



## Erfahrungen in der Entwicklung und Nutzung komponentenbasierter Agentenarchitekturen am Beispiel PAXSEM

Daniel Kallfass (Dipl.-Inform.) – System Design Center (COEDS1)  
21.01.2013

# Agenda

I. Einführung & Motivation

II. Einheitliche & modulare Softwareplattform

III. Komponentenbasierte Agentenarchitektur in PAXSEM

IV. Zusammenfassung

## Das System Design Center Abteilung "Studies and Innovative Concepts (COEDS1)"

- Das SDC ist das "**System of Systems**" Kompetenzzentrum für komplexe Systemkonzepte und Architekturen von CASSIDIAN
- **Konzeptstudien und Operational Research** (NT- und F&T-Studien)
- Unterstützung bei **CD&E Vorhaben** und **Beschaffungsprojekten** mithilfe von:
  - Mil. operationelle Beratungsservices (durch ehem. Offiziere der Bundeswehr)
  - Entwicklung und Anwendung von Simulationssystemen (u.a. PAXSEM)
  - Entwicklung und Anwendung von Werkzeugen wie z.B. Data Farming Umgebung
  - Aufbau vernetzter Simulationen (e.g. VIntEL, NCO-Simulationen)



# PAXSEM - Überblick

- Agentenbasiertes 3D-Simulationsmodell
- Entwickelt BAaINBw P2.3 und angewandt u.a. bei **CD&E** und zur **Unterstützung im CPM**
- Schwerpunkt auf physikalischer Abbildung von **Sensoren und Effektoren** im NetOpFü Kontext
- Durchgängige **taktisch-technische Untersuchungen** vom **Data Farming** zum **Testbed** (VIntEL)

## Fragestellung



- ✓ Vernetzung PSISA/HLA/DIS
- ✓ VIntEL Services EBS/WES/CES
- ✓ Szenarioaustausch MSDL
- ✓ Geländedatenbasis SEDRIS/GML
- ✓ FüInfoSysH Anbindung NFFI
- ✓ Visuelle/Thermische Reichweitenmodell (TRM)

## Problemstellung aus der Vergangenheit

- **Unterschiedliche Simulationssysteme** für unterschiedliche Anwendungsfälle
  - Beispiele: IRIS, SMARAGD, PAX, VIPER-3D,....
- **Unterschiedliche Codebasen** bzw. auseinanderlaufende Entwicklungsstränge
- Aufwändige oder **keine Wiederverwendbarkeit** von Funktionalitäten zwischen Projekten
- Begrenzte **Ressourcen** (SW-Entwickler, Zeit, Geld)
- **Modulare Softwarearchitektur**
  - Modulare Funktionalitäten
  - Modulare Oberfläche
- **Komponentenbasierte Agentenarchitektur**
  - Fähigkeiten der Agenten als austauschbare / erweiterbare Komponenten

# Agenda

I. Einführung & Motivation

II. Einheitliche & modulare Softwareplattform

III. Komponentenbasierte Agentenarchitektur in PAXSEM

IV. Zusammenfassung



## SimCore-basierte Applikationen

### Schichtenmodell

- COEDS1 hat eine **einheitliche Software-Plattform** für die Entwicklung von (Simulations-)Applikationen entwickelt
- Aufbauend hierauf sind eine **Vielzahl von Anwendungen** entstanden, teilweise als Produkt

Softwaremodell aus **drei Hauptschichten**:

- Kern **"SimCore"**: Gemeinsame Basis und Infrastruktur für unterschiedliche Anwendungen
- Modulare **Anwendungslogik**
- Konfigurierbare **Oberfläche**



## SimCore-basierte Applikationen

### Applikationskern „SimCore“

- **Gemeinsame Software-Basis** für verschiedene Anwendungen
- **Simulationskern**
- **Gemeinsames Objektmodell**
  - Einfacher Datenaustausch zwischen Modulen
  - Module leichter kombinierbar
- **Allgemeine Funktionalitäten**
- **Oberflächen-Grundgerüst**
- **Plug-In-System** für verschiedene Module



# SimCore-basierte Applikationen

## Anwendungslogik

- **Anwendungslogik** definiert durch **Verknüpfung von unterschiedlichen Modulen** für unterschiedliche Aufgaben
- Module arbeiten alle **auf SimCore-Basis**
- Verknüpfung von Modulen ist **konfigurationsgesteuert**
- Sehr **flexibles** System

