



Visual Active Memory Processes and Interactive REtrieval

Einsatz von XML Technologien zur Systemintegration im VAMPIRE Projekt

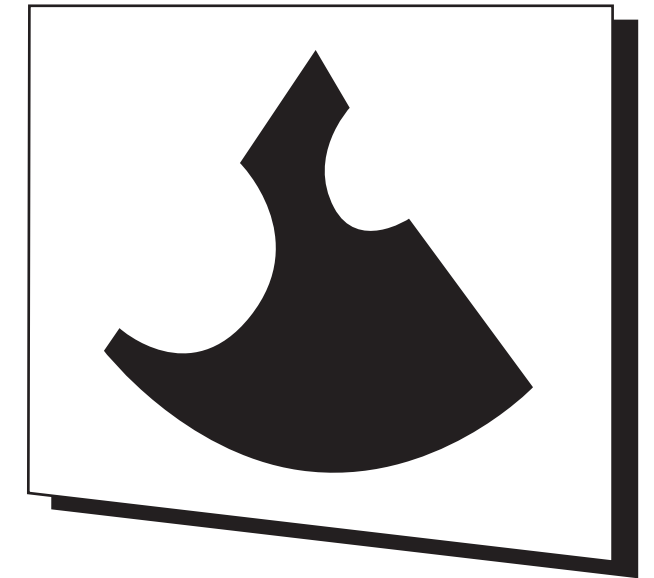
Jan Schaefer

Highspeed Kamera
und Trägheitssensor



Kameras in
Augenabstand

Head-Mounted-Display



Cheesetarget

Verteiltes System



Überblick

VAMPIRE: Idee

VAMPIRE: Partner und Module

Integration: Ziele

Integration: XCF Framework

XCF: Highlevel-Communication Framework

XCF: Active Memory

XCF: Memory Processes

Integration: Ziele erreicht?

VAMPIRE: Action!!!

VAMPIRE: Idee

Visual **A**ctive **M**emory **P**rocesses and **I**nteractive **R**etrieval

Brille mit Gedächtnis

Hier nicht behandelt: Interactive Retrieval

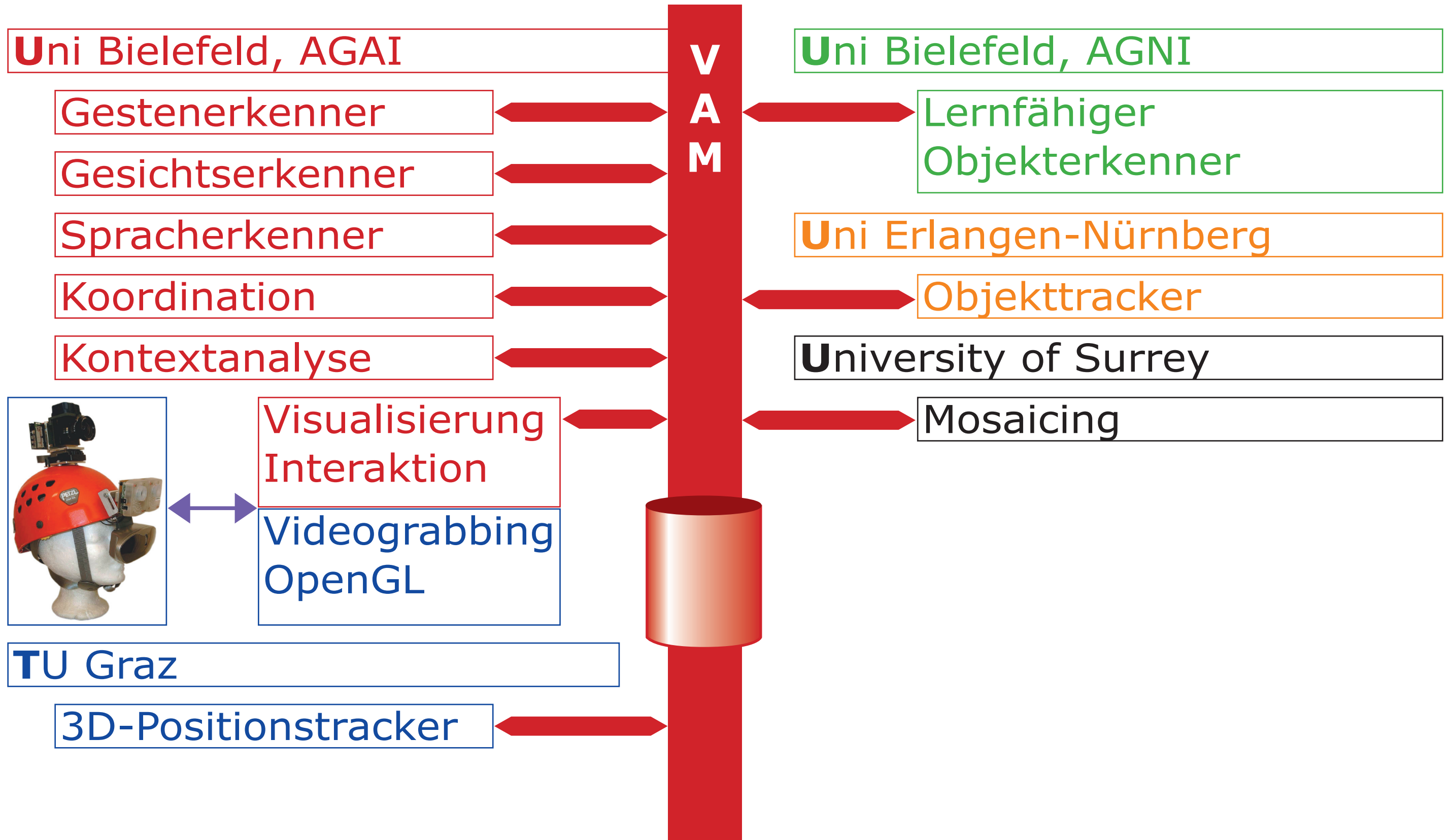
Ein Integrationsprojekt (unter anderem)

