# SPARK 2014 tapasztalatok ETCS követelményekkel

Esettanulmány

Forrás: Rail, Space, Security: Three Case Studies for SPARK 2014

Szerzők: Claire Dross, Pavlos Efstathopoulos, David Lesens, David Mentre and Yannick Moy

# Tartalom

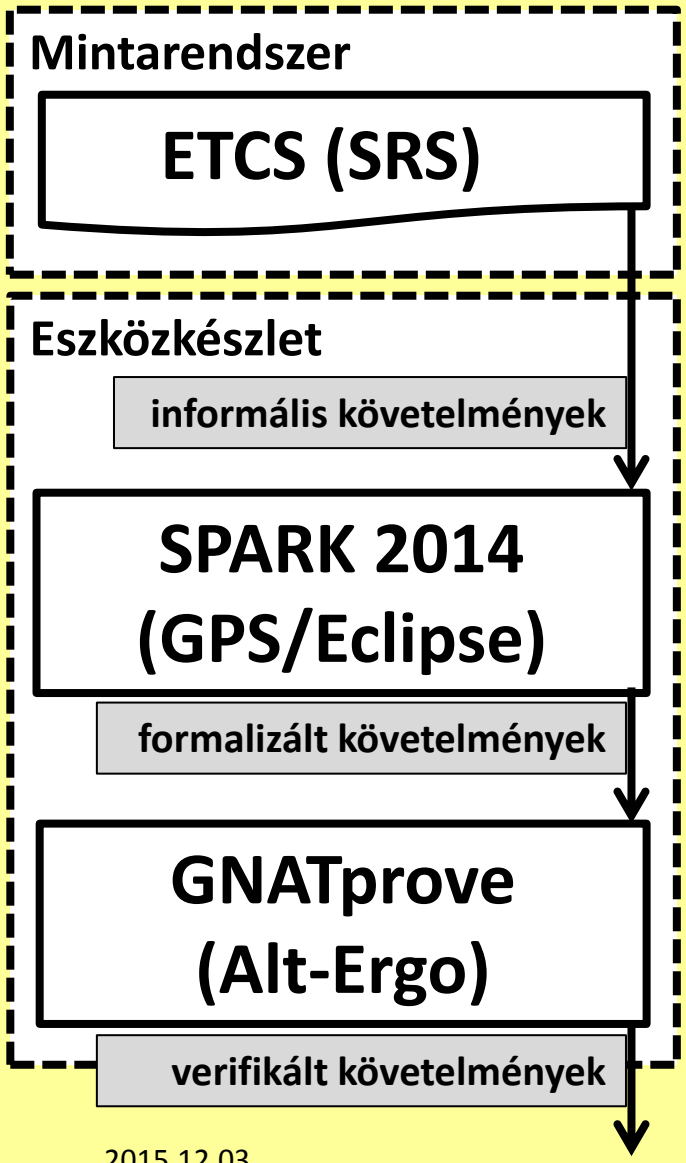
1. Az esettanulmány céljai
2. Az esettanulmány vizsgálati módszere
3. Az esettanulmány bemutatása

# Az esettanulmány céljai

- ETCS SRS követelmények SPARK 2014 programozási nyelvvel való formalizálási lehetőségeinek bemutatása, értékelése
- SPARK 2014 automatikus helyességbizonyítási képességeinek bemutatása, értékelése

- ETCS? SPARK 2014?

# Az esettanulmány vizsgálati módszere




**ETCS:** ***Egységes Európai Vonatbefolyásoló Rendszer*** (Europen Train Control System)
SRS: Rendszer Követelmény Specifikáció (System Requirement Spcification)

---

**SPARK 2014**: ***ADA alapú programozás nyelv***, amely formális módszerek alkalmazását célozza biztonság és védettségkritikus rendszerekben

**GPS/Eclipse**: SPARK 2014 támogatására szolgáló ***IDE*** (Integrated Development Environtment)

---

**GNATprove**: ***Formális verifikációs tool*** a SPARK 2014 kód elemzéséhez (GNAT compiler, Why3 platform)

**Alt-Ergo**: A GNATprove által alapértelmezésben használt nyílt forráskódú ***automatikus képletmegoldó*** SMT alapokon (Satisfiability Modulo Theories )

