

SILAB und JDVE

Sven Müller

20.05.2010

Seminar der Projektgruppe StreamCars
DLR
Universität Oldenburg
Department für Informatik
Abteilung Informationssysteme

Einleitung

- Verkehrssimulator vs. Fahrsimulator
- in dieser Präsentation: zwei näher betrachten – SILAB und JDVE

Inhalt

- 1 Einleitung
- 2 Ausgewählte Fahrsimulatoren im Überblick
 - SILAB
 - SiVIC
 - COSMO-SiVIC
 - DOMINION
 - JDVE
- 3 SILAB
 - SILAB
 - DPUs
 - Projekte und Konfigurationsdateien
- Aufbau von Konfigurationsdateien
- Erweiterung/Anbindung
- 4 JDVE
 - JDVE
 - Module
 - Variabler, synchroner Zeitablauf
 - Umweltmodelle
 - Erweiterung/Anbindung
- 5 Zusammenfassung
- 6 Fazit

SILAB

- Entwicklung durch das wiwv
- Einfache Zugänglichkeit
- Konfigurationsdateien
- Modularität durch DPUs
- Auf mehrere Rechner verteilte Simulation
- Verwendung im realen Fahrzeug möglich

SiVIC

- Entwickelt von INRETS-LIVIC
- Fokus:
 - Simulation von Sensoren
 - und des Fahrzeugs
 - so realitätsnah wie möglich
- Ziel: Vereinfachung der Entwicklung von eingebetteten Systemen eines Fahrzeugs
- RT Maps simuliert eingebettetes System
- Durch Zusammenspiel wird das Ziel erreicht
- Technischer Aufbau: Plug-ins

COSMO-SiVIC

- Von INRETS-LESCOT verwirklicht
- Gemäß Anforderungen des EU-Projekts ISi-PADAS
- Weiterentwicklung von SiVIC, gekapselt in Plug-in:
 - Integration des Fahrermodells COSMODRIVE
 - neue grafische Schnittstelle
 - Datenbankanbindung
 - Nutzung mehrerer Programminstanzen
 - Durchführung sekundärer Aufgaben während der Simulation
- Verschaltung mit JDVE angestrebt

DOMINION

- Vom DLR entwickelt
- Fokus:
 - Flexibilität, Anpassbarkeit
 - Realzeitsysteme
 - Sichere, verlässliche eingebettete Systeme entwickeln
 - orientiert an dienstebasierter Softwarearchitektur
- Mittelschicht für Fahrzeuganwendungen
- Technischer Aufbau: Module als eigenständige Programme
- Kommunikation nicht direkt, sondern über gemeinsame Zone
- Datenkern: Gemeinsam genutzte Variablen, Metadaten und Module
- Codegenerierung, Unterstützung diverser Plattformen
- Einbindung von Webdiensten

JDVE

- Entwicklung durch das DLR
- starker Entwicklungscharakter
- Gemäß Anforderungen von ISi-PADAS
- basiert auf DOMINION, nutzt nicht alle Module
- Erweiterungen/Anpassungen:
 - nutzt Instanz der DOMINION-Ontologie
 - Variabler, synchroner Zeitablauf
 - Integration verschiedener Umweltmodelle
- Angestrebt:
 - Verschaltung mit COSMO-SiVIC
 - Integration von Fahrassistenzsystemen

SILAB

- Entwicklung durch das wiwv
- Einfache Zugänglichkeit
- Konfigurationsdateien
- Modularität durch DPU_s
- Auf mehrere Rechner verteilte Simulation
- Verwendung im realen Fahrzeug möglich

DPU_s

- Digital Processing Units
- DPU-Klassen und -Instanzen
- jede DPU-Klasse erfüllt bestimmte Aufgabe
- ist parametrisiert
- besitzt Ein- und Ausgänge
- wird instanziiert
- Verschaltung mehrerer DPU-Instanzen: DPU-Netz