Zentrale Idee: VAM als Integrationsknoten für
Daten- und Kontrollfluss

Anlehnung an Gehirn: Kontextbezogene Wahrnehmung

Mensch sieht was er sehen will

VAMPIRE: Partner und Module



Integration: Ziele

Datenfusion und Koordination durch VAM

Möglichst geringe Einstiegsschwelle

Usability, da Modul-Entwickler oft keine Softwareentwicklungsprofis

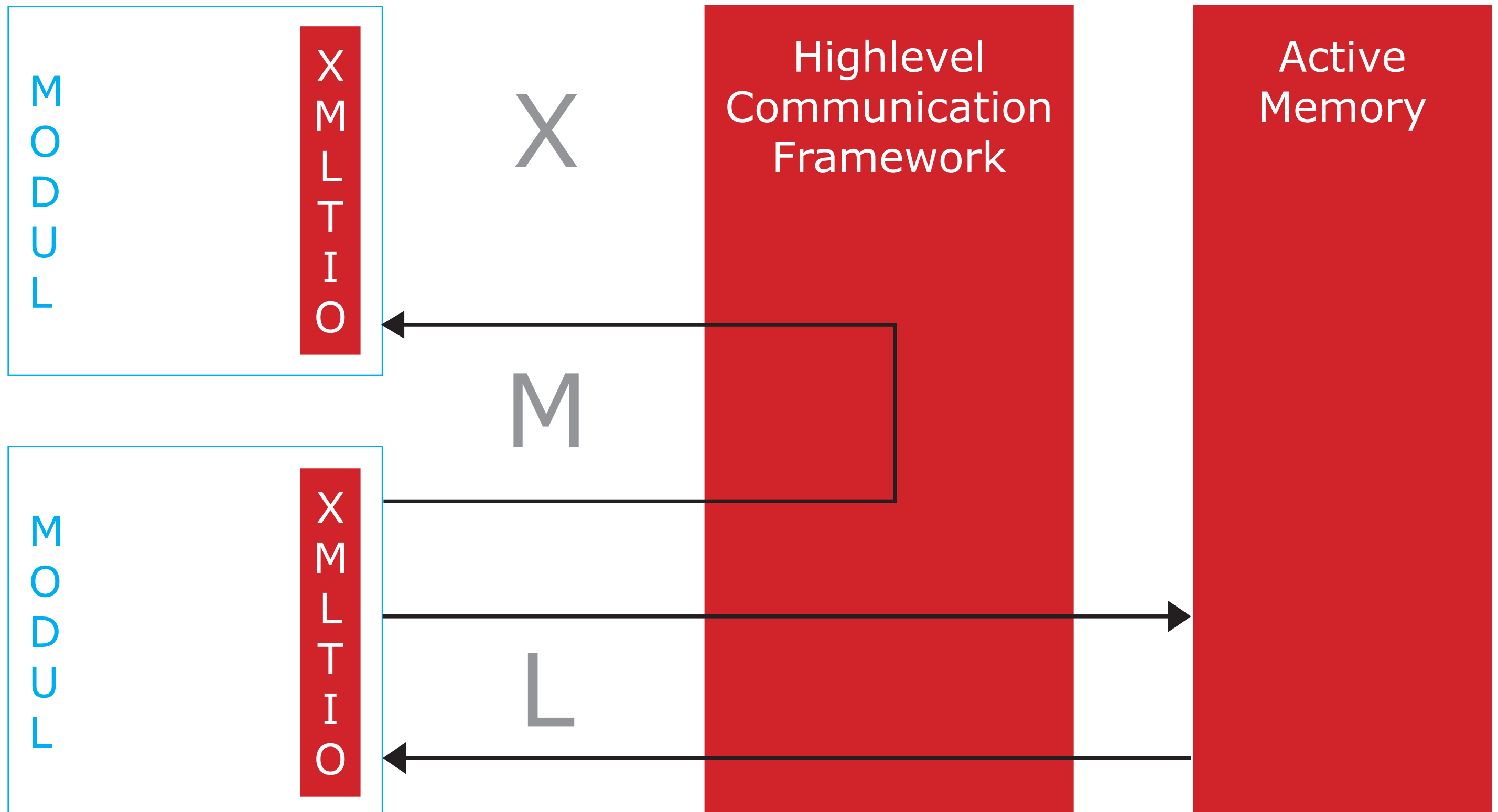
Möglichst einfache und verständliche Konzepte

Große Flexibilität, da viele unterschiedliche Module

Einfache Analyse/Debugging des laufenden Systems

Erweiterbarkeit

Integration: XCF Framework



XCF: Highlevel-Communication-Framework

Message-Oriented-Middleware (abgespeckt)

Basiert auf ZeroC's Internet Communications Engine (ICE)

Implementiert Entwurfs-/Kommunikationsmuster
Publisher-Subscriber, Client-Server

Für Entwickler extrem einfach zu integrieren

Es werden XML-Dokumente übertragen

XOP-artige Übertragung von Binärdaten

XCF: Active Memory

XML-Datenbank-Server basiert auf Berkeley DB XML

Zugriff durch Communication-Framework (API-gekapselt)

Datenzugriff anhand von XPath

Intrinsic/Extrinsic Memory Processes (IMP/EMP)

Eventhandler registriert auf XPath/DB-Operation
z.B. („/OBJECT“, INSERT)

Eventhandler erhält auslösende XML-Source

Eventhandler wird bei passendem DB-Zugriff ausgelöst

Daher Daten- und Kontrollfluss vereinigt

XCF: Memory Processes

IMP

Synchrone serverseitige Ausführung

Direkter DB-Zugriff

Transaktionssicher (Bei Fehler Rollback)

XML-Deklaration gespeichert im Memory

EMP

Asynchrone Ausführung im verteilten System

DB-Zugriff durch Communication-Framework

Implementierung beliebig, da clientseitig

XCF: Beispiele für Memory Processes

IMP-Beispiel: Vergessen

Wertet XML-Schnittstelle Hypothesis aus

Alte Nodes mit geringer Reliability werden gelöscht

Prüft alle Nodes mit Hypothesis-Child: Viele Daten

EMP-Beispiel: Konsistenzprüfung

Prüft die Konsistenz von Nodes untereinander (Bayes-Netze)

Setzt Reