Komunikační protokol pro automobilový varovný systém

Jan Soukup

24. ledna 2006

Abstrakt

Tato práce popisuje návrh komunikačního protokolu, který je součástí diplomové práce věnující se automobilovému varovnému systému. Tento varovný systém si dává za úkol včas upozornit řidiče na nebezpečnou situaci před ním. Vysílací část komunikačního protokolu zde popsaná nepředpokládá momentálně žádné nadstandardní technologie (např. GPS) a je navržena maximálně jednoduše. Jak přijímač tak vysílač byl rozdělen do jednodušších podcelků, které jsou dále detailněji popsány. Byla ověřena funkčnost celého systému pomocí verifikačního nástroje UPPAL, *který neodhalil žádné nedostatky navrženého komunikačního protokolu. Lze tedy podotknout, že komunikační schéma je kompletní a může být v dalším kroku testováno na zhotovených prototypech.*

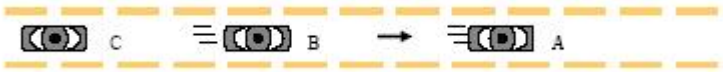
1 Zadání

Navrhněte komunikační protokol pro automobilový distribuovaný varovný systém pomocí verifikačního nástroje *UPPAAL*. Ověřte funkčnost protokolu pro jedno i více zařízení, otestujte systém na deadlock, dosažení určitých stavů a jeho logickou správnost. Navržený systém by měl být co nejvíce zpřehledněn a zjednodušen, zároveň je ale požadována jeho robustnost. Zařízení nepředpokládá použití speciálních technologií jako je např. GPS.

2 Rozbor problému

V současné době se řeší jak varovat řidiče před blížícím se nebezpečím. Příkladem může být dálnice. Představme si situaci, kdy je výrazně snížena viditelnost na silnici, například mlha nebo sněžení, a představme si, že se 100 metrů před námi utvořila kolona. Pokud zkombinujeme tedy nepříznivé podmínky s vysokou rychlostí na dálnicích a často i s nepozorností řidičů může velice lehce dojít k hromadným haváriím s tragickými následky.

Systém by měl tedy varovat dostatečně včas řidiče před blížícím se nebezpečím, ať už při špatné nebo dobré viditelnosti a dále pak varovat řidiče pohybující se za mým automobilem, pokud už nebyli taktéž varováni. Samozřejmě je nutno předejít nesprávnému varování, protože by to mohlo například způsobit další nehodu. Je tedy nutno vyloučit auta jedoucí v protisměru.
Dále by měl systém být schopen varovat při náhlých změnách rychlosti nebo směru automobilů pohybujících se přede mnou. Studie totiž ukazují, že téměř 60-ti% automobilových nehod by mohlo být zabráněno, kdyby byl řidič upozorněn alespoň půl sekundy před kolizí. Vezměme si případ z obrázku 1.



Obrázek 1: Příklad

Uvažujme, že všechny auta se pohybují rychlostí 35m/s (126km/hod). Pokud auto A uvidí překážku a náhle zabrzdí, řidič auta B uvidí brzdová světla. V závislosti na lidských reakcích, které se pohybují u člověka v rozmezí 0,7 - 1,5s, řidič vozu B začne brzdit. Řekněme s reakcí 1s. Pokud je tedy vzdálenost obou vozidel menší než 35m může dojít k nehodě. Zapojíme do schématu vůz C a předpokládáme, že řidič toho vozu nevidí brzdová světla vozu A. Jeho reakce bude také zpožděna o 1s. Takže začne-li A brzdit, vůz C se 2s pohybuje stále stejnou rychlostí a s největší pravděpodobností, pokud vzdálenost vozu C a B je menší než 35 metrů nebo C a A je menší než 70 metrů, dojde k nehodě.
Pokud by A mohl bezprostředně začít vysílat varovný signál, který by oba následující vozy zachytili s velmi malým zpožděním, C by měl poměrně velkou šanci, jak předejít nehodě a B by z této situace vytěžil například při špatné viditelnosti nebo při nepozornosti.

3 Požadavky na komunikační protokol

Aby mohl celý systém správně fungovat, je nutné správně navrhnout komunikaci mezi jednotlivými zařízeními a vypořádat se co nejlépe s chybami, které mohou nastat. Koncový výrobek by měl být pro řidiče cenově dobře dostupný, proto se snažím vyhnout použití GPS. Protože zařízení bude určitým způsobem vyhodnocovat rychlost, zrychlení a směr vozu je funkčnost prozatím omezena na rychlostní komunikace popřípadě může být využita jen na nebezpečné úseky. Základní funkce zařízení je včasné varování před překážkou na vozovce a nebezpečnými manévry při jízdě tak, aby šlo předejít nehodě. Z toho vychází předpoklad na okamžité zahájení vysílání při nebezpečné situaci a to po celou dobu trvání nebezpečí. Zároveň je potřeba

vyslat větší počet zpráv, kvůli chybám v komunikaci, ale jedná se o bezdrátový přenos, takže kapacita kanálu je omezena. Z tohoto důvodu je potřeba nějaký mechanismus pro snižování počtu zpráv.
Dalším problémem je varování pouze vybrané skupiny automobilů. Systém tedy musí být schopen rozeznat protijedoucí automobily, které nemají být varovány, ale zároveň mohou přeposílat varování dál. Posledním vhodným požadavkem je automatická aktivace a deaktivace systému při vjezdu do zabezpečeného úseku.

4 Prostředí

Komunikační protokol jsem navrhoval v prostředí *UPPAAL*, kde jsem následně provedl i jeho verifikaci. Z důvodu složitosti schématu jsem byl nucen rozdělit celý návrh do dvou částí - vysílací část, přijímací část. Přes veškerou snahu zjednodušovat schema se mi podařilo zverifikovat schéma vysílače pouze pro dvě zařízení. Při větším počtu již nestačí výkon počítače.

5 Vysílací část protokolu

5.1 Princip návrhu

Zařízení začne reagovat pokud se automobil dostane do abnormálního stavu. Pokud ano začne ihned vysílat varovné zprávy EWM (Emergency Warning Message) s maximální frekvencí a postupně, aby uvolnil kanál, frekvenci snižuje. Vysílá tak dlouho dokud je nebezpečný pro své okolí. Druhou možností, kdy systém vysílá EWM je, když přijme varování od jiného vozu. To se děje kvůli preventivnímu varování vzdálenějších vozů nebo vozů, které díky rušení nemohli zachytit signál z nebezpečného vozu.
Aby byla zjednodušena komunikace mezi jednotlivými vozidly a nějakým způsobem ošetřena chybovost přenosu, nekontroluje vysílač zda zprávu někdo přijal, ale vysílá EWM stále dokola s náhodnou dobou mezi jednotlivými přenosy. Odpadá tedy kontrola přijatých zpráv a díky systému "předávání" EWMs se zajistí redundantní množství zpráv.

5.2 Podrobný popis funkce

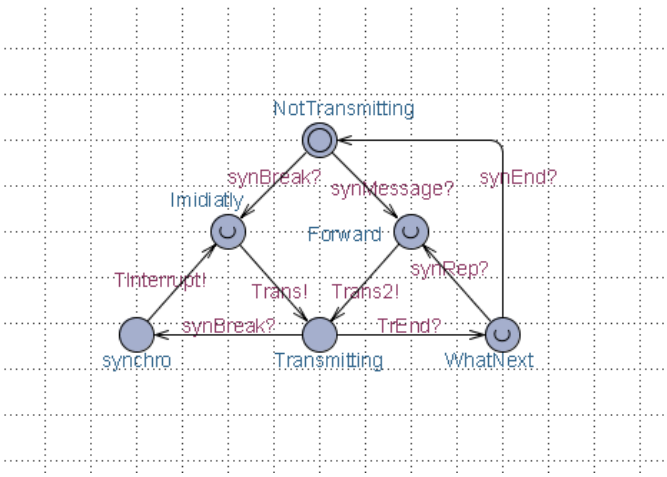
Celý návrh jsem se snažil co nejvíce zjednodušit a tím pádem použít i co nejméně časových proměnných, abych šetřil výpočetní výkon stroje. Definice globálních proměnných obsahuje pole kanálů - každý automobil má 10 synchronizačních kanálů, přes které jsou časově propojeny všechny části návrhu. Dále je zde vysílací vektor, který definuje dobu mezi jednotlivými zprávami. Ve schématech Vysílač1 a Vysílač2 je pak ještě použita jedna proměnná `clock`, která omezuje přechody mezi stavy a proměnná typu `int`,

která hlídá počet průchodů smyčkou. Vysílací část komunikačního protokolu jsem rozdělil do následujících schémat:

- *Auto* - toto schéma popisuje základní stavy automobilu. Přechody mezi jednotlivými stavy jsou synchronizovány s ostatními schématy.
- *Vysílač1* - schéma vysílače, který má proměnnou délku mezi jednotlivými vysílanými zprávami. Začíná vysílat s maximální frekvencí, která se postupem času snižuje. To se děje z důvodu požadavku okamžitého varování při nebezpečí a následném uvolňování kapacity přenosového kanálu.
- *Vysílač2* - tento vysílač slouží pro vysílání zpráv s konstantní frekvencí. Ta je samozřejmě nižší než u vysílače1, právě proto, aby příliš nezahlcoval kanál. To nastává buď po odvysílání celého vektoru Vysílače1 nebo při předávávání zpráv (preventivní varování).
  Oba vysílače byli původně zapouzdřeny do jednoho schématu, ovšem pro přehlednost a další zjednodušení byli rozděleny.
- *Překážka* - slouží pro detekci nebezpečného stavu a tím spustí veškeré vysílání varovných zpráv.
- *Transmission* - slouží pro detekci, že zpráva byla celá odvysílána.

Nyní podrobněji popíšu jednotlivá schémata a jejich stavy.

5.2.1 *Auto*

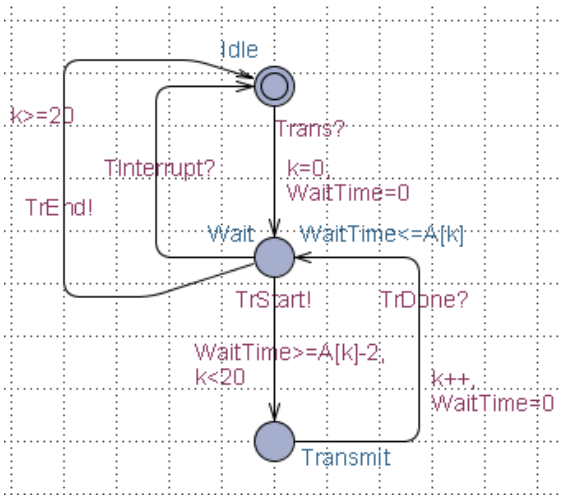


Obrázek 2: Verifikační schéma hlavní části

Základním stavem tohoto schématu je NotTransmitting, což je klidový stav a celý systém pouze poslouchá okolí a čeká na EWM. V případě, že

řidič například sešlápne prudce brzdy schéma *Překážka* synchronizuje přes `synBreak` *Auto* a dostaneme se do dalšího stavu Immediately. Tento stav je typu `Urgent` a proto v nulovém čase synchronizujeme *Vysílač1* a zahájíme vysílání. Po odvysílání se přes `TrEnd` dostaneme do stavu WhatNext, kde se rozhodne, zda-li má vozidlo stále abnormální chování - v tom případě se spustí *Vysílač2*, nebo nikoliv a přejdeme opět do klidového stavu.
Automobil může v tomto cyklu fyzicky již přestat být nebezpečným pro okolí, nicméně se dokončí vysílací cyklus (řádově do 3 sekund) a až potom se ukončí vysílání. Během této doby ale může již dojít k dalšímu nebezpečnému manévru a proto je třeba obnovit opět maximální frekvenci vysílání. K tomu slouží opět synchronizační kanál `synBreak` ze stavu *Transmitting* a následně spustí opět *Vysílač1*.

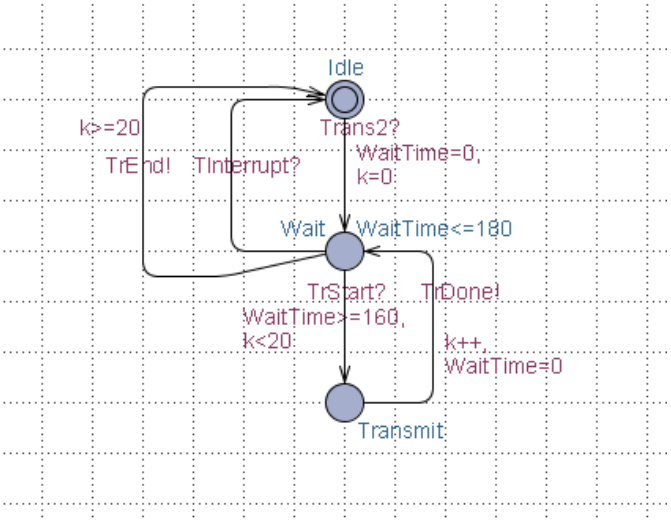
5.2.2 *Vysílač1*



Obrázek 3: schéma vysílače s vysílacím vektorem

Vysílač čeká na synchronizaci od *Auto* a následně vynuluje proměnnou int `k` (tato celočíselná proměnná indikuje počet průchodů smyčkou a po dvaceti opakováních vysílání ukončí) a proměnnou clock `WaitTime`, která nastavuje dobu čekání mezi jednotlivým odesíláním zpráv a přejde do stavu Wait. V tomhle stavu čeká definovanou dobu - podle vysílacího vektoru, který je zadán jako vstupní parametr a následně zahájí vysílání (pokud počet průchodů smyčkou je menší jak 20). Tím zároveň synchronizuje *Transmission*, který hlídá odeslání zprávy. Po odeslání předá slovo zpět vysílači, který ukončí vysílání a přejde opět do stavu Wait. Cykly se opakují do doby než je počet průchodů větší jak 20 nebo se automobil znovu dostane do nebezpečného stavu (synchr. kanál `TInterrupt`) a následně předá řízení schématu *Auto*.

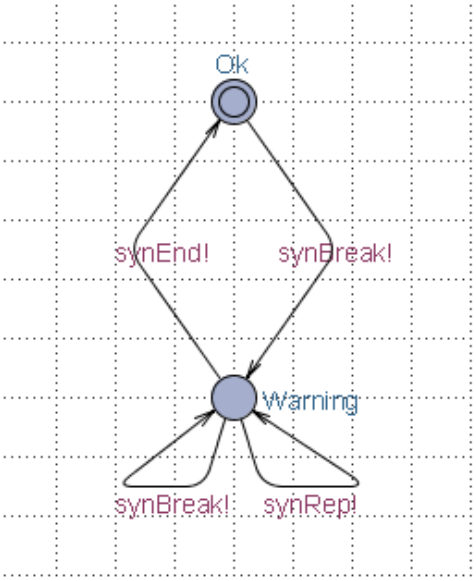
5.2.3 *Vysílač2*



Obrázek 4: schéma vysílače s minimální frekvencí vysílání

Vysílač je téměř shodný s *Vysílač1* pouze doba `WaitTime` je po celou dobu stejná a počet průchodů smyčkou je zmenšen na 10, aby se snížila doba strávená ve smyčce.

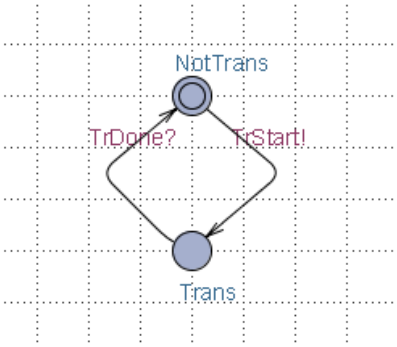
5.2.4 *Překážka*



Obrázek 5: schéma pro detekci nebezpečného chování

Slouží pro detekci nebezpečného chování. Pokud například řidič prudce zabrzdí přejde automat do stavu Warning, tím synchronizuje *Auto* a spustí vysílání. Ze stavu Warning může buď přejít zpět do stavu OK a tím ukončí vysílání nebo v tomto stavu zůstane a vysílání se opakuje přes synchronizační kanály `synRep` nebo `synBreak`.