Szenareditoren  
2D-Visualisierung  
3D-Visualisierung  
Agentenlogik  
Video-Streaming  
Record / Replay für AAA  
Schnittstellen

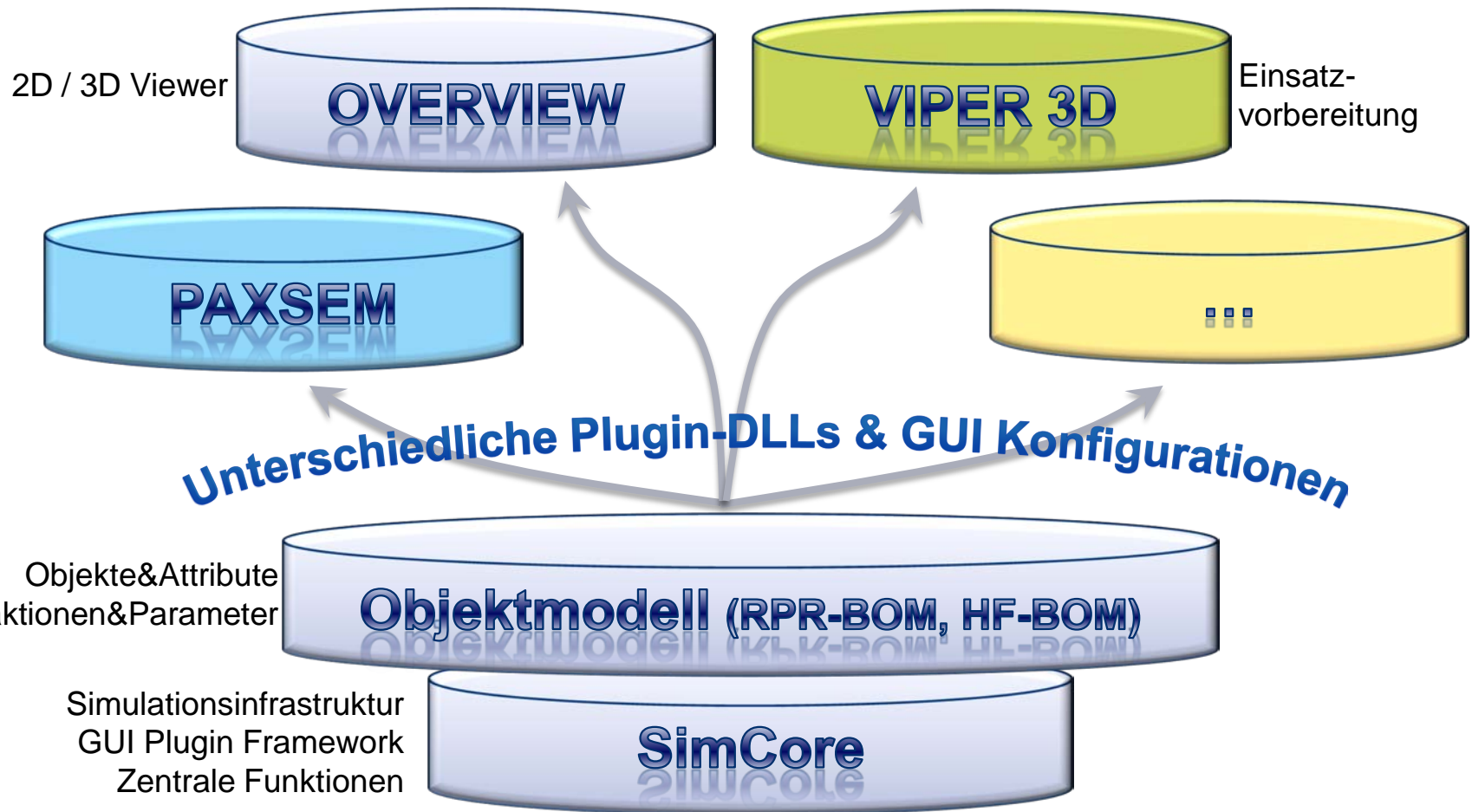


## SimCore-basierte Applikationen Benutzeroberfläche

- Für jeden „Geschmack“ **anpassbar**
- In der Regel arrangiert um eine **zentrale 2D- und/oder 3D-Ansicht**
- Flexibel, da in **Oberflächendesigner** frei definierbar
- **Wiederverwendung** bestehender Elemente
- Abgestimmt für den jeweiligen **Anwendungszweck**

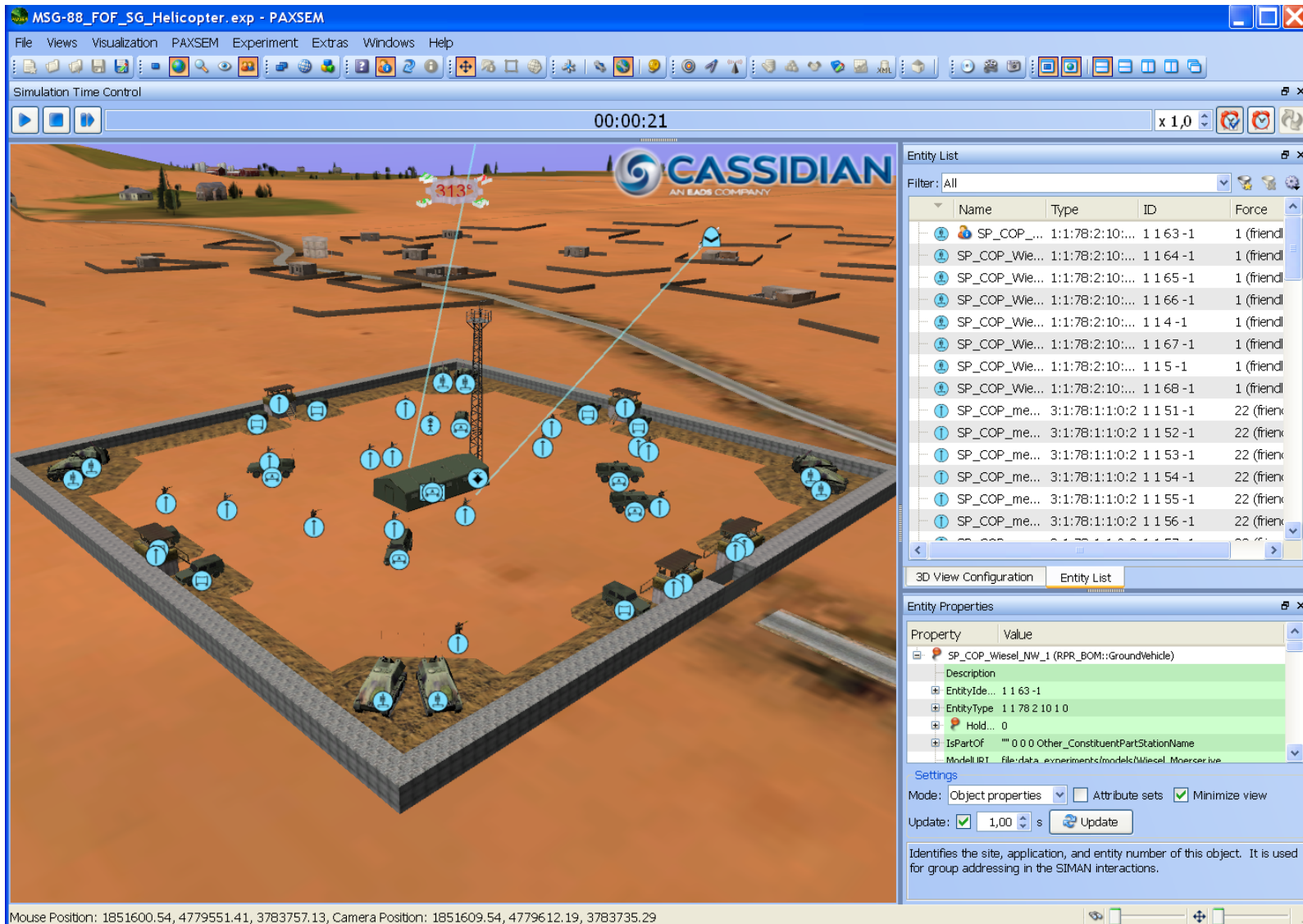
# SimCore-basierte Applikationen

## Anwendungen



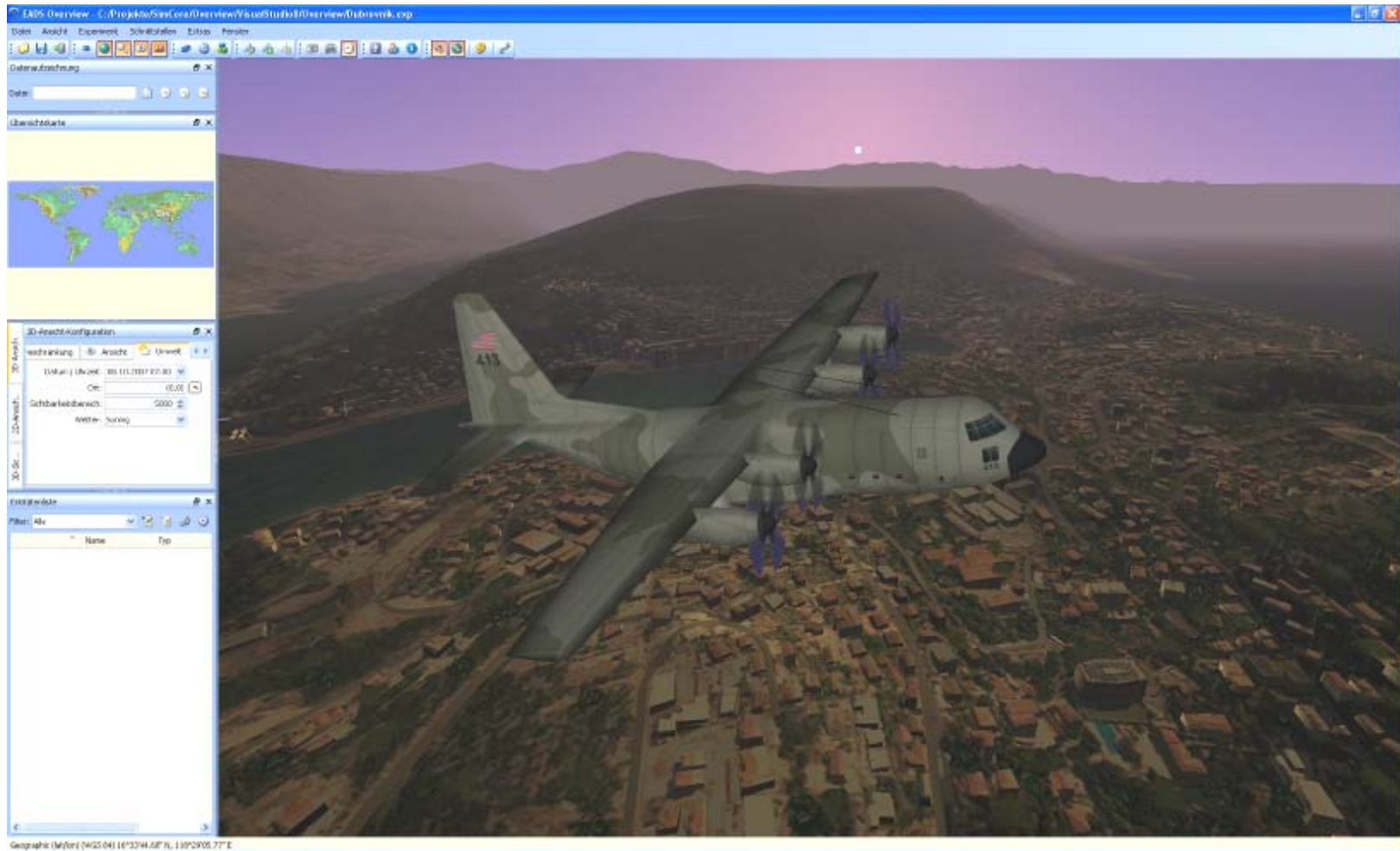
# SimCore-basierte Applikationen

## Benutzeroberfläche – Beispiel „PAXSEM“



# SimCore-basierte Applikationen

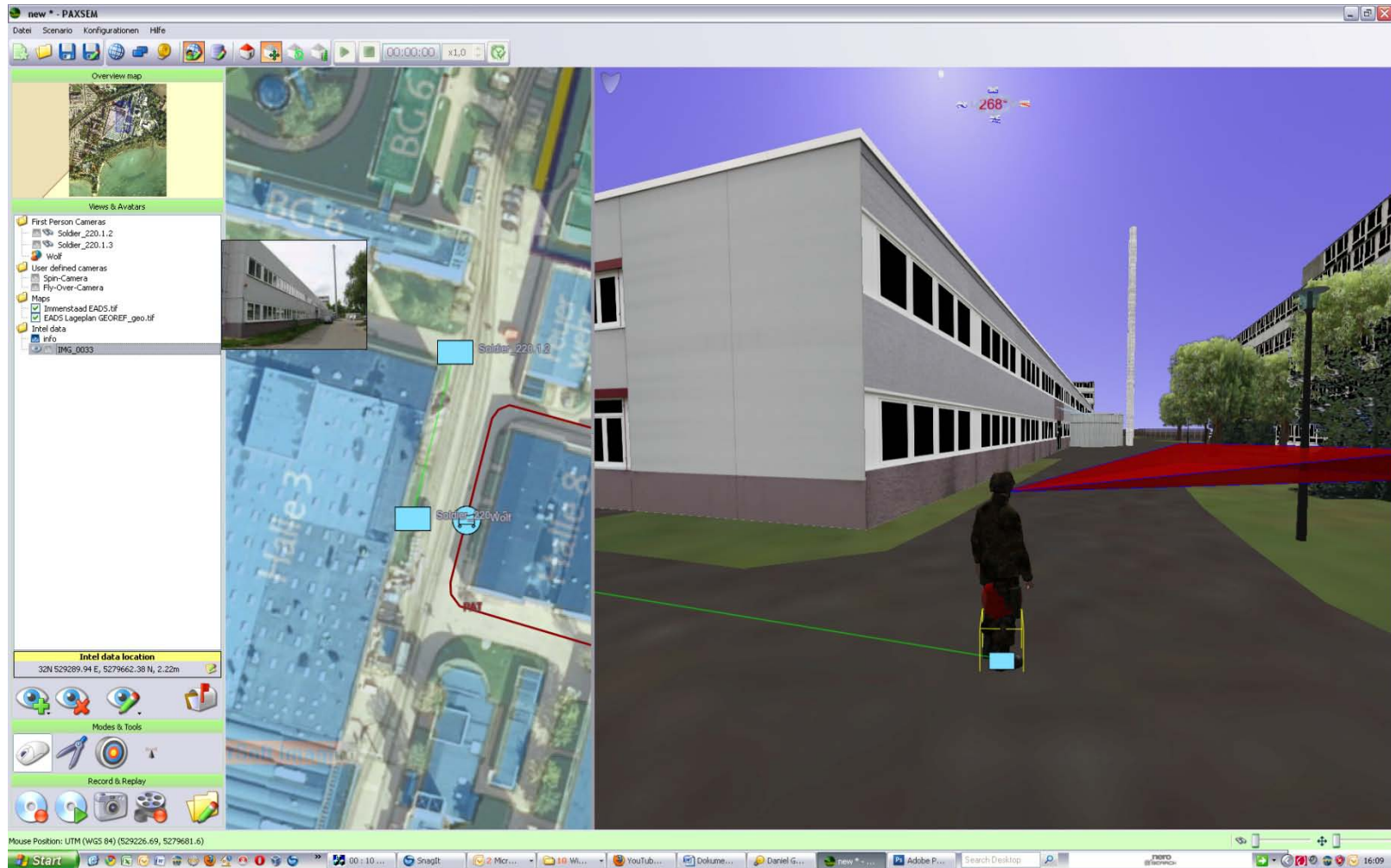
## Benutzeroberfläche – Beispiel „OVERVIEW“ (2D/3D-Viewer)





# SimCore-basierte Applikationen

## Benutzeroberfläche – Beispiel „VIPER 3D“



# Agenda

I. Einführung & Motivation

II. Einheitliche & modulare Softwareplattform

III. Komponentenbasierte Agentenarchitektur in PAXSEM

IV. Zusammenfassung



# Komponentenbasierte Agentenarchitektur

## Problemstellung

- Unterschiedliche Untersuchungsfragen und Anwendungsfälle haben unterschiedliche Anforderungen an das Simulationsmodell PAXSEM

### Beispiele:

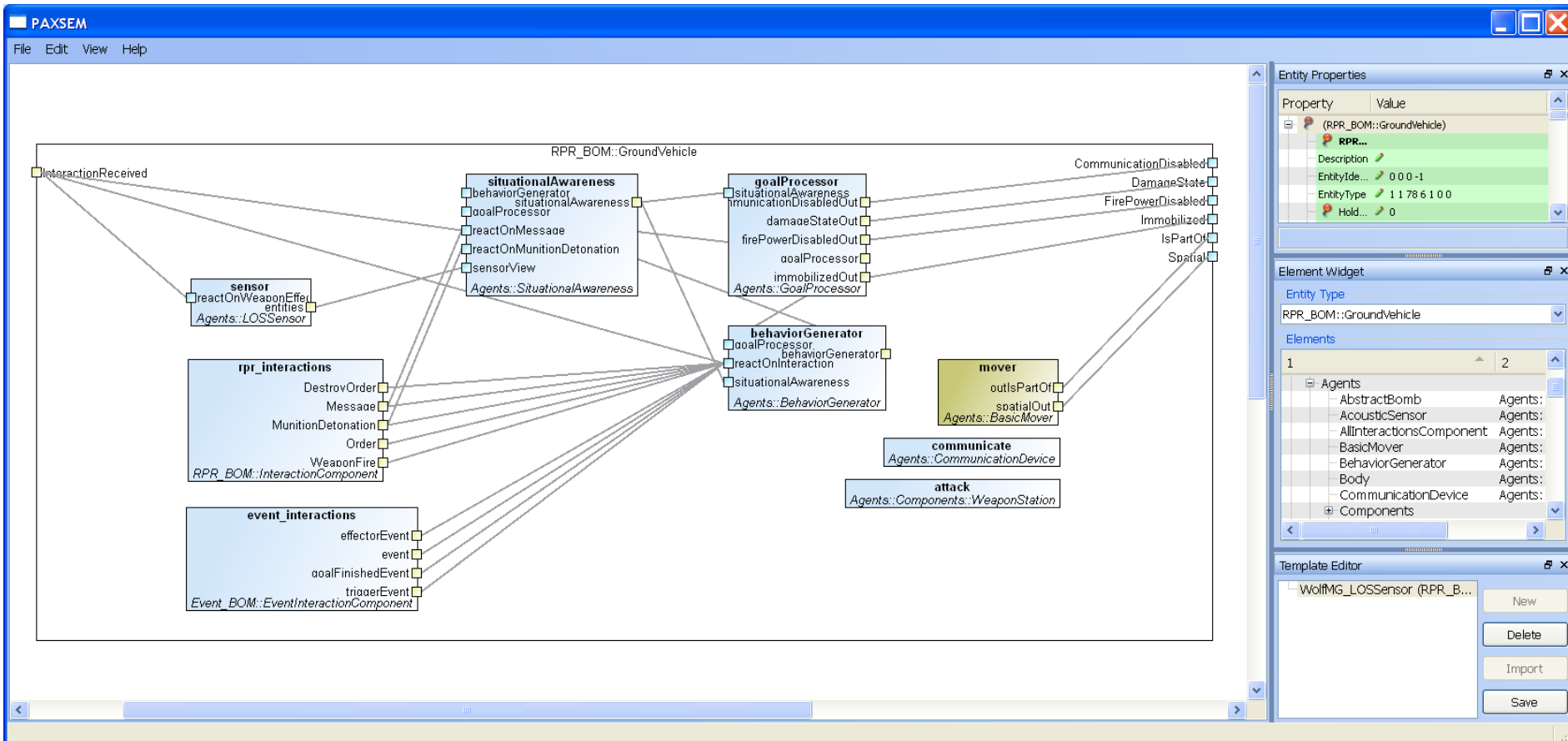
1. **Bewegungsmodellierung** im Data Farming Betrieb vs. im HLA-Verbund mit anderen virtuellen Simulatoren
  - unterschiedliche Detaillierung/Performance der Bewegungsmodellierung
  - unterschiedliche Bewegungskomponenten in PAXSEM
2. **Sensormodellierung**: Simulation der Aufklärungsleistung eines Radarsensors vs. Simulation eines Gefechts im Hinterhalt
  - unterschiedliche Detaillierung/Performance der Sensorkomponente (Line-Of-Sight vs. Radarsensor)
  - unterschiedliche Sensorkomponenten in PAXSEM

# SimCore – Basisfunktionalitäten des Simulationskerns

- RTI-ähnliche Funktionalitäten
- Zentrales Objektmodell
  - Objekten mit Attribute
  - Interaktionen mit Parameter
- Anwendungsmodule (sog. Linklayer) können sich auf bestimmte Objekte oder Interaktionen registrieren (Publish/Subscribe)
- Komponenten können über Datenschnittstellen (sog. IN/OUT-Ports) miteinander verschaltet werden
- Hierüber können Komponenten Attribute von Objekten lesen und schreiben (z.B. Bewegungskomponente beschreibt die Position eines Agenten)
- Ereignis- und zeitgesteuertes Scheduling der Komponente
- Deterministisches Zeitverhalten
- Multithreading-fähig
- Hochperformant in C++ entwickelt

# Komponentenbasierter Agentenaufbau

- Beispiel: Agent mit Sensor-, Effektor-, Kommunikation-, Weltbild und Verhaltenskomponente



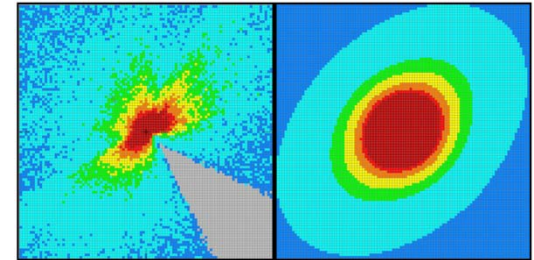
## Erweiterung der Komponenten

- Agent wird über **XML definiert** und konfiguriert
  - Objekttyp (z.B. GroundVehicle aus RPR-FOM)
  - initiale Attributwerten (z.B. Marking)
  - Komponenten (z.B. Bewegungskomponente)
  - Konnektierungen zwischen Komponenten und Attributen
- Erzeugung der Objekte und Komponenten über sog. **Factories**
- Die Komponenten können aus beliebigen **Plugin-DLLs** kommen
- Direkter 1:1 Austausch von Komponenten (z.B. Sensorkomponente) möglich, wenn die **identische Schnittstelle** (Ports) implementiert wurde.

## Erweiterung der Komponenten

### Beispiel: Integration UniVeMo-Verwundbarkeitsmodell

Motivation: Nutzung von detaillierten und eingestuften Wirkdaten aus dem UniVeMo-Verwundbarkeitsmodell der IABG bzw. BAAINBw K1.2



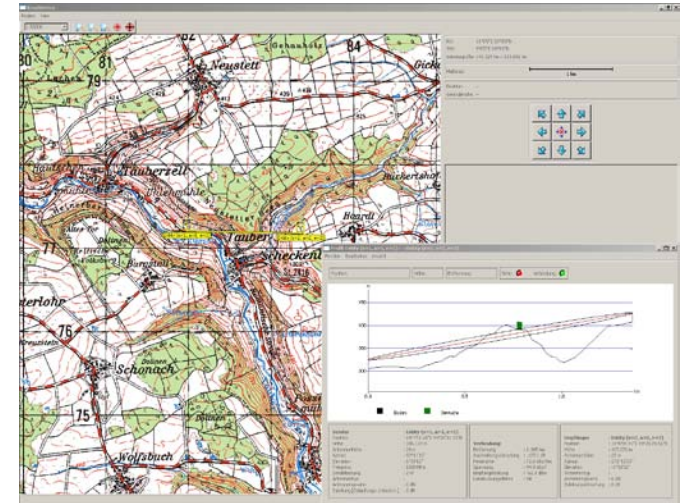
- Die Wirkberechnung in PAXSEM geschieht VIntEL-konform über eine **zentrale Waffenwirkungsentität (WES)** mit entsprechender **Waffenwirkungskomponente**
- Zur Integration wurde eine **neue AHEAD-WES Komponente** mit identischen Schnittstellen geschrieben, die per Plugin-DLL in PAXSEM optional geladen und aktiviert werden kann.
- **AHEAD-WES Komponente** nutzt eine **DLL der IABG** zum Zugriff auf eine **Datenbank mit vorberechneten Verwundbarkeitsdaten** von UniVeMo.
- **Mapping der Datenmodelle** notwendig (Munitionstypen, Zieltypen)
- Sind Ziel/Munitionskombinationen nicht verfügbar, wird als **Fallback auf den internen PAXSEM-WES** zugegriffen

# Erweiterung der Komponenten

## Beispiel: KESS/CES-Proxy für Funkerreichbarkeit

Motivation: Detaillierte Berechnung der Funkerreichbarkeit zwischen zwei Einheiten auf Basis des Tools KESS von Thales

1. Ein **Communication Effect Service (CES)** steht zentral als Webservice bereit (VIntEL-Service)
2. Zur Kommunikation wird anstelle der **CommunicationDevice**-Komponente die **CESRadioCommunicationDevice**-Komponente des Agenten erzeugt und genutzt
3. Die CESRadioCommunicationDevice Komponente **prüft die Funkerreichbarkeit über einen Webservice-Aufruf** an einen zentralen CES-Service
4. Der **CES berechnet die Funkerreichbarkeit** und gibt das Ergebnis zurück.



# Agenda

I. Einführung & Motivation

II. Einheitliche & modulare Softwareplattform

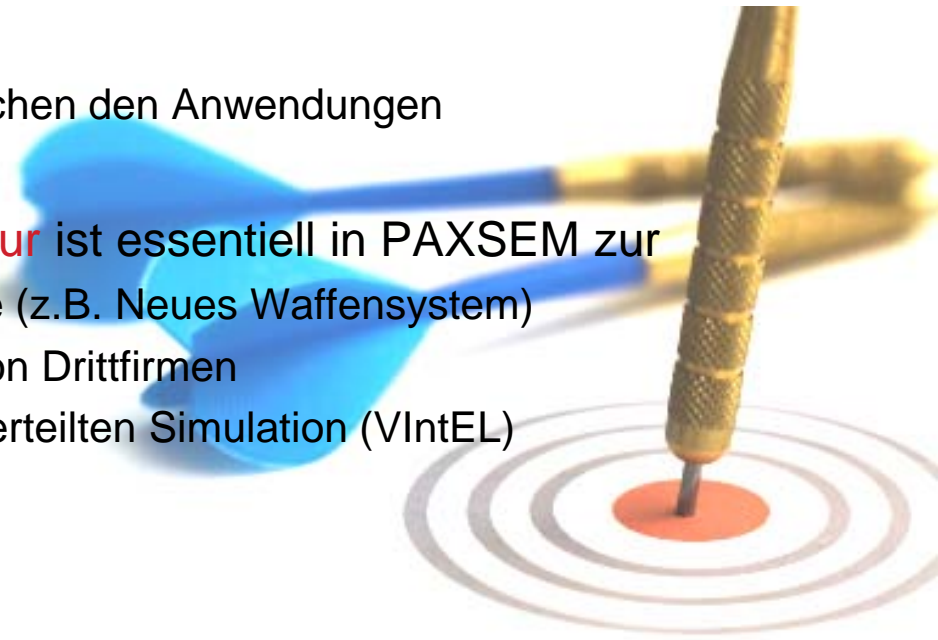
III. Komponentenbasierte Agentenarchitektur in PAXSEM

IV. Zusammenfassung



# Zusammenfassung

- Breites Anwendungsspektrum des Simulationssystems PAXSEM erfordert **Agilität bei Funktionserweiterungen**
- **Modulare Softwarearchitektur**
  - Hochgradig anpassbar (Module, GUI)
  - Schnelle Entwicklung weiterer Module
  - Wiederverwendbarkeit der Module zwischen den Anwendungen
- **Komponentenbasierte Agentenarchitektur** ist essentiell in PAXSEM zur
  - einfachen Erweiterung um neue Modelle (z.B. Neues Waffensystem)
  - Integration von existierenden Modelle von Drittfirmen
  - Nutzung von externen Services in der verteilten Simulation (VIntEL)



Thank you for your attention!

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization is prohibited. Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.