Projekte und Konfigurationsdateien

- Projekt: Simulation, z. B. SILABDemo
- Einstiegspunkt für SILAB in Projekt: Konfigurationsdatei
 - einfache Textdatei
 - .cfg
 - kann weitere Konfigurationsdateien inkludieren
 - Konfigurationsarchitektur

Aufbau von Konfigurationsdateien

Grundlegende Elemente zur Konfiguration des SILAB-Systems:

- System: Globale, allgemeine Einstellungen
- Configuration
 - Computerconfiguration: Konfiguration der einzelnen Rechner
 - DPUConfiguration: Instanziierung der DPUs
 - Connections: Verschaltung der DPUs
- TRF: Allgemeine Angaben zur Verkehrssimulation
- SCN: Definition des Szenarios (Strecke, Umgebung, Verhalten der anderen Verkehrsteilnehmer)

Erweiterung/Anbindung

- DPUs können in C++ geschrieben werden
- Assistent baut dafür Grundgerüst
- Viele, vorgefertigte DPUs:
 - universelle DPUs: Logische Operationen, Einfache Berechnungen, ...
 - DPUScript: Einbindung von einfachen Skripten (Silascript)
 - DPURuby: Ruby-Skripte
 - DPUSocket: Socket-Verbindungen
 - ...
- relativ umfangreiche Dokumentation

JDVE

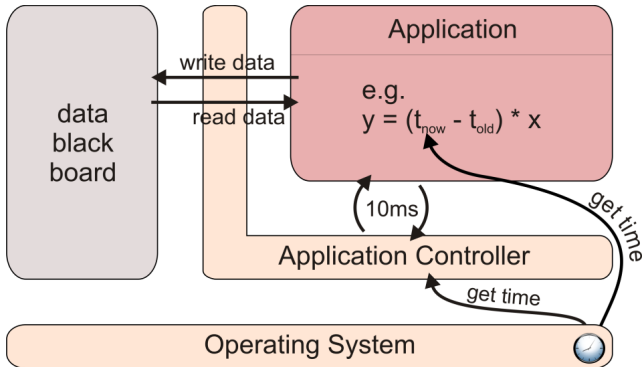
- Entwicklung durch das DLR
- starker Entwicklungscharakter
- Gemäß Anforderungen von ISi-PADAS
- basiert auf DOMINION, nutzt nicht alle Module
- Erweiterungen/Anpassungen:
 - nutzt Instanz der DOMINION-Ontologie
 - Variabler, synchroner Zeitablauf
 - Integration verschiedener Umweltmodelle
- Angestrebt:
 - Verschaltung mit COSMO-SiVIC
 - Integration von Fahrassistenzsystemen

Die Module des JDVEs

- DominionServer
- DominionServerConsole
- EnvironmentManager
- NexGenViewer
- StickTracerWindow
- DominionNLTwotrack
- DynamicObjectsSimulation
- DominionRecord
- Module zur Integration von COSMO-SiVIC und Fahrassistenzsystemen
- isiPADASScheduler2: Variabler, synchroner Zeitablauf

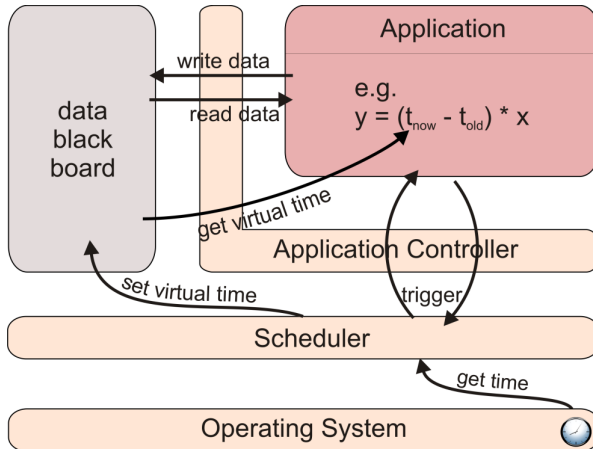
Variabler, synchroner Zeitablauf

Vorher:



Variabler, synchroner Zeitablauf

Nachher:



Umweltmodelle

- passive, statische Umwelt: unbeweglich und unveränderlich
 - Häuser, Verkehrsschilder, ...
 - grafische, aber keine logische Repräsentation nötig
 - Ausnahme: Straßennetz
- passive, aktive Umwelt: unbeweglich, aber veränderlich
 - Ampeln, Verkehrskontrollsysteme, ...
 - grafische und logische Repräsentation
- dynamische Umwelt: beweglich
 - Autos, Fußgänger, ...
 - grafische und logische Repräsentation

JDVE

- wie bei DOMINION
- Auswahl der Module
- Neues Modul erstellen
 - auf Basis der Ontologie bzw. des Datenkerns
 - mittels DominionBAAL: Grundgerüst in C++
 - Vorgefertigte UDP-Funktionalität

Zusammenfassung

- SiVIC
- COSMO-SiVIC
- DOMINION
- SILAB
 - DPUs
 - Projekte und Konfigurationsdateien
 - Erweiterung/Anbindung
- JDVE
 - basiert auf DOMINION
 - Module als Programme
 - Zeitablauf
 - Umweltmodelle

Fazit

Fahrsimulatoren unterscheiden sich im Detail maßgeblich

- je nach Anforderungen
- je nach Entwicklungsstand

Hervortretende Eigenschaften:

- SILAB:
 - Einfacher Einstieg
 - Gute Dokumentation
 - Anpassung der Simulationen problemlos
 - Viel Vorgefertigtes
 - „gereift“
- JDVE:
 - starker Entwicklungscharakter
 - sehr hohe Flexibilität
 - Andere Art der Modulkommunikation
 - Unterstützung verschiedener Plattformen

Danke!

Vielen Dank!

Fragen?

Fragen?

Danke!

Vielen Dank!

Literatur I



T. Bellet (INRETS), P. Mayenobe (INRETS), D. Gruyer (INRETS), J. C. Bornard (INRETS), J. Schindler (DLR), T. Krehle (DLR), Jens Alsen (OFFIS), and Andreas Lüdke (OFFIS).
Joint-DVE Simulation Platform for phase 1.
ISi-PADAS-Bericht, "Deliverable"4.2, Dezember 2009.
Vertraulich.



ISi-PADAS-Newsletter 1.
http://www.isi-padas.eu/newsletter/ISi-PADAS_Newsletter1.pdf, März 2010.
Letzter Zugriff im April 2010.



Ulf Noyer, Marco Hannibal, and Christian Harms.
Dominion Einführung.
Präsentation, Mai 2010.
Präsentation im OFFIS, Oldenburg.



Julian Schindler (DLR), Ulf Noyer (DLR), Christian Harms (DLR), Frank Flemisch (DLR), Aladino Amantini (Kite Solutions), Dominique Gruyer (INRETS-LCPC), Malte Zilinski (OFFIS), Fabio Tango (CRF), and Frank Köster (DLR).
Ontology and Basic Version of the Joint DVE Simulation Platform.
ISi-PADAS-Bericht, "Deliverable"4.1, Mai 2009.
Vertraulich.

Literatur II



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: DPURubyScript.
SILAB-Version 2.5, Stand: Januar 2009.



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: DPUScript.
SILAB-Version 2.5, Stand: Januar 2009.



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: DPUSocket.
SILAB-Version 2.5, Stand: Januar 2009.



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: Erste Schritte.
SILAB-Version 2.5, Stand: Januar 2009.



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: Konfiguration und Bedienung.
SILAB-Version 2.5, Stand: Januar 2009.

Literatur III



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: Ruby API.
SILAB-Version 2.5, Stand: Oktober 2007.



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: Silascript.
SILAB-Version 2.5, Stand: Januar 2009.



Würzburger Institut für Verkehrswissenschaften (wivw), Ansprechpartner: Prof. Dr. Hand-Peter Krüger, Raiffeisenstraße 17, 97209 Veitshöchheim.
SILAB: Standard-Konfigurationsarchitektur.
SILAB-Version 2.5, Stand: Januar 2009.