5.2.5 *Transmission*



Obrázek 6: schéma pro detekci odvysílané zprávy

Tento automat slouží pro detekci odvysílání celé zprávy. Je spuštěn od vysílače, následně čeká dokud je zpráva celá odvysílána a opět předá řízení vysílači.

5.3 Verifikace

Po vytvoření všech částí vysílací části jsem verifikoval systém pomocí *UPPAAL verifier* a testoval jeho vlastnosti. Zde jsou některé důležité verifikační dotazy a jejich formule.

1. **Testování systému na deadlock**

   ```
   A[] not deadlock

   Property is satisfied
   ```

   Popis: testování systému na deadlock, z výpočetních důvodů otestováno pouze pro dvě zařízení.

2. **Existuje stav, kdy vysílají např. 3 vysílače současně?**

   ```
   E<> Vx1.Transmit && Vx2.Transmit && Vx3.Transmit
   ```

```
Property is satisfied
```

Popis: dotaz ukazuje, že je možné, aby v jednu chvíli vysílalo více vozidel. Proto je nutné opakovat vysílání EWM a přidat před každé vysílání náhodný čas.

3. **Je možné, aby vysílač začal znovu vysílat aniž by dokončil předchozí cyklus?**

```
E<> A1.synchro && P1.Warning

Property is satisfied
```

Popis: automobil může začít generovat varovný signál a chvíli na to se chovat jako bezpečné vozidlo. Poté se ovšem dostane do nebezpečného stavu a je potřeba, aby vysílač začal opět vysílat s maximální rychlostí, což, jak je vidět, jde.

4. **Je možné, aby zařízení vysílalo aniž by nenarušovalo bezpečný provoz?**

```
E<> A2.Transmitting && P2.Ok

Property is not satisfied
```

Popis: není to možné právě proto, že tento model neobsahuje přijímací část a systém zatím není schopen rozpoznat, zda-li přijal nějakou zprávu a mohl ji předat dál.

Další dotazy modelují různé situace a jejich formule jsou podobné, proto není nutno je zde všechny uvádět.

6 Přijímací část protokolu

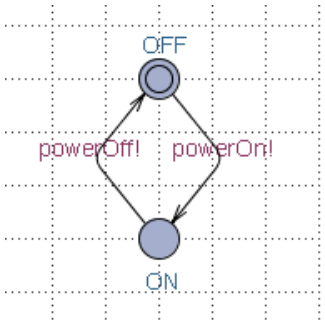
6.1 Princip návrhu

Přijímací část komunikačního protokolu jsem také rozdělil na několik jednoduchých podautomatů, které jsou navzájem provázány. Tato část již ale zdaleka není tak složitá jako vysílací schéma, protože celý proces se odehrává uvnitř zařízení. V případě u vysílací části musíme brát v úvahu chování automobilu navenek - to jest, jakým způsobem ovlivňuje komunikaci mezi jednotlivými zařízeními. Všechny automobily mají vlastní vysílače, detektory

překážek atd. a ty musí být jednoznačně přiřazeny jako jeden funkční celek. Naproti tomu schéma přijímače nikterak nezasahuje do vnějších záležitostí a pouze „poslouchá okolí“. Proto nemusím vytvářet složitou provázanost a můžu toto schéma navrhnout zcela odděleně