# Az esettanulmány

**Mitsubishi Electric R&D Centre Europe**

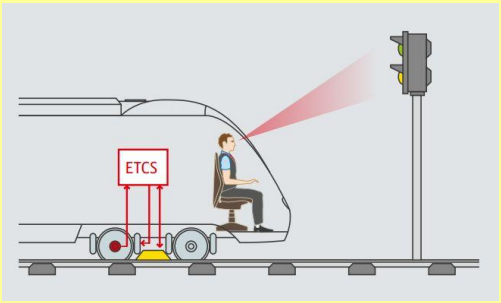
- SPARK 2014 használati tapasztalatok értékelése az ETCS (European Train Control System) követelmények egy kiragadott kis részén
- ERA, ETCS SRS SUBSET-026-1 3.4.0 (555 oldal)
- openETCS projekt

http://etcs.hu/
http://www.ertms.hu/
http://openetcs.org/
http://www.era.europa.eu/Core-Activities/ERTMS/Pages/home.aspx

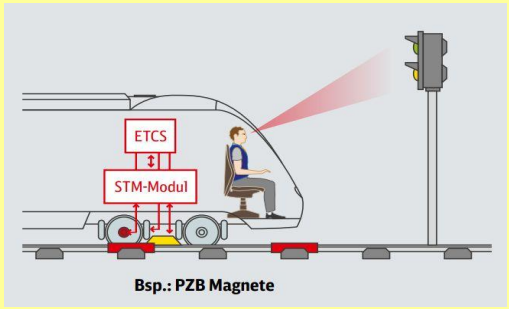
# A vizsgált rendszer: ETCS 1/2

- Európai Vonatbefolyásoló Rendszer
- **Célja:** egységes európai vonatbefolyásolás
- **Alapelve:** rádió alapú vonatbefolyásolás
- **Alkalmazási szintek:**
  - ETCS L0
  - STM
  - ETCS L1
  - ETCS L2
  - ETCS L3

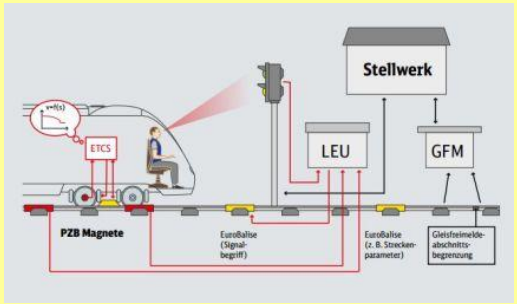
# A vizsgált rendszer: ETCS 2/2



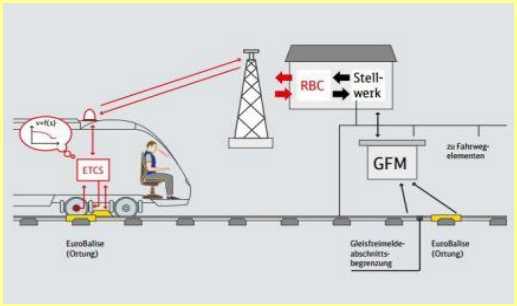

ETCS L0



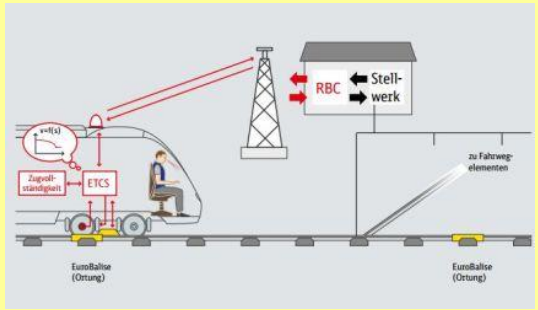

ETCS STM




ETCS L1




ETCS L2




ETCS L3

http://fahrweg.dbnetze.com/fahrweg-de/technik/strukturseite/funktionsprinzip/etcs_level/

# A vizsgált funkcionális követelmény csoport

- ETCS SRS 3.13 (SUBSET-026-3 3.4.0 12/05/14 )
- Kb. 300 követelmény

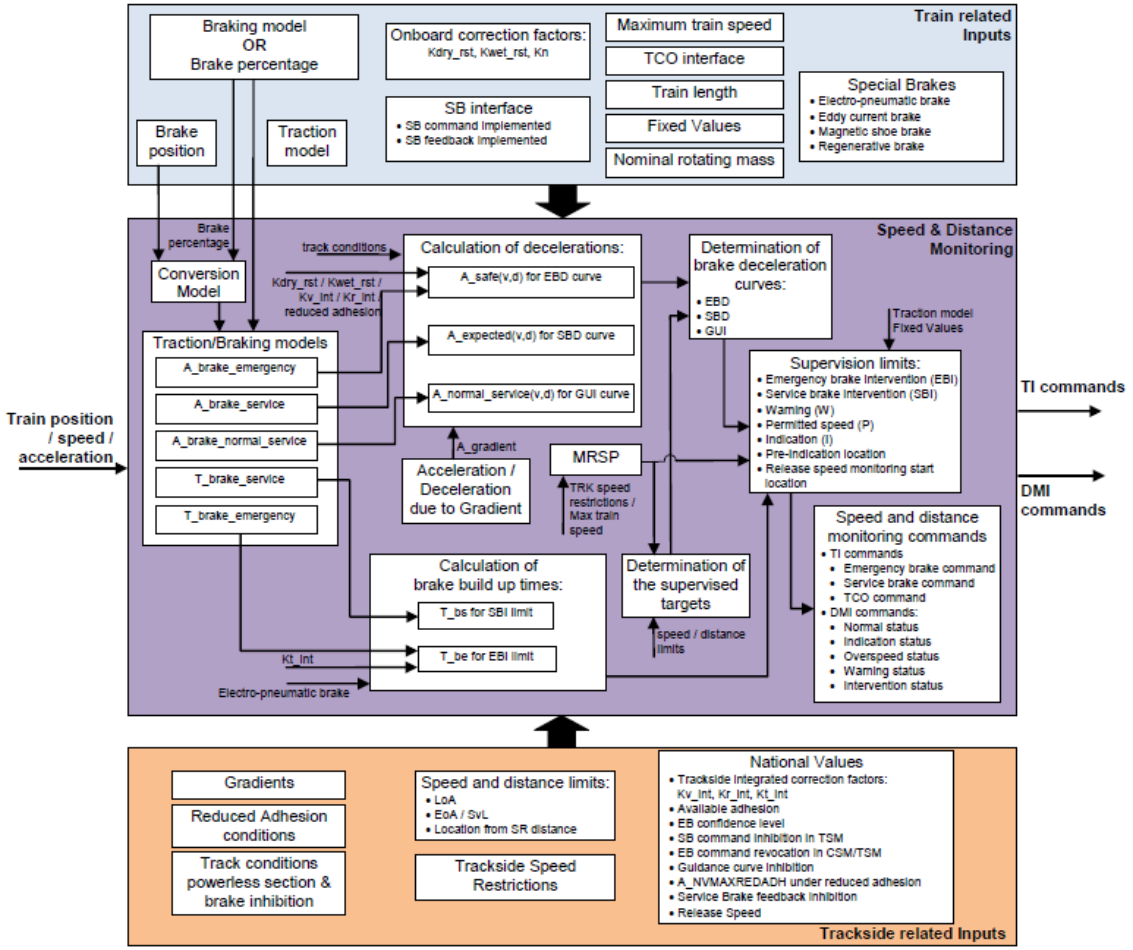



**Figure 28: Speed and distance monitoring overview**

# A vizsgált funkciók

- Sebesség és távolságfelügyeletre:
  - Lépcsős függvény kódolása
  - Szakaszonként konstans  függvény kódolása
- Csak funkciók modellezése
  (pl. sebességkorlátozás a távolsággal szemben)
- Cél: modellezni a két sebességkorlátozás összeolvadását, figyelembe véve a leginkább korlátozót

# A vizsgált tulajdonság

3.13.10.1.2 The following types of speed and distance monitoring are defined:

- Ceiling speed monitoring (CSM)
- Target speed monitoring (TSM)
- Release speed monitoring (RSM)

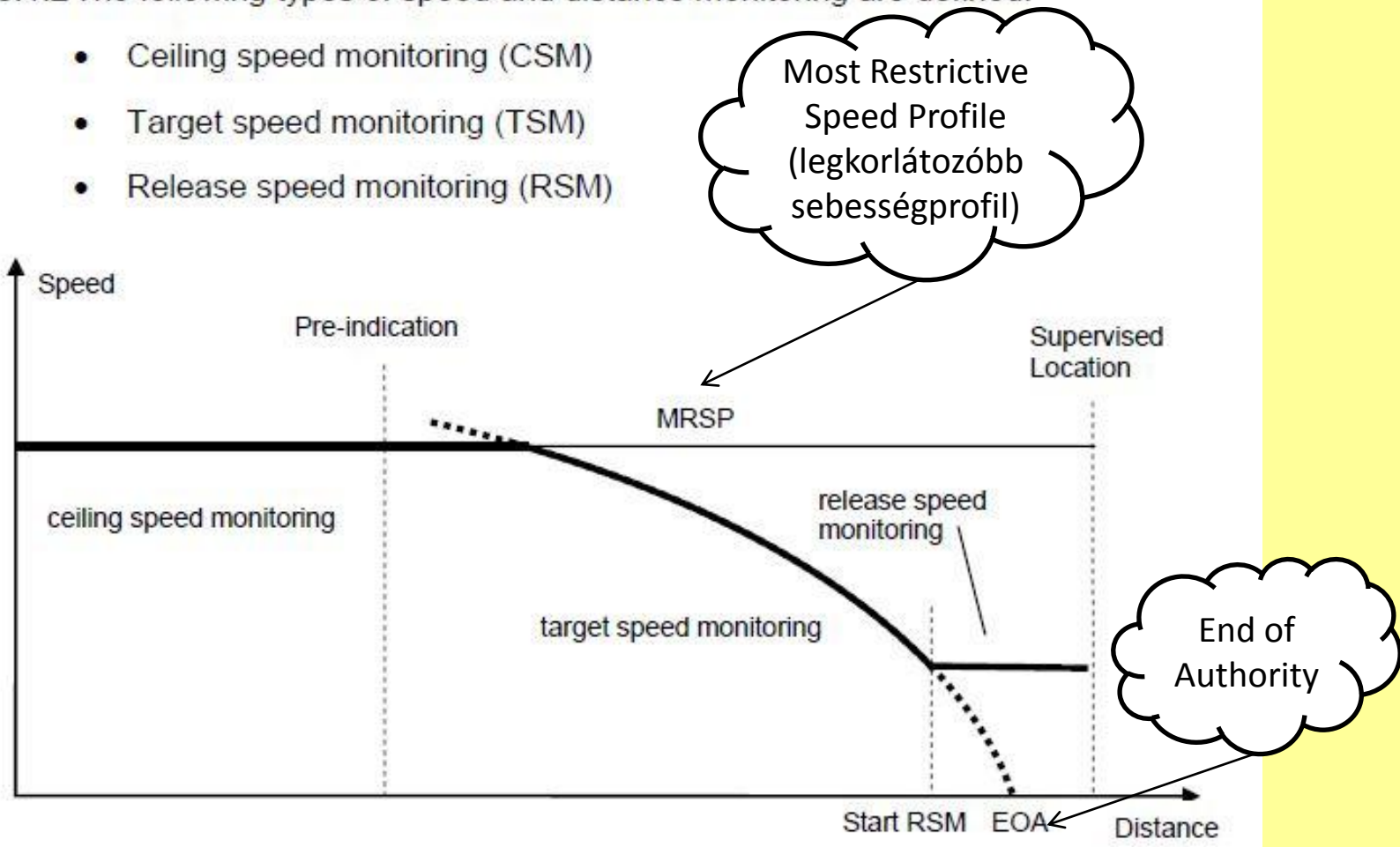



**Figure 54: Different types of speed and distance monitoring**

# A tulajdonság formalizálása példa 1/2

## A lépcsős függvények kódoláshoz használt adatstruktúrák

```
type Num_Delimiters_Range is range 0 .. 10;
```

A **range** is a signed integer value which ranges from a First to a Last.

```
type Function_Range is new Natural;
```


```
type Delimiter_Entry is record
   Delimiter : Function_Range;
   Value     : Float;
end record;
```

```
type Delimiter_Values is array
  (Num_Delimiters_Range) of Delimiter_Entry;
```

An **array** is a collection of elements which can be accessed by one or more index values.

```
type Step_Function_t is record
   Num_Delim : Num_Delimiters_Range;
   Step      : Delimiter_Values;
end record;
```

A **record** is a composite type that groups one or more fields.

# Tulajdonságok formalizálása példa 2/2

```
function Minimum_Until_Point
    (SFun : Step_Function_t; X : Function_Range)
                                  return Float
with
Pre  => Is_Valid(SFun),
Post =>
  -- (1) Returned value is the minimum before X
  (for all i in
     Num_Delimiters_Range'First..SFun.Num_delim =>
      (if X >= SFun.Step(i).Delimiter then
       Minimum_Until_Point'Result
          <= SFun.Step(i).Value))
  and
  -- (2) Returned value is a value of the step
  --     function before X
  ((for some i in
     Num_Delimiters_Range'First..SFun.Num_delim =>
      (X >= SFun.Step(i).Delimiter
       and
      (Minimum_Until_Point'Result
        = SFun.Step(i).Value))));
```

szerződés feje

előfeltétel

univerzális kvantor

utófeltétel

egzisztenciális kvantor

szerződés

# Tulajdonságok formalizálásának tapasztalatai 1/2

- **Program kódsorok száma ≈≈ fordítási direktívák** (assertion pragmas, beleértve a ciklus invariánsokat) **+ szerződések sorainak száma**
- A SPARK képességei nem mindenben egyeznek meg az ADA-val
  (pl. egyszerűsített szerződések használata)

# Tulajdonságok formalizálásának tapasztalatai 2/2

- A legtöbb követelmény formalizálása lehetséges (szöveg, táblázat, fékezési görbe egyenlet stb.)
- A SPARK 2014 legkifejezőbb adatstruktúrái (record, array, enumeration stb.) hasznosak más specifikációs nyelvekkel (B, ACSL stb.) való összehasonlításban
- Az említett adatstruktúrák használata könnyen olvasható/értelmezhető specifikációt eredményez

# A formális verifikáció eredményei

**GNATprove és Alt-Ergo (helyességbizonyítási) tapasztalatok**

- Bizonyos eljárások utófeltétele és ciklus invariánsa nem bizonyítható automatikusan az Alt-Ergo használatával
- A bizonyítási összefüggések túl nagyok, és az Alt-Ergo elveszíti az összes lehetséges kvantor feletti uralmat
- **DE:** a ciklus invariánsok néhány része mégis bizonyítható automatikusan (irreleváns hipotézisek kézi beállításával)
- Emellett a helyességbe vetett bizalom növelhető a szerződések és a fordítási direktívák (assertion pragmas) tesztelésével

# Tanulságok összefoglalása 1/5

- **SPARK 2014 kifejezőképessége:**
  - Specifikáció készítés támogatása
  - Egyszerű kódolás
  - Olvasható kód
  - ADA-tól örökölt képességek hasznai az előbbi szempontoknál kritikusak
    (pl. új adattípus összeférhetetlensége más már meglévő típusokkal)

# Tanulságok összefoglalása 2/5

**A kód a helyességbizonyítás szellemében írandó**

- pl. egy függvény, amely kezdetben bizonyíthatatlan volt a ciklusból való korai kilépés (ami a helyes viselkedéshez nem szükséges) miatt a következő formulához vezetett:

$$(\forall\ K.A(K-1) < A(K)) \wedge X < A(1) \rightarrow (\forall\ K.K > 1 \rightarrow X < A(K))$$

- A formula igényli az automatikus helyességbizonyítóban az indukció képességét, azonban ez a legtöbb helyességbizonyítóból hiányzik
- Megoldás a pl. esetében: korai kilépés eltávolítása a kódból (eredménye: Alt-Ergo a bizonyítást elvégezte)

# Tanulságok összefoglalása 3/5/a.

**Az automatikusan bizonyítható szerződések nem mindig a „legtermészetesebb” szerződések**

- Ideális esetben a tool támogatja az automatikusan nem bizonyítható szerződések kézi bizonyítását pl. B módszerrel
- Példa:

# Tanulságok összefoglalása 3/5/b.

Eredeti (nem bizonyítható automatikusan):

```
Post ⇒
  (for all i in Function_Range ⇒
   (Get_Value(Merge, i) =
  Min(Get_Value(SFun1, i), Get_Value(SFun2, i))));
```

minden lehetséges bemeneti érték

Módosított (logikailag ekvivalens az előzővel és automatikusan bizonyítható is):

```
Post ⇒
  (for all i in
      Num_Delimiters_Range'First..Merge.Num_Delim ⇒
   (Merge.Step(i).Value = Min
    (Get_Value(SFun1, Merge.Step(i).Delimiter),
     Get_Value(SFun2, Merge.Step(i).Delimiter))));
```

minden lehetséges határérték

# Tanulságok összefoglalása 4/5

- **A helyes ciklus invariáns felírása** komplex ciklus esetében **nem könnyű feladat**
  - Igények:
    - Helyességbizonyításra alkalmas környezet
    - Szakértő személyzet
    - Idő
    - A ciklus invariánshoz kapcsolódó debug

# Tanulságok összefoglalása 5/5

- **Programozói oktatás speciális fejlesztőkörnyezetben:**
  - Helyességbizonyítás
  - Debug és tesztelési lehetőségek
- **Költségek csökkentése a program teljes helyesség-bizonyításának területén**
  - Definiált munkafolyamatok tervezése
  - A tervezés célja: annak elkerülése, hogy ne töltsenek (túl sok) munkaidőt olyan helyesség bizonyításokkal, melyek várhatóan a kód változtatásában erősen érintettek lesznek

# Felhasznált további források, kitekintés

http://www.spark-2014.org/uploads/erts_2014.pdf

http://www.adacore.com/

http://libre.adacore.com/tools/

http://www.spark-2014.org/

http://docs.adacore.com/spark2014-docs/html/ug/gnatprove.html

http://www.univ-orleans.fr/sciences/info/ressources/webada/doc/spark/ug/usage.html

http://www.open-do.org/projects/hi-lite/

http://alt-ergo.ocamlpro.com/index.php

# Köszönöm a figyelmet!