	51	00:02,0	1,815
9991			12:50,6	167	00:00,1	51	00:00,3	0,189
1266400	92	a	12:21,2	12	00:59,0	7	00:59,2	0,254
432936	55	a	12:16,5	12	00:58,6	10	00:58,9	0,331
9991			12:52,2	168	00:00,1	51	00:01,6	1,433
9991			12:52,6	169	00:00,1	51	00:00,4	0,264
1248877	94	a	12:07,8	12	00:58,0	10	00:59,0	1,025
32563	100	a	11:47,9	12	00:57,9	4	00:59,8	1,884
9991			12:56,1	170	00:00,1	51	00:03,5	3,338
312042	50	a	11:01,5	11	00:58,2	8	01:03,7	5,433
9991			12:57,8	171	00:00,1	51	00:01,7	1,562
1558576	51	a	12:57,8	13	00:58,1	9	01:00,5	2,404
9991			13:14,8	172	00:00,1	51	00:17,0	16,888
1691303	99	a	13:13,5	13	00:57,2	6	00:57,9	0,69
9991			13:24,4	173	00:00,1	51	00:09,5	9,407

Transponder	Cislo	Prijmeni	CelkovyCas	PK	NejCas	NK	PosledniKolo	Zlepseni
1113482	91	a	12:47,4	13	00:58,3	9	00:58,6	0,319
9991			13:27,0	174	00:00,1	51	00:02,7	2,53
1802633	77	a	10:12,2	10	00:59,5	9	00:59,6	0,115
9991			13:36,2	175	00:00,1	51	00:09,2	9,031
258440	68	a	11:46,1	12	00:58,1	7	00:58,5	0,359
9991			13:46,4	176	00:00,1	51	00:10,2	10,025
941606	72	a	13:41,2	14	00:57,6	13	00:58,0	0,428
9991			13:46,8	177	00:00,1	51	00:00,5	0,324
536130	88	a	12:48,2	13	00:58,3	6	00:58,6	0,367
9991			13:50,5	178	00:00,1	51	00:03,7	3,583
1266400	92	a	13:21,4	13	00:59,0	7	01:00,2	1,239
9991			13:51,3	179	00:00,1	51	00:00,8	0,639
432936	55	a	13:17,1	13	00:58,6	10	01:00,7	2,117
9991			13:53,7	180	00:00,1	51	00:02,4	2,288
32563	100	a	12:49,0	13	00:57,9	4	01:01,1	3,261
9991			13:54,9	181	00:00,1	51	00:01,1	0,974
312042	50	a	12:00,3	12	00:58,2	8	00:58,8	0,543
9991			13:57,7	182	00:00,1	51	00:02,8	2,708
1248877	94	a	13:13,2	13	00:58,0	10	01:05,5	7,515
9991			14:12,6	183	00:00,1	51	00:14,9	14,744
1691303	99	a	14:11,2	14	00:57,2	6	00:57,8	0,55
9991			14:23,0	184	00:00,1	51	00:10,4	10,242
1113482	91	a	13:46,0	14	00:58,3	9	00:58,6	0,358
9991			14:26,8	185	00:00,1	51	00:03,8	3,664
1802633	77	a	11:11,9	11	00:59,5	9	00:59,7	0,272
9991			14:34,8	186	00:00,1	51	00:08,1	7,919
258440	68	a	12:44,8	13	00:58,1	7	00:58,6	0,489
9991			14:44,7	187	00:00,1	51	00:09,9	9,755
941606	72	a	14:39,6	15	00:57,6	13	00:58,3	0,761
9991			14:45,1	188	00:00,1	51	00:00,4	0,287
536130	88	a	13:46,5	14	00:58,3	6	00:58,3	0,021
9991			14:50,7	189	00:00,1	51	00:05,5	5,407
432936	55	a	14:16,5	14	00:58,6	10	00:59,4	0,813
9991			14:52,1	190	00:00,1	51	00:01,4	1,253
32563	100	a	13:47,4	14	00:57,9	4	00:58,3	0,443
9991			14:53,1	191	00:00,1	51	00:01,0	0,844
312042	50	a	12:58,5	13	00:58,2	13	00:58,2	-0,023
9991			15:00,0	192	00:00,1	51	00:06,9	6,808
1248877	94	a	14:15,6	14	00:58,0	10	01:02,3	4,349
9991			15:10,7	193	00:00,1	51	00:10,7	10,61
1691303	99	a	15:09,4	15	00:57,2	6	00:58,2	0,934
9991			15:21,4	194	00:00,1	51	00:10,6	10,466
1113482	91	a	14:44,4	15	00:58,3	9	00:58,4	0,144
9991			15:26,5	195	00:00,1	51	00:05,2	5,054
1802633	77	a	12:11,7	12	00:59,5	9	00:59,8	0,311
9991			15:38,3	196	00:00,1	51	00:11,8	11,615
258440	68	a	13:48,2	14	00:58,1	7	01:03,5	5,335

Zavodnik	NejCas	PosledniKolo	PK	Zlepseni
52 a	00:56,6	01:09,4	9	12,78
99 a	00:57,2	00:58,2	15	0,934
72 a	00:57,6	00:58,3	15	0,761
100 a	00:57,9	00:58,3	14	0,443
94 a	00:58,0	01:02,3	14	4,349
51 a	00:58,1	01:00,5	13	2,404
68 a	00:58,1	01:03,5	14	5,335
50 a	00:58,2	00:58,2	13	-0,023
91 a	00:58,3	00:58,4	15	0,144
88 a	00:58,3	00:58,3	14	0,021
55 a	00:58,6	00:59,4	14	0,813
92 a	00:59,0	01:00,2	13	1,239
66 a	00:59,4	00:59,4	7	-0,357
77 a	00:59,5	00:59,8	12	0,311
67 a	01:00,9	01:00,9	3	-3,275

Používané vlastní utility

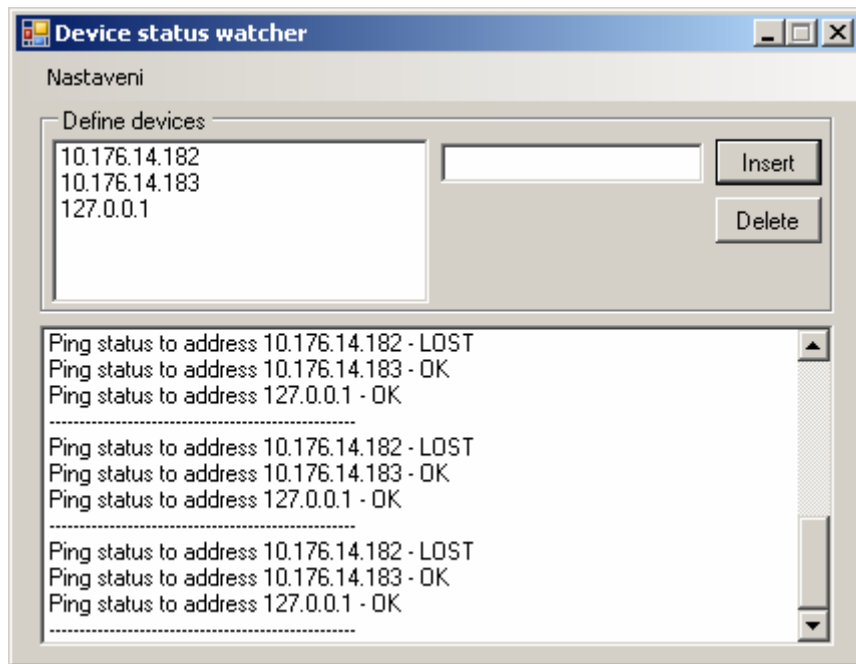
Simulátor dekodéru

Velkým problémem pro testování systému je fakt, že dekodér je finančně náročný a v České republice je jen pár takovýchto zařízení. Z tohoto důvodu jsem si vytvořil program, který tento dekodér simuluje a zastupuje.

Uživatelsky je tento program triviální. Nejprve musí být uživatelem zadáno číslo portu, na kterém daný dekodér poběží. V dalším kroku se musí nadefinovat tabulka transpondérů a časů. Jednotlivé řádky tabulky představují průjezdy cílovou čarou v definovaném čase. V posledním kroku se klikne na tlačítko „Start“ a program již běží samostatně.

Po spuštění programu TimeKeeper se uživatel přihlásí k dekodéru na adrese daného PC a portu, který byl v simulátoru definován.

Ping status



Tato malá utilita řeší problém sledování správné funkce počítačové sítě. Při používání informačního systému je velice důležité, aby jednotlivé uzly sítě byly připojeny a komunikovaly s ostatními.

Do tohoto programu se nastaví IP adresa nebo adresy, které se mají sledovat a daná utilita již pracuje samostatně. V definovaných sekundových cyklech se program stále dotazuje na status daných zařízení za pomoci příkazu Ping a dané IP adresy. Výsledky dotazování se zobrazují přehledně v textovém poli.

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

NÁZEV PRÁCE	Informační systém okruhových závodů
AUTOR PRÁCE	Bc. Radek Paclt
OBOR	Aplikovaná informatika v dopravě
ROK OBHAJOBY	2006
VEDOUCÍ PRÁCE	Ing. Karel Greiner
ANOTACE	Tvorba informačního systému okruhových závodů na definovaných požadavcích za pomoci modelování v UML.
KLÍČOVÁ SLOVA	Informační, systém, modelování, UML, okruhové, závody, program, měření, čas, analýza, návrh, nasazení, implementace