6.2 *Podrobný popis funkce*

6.2.1 powerOn

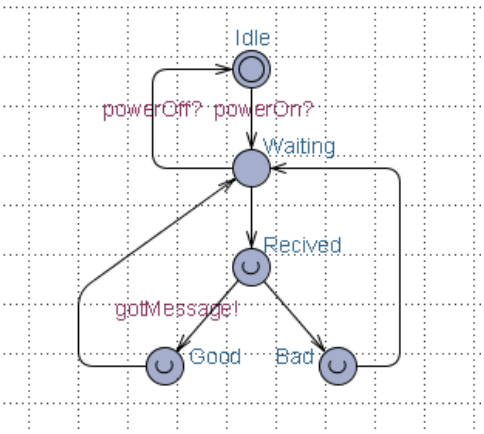


Obrázek 7: schéma zapíná celé zařízení

Tento jednoduchý automat uvádí celé zařízení automaticky do chodu. Jak bylo již popsáno v druhé části, celý systém je funkční v omezeném rozsahu - tedy například dálnice nebo pouze hlídaný úsek komunikace. Momentální představa je taková, že na určitých místech podél komunikace budou rozmístěny vysílače, které zapnou zařízení a předají mu potřebné informace. Při příchodu takovéhoto signálu přejde tento automat do stavu ON a přes synchronizační kanál `powerOn` zapne přijímač. Při příchodu vypínacího signálu přejde automat zpět do stavu OFF a synchronizuje opět schéma přijímače, čímž se celé zařízení vypne.

6.2.2 *ReciveCheck*

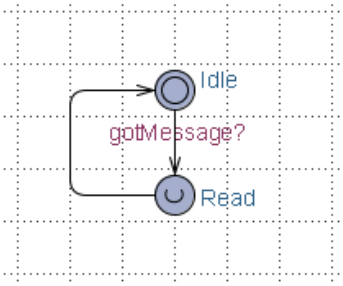
Uvažujeme zde jednoduchý automat, který po zapnutí začne přijímat zprávy od okolí a nijak toto okolí neovlivňuje. Po přijmutí signálu zkontroluje, zda-li zpráva byla přijata celá a obsahuje požadované informace, pokud ano, přečte ji a opět přijímá další zprávy, pokud nebyla dobře přijata, zprávu zahodí a čeká na novou.



Obrázek 8: schéma přijímá a kontroluje zprávy

6.2.3 *Message*

Toto schéma má za úkol po přijetí zprávy přejít do stavu READ, aby umožnilo přečtení zprávy. Automat je synchronizován kanálem gotMessage automatu *ReciveCheck*, který tuto synchronizaci umožní po příchodu správného formátu zprávy.



Obrázek 9: schéma zpracovává přijaté zprávy

6.3 Verifikace

1. **Testování systému na deadlock**

```
A[] not deadlock

Property is satisfied
```

Popis: testování systému na deadlock, systém neuvázne.

2. **Existuje stav, kdy přijímač dostal zprávu a ta je přečtena?**

```
E<> RCh.Good && M.Read

Property is satisfied
```

   Popis: dotaz ukazuje, že je možno přijmout zprávu a načíst informace, které obsahuje.

3. **Je možné, aby systém zpracovával přijatou zprávu a byla přijata nová?**

```
E<> RCh.Recived && M.Read

Property is satisfied
```

   Popis: je vidět, že pokud se přijme zpráva a ta je následně zpracovávána, může celý systém přijmout další zprávu.

4. **Je možné, aby zařízení přečetlo přijatou zprávu, ikdyž nebyla přijata správně?**

```
M.Read --> RCh.Recived

Property is not satisfied
```

   Popis: není možné, protože schéma *Message* čeká na „pokyn“ k přečtení zprávy.

5. **Může zařízení přijímat zprávy pokud je vypnuté?**

```
E<> P.OFF && RCh.Waiting

Property is not satisfied
```

   Popis: zařízení nemůže pracovat dokud nebylo aktivováno.

7 Závěr

Navrhované části - vysílací schéma komunikačního protokolu a přijímací schéma komunikačního protokolu - jsou maximálně zjednodušeny a jsou obě zverifikovány. Jejich funkčnost je momentálně uspokojivá, ovšem v dalším kroku bude nutné je skloubit dohromady a odsimulovat jejich chování jako celku. Tyto simulace, kde bude figurovat více zařízení chovajících se jako samostatná vozidla, budou schopny ukázat na potřebu doladit časové konstanty jednotlivých vysílačů a ukáží další chyby, které mohou nastat.
Posledním důležitým problémem, který musí být vyřešen je možnost automatického zapínání a vypínání přístroje. V současnosti mám rozpracováno několik způsobů, jak toho dosáhnout, ovšem to už není předmětem této zprávy.
Celkově lze říci, že momentálně je připravena testovací verze komunikačního protokolu a lze přistoupit k jeho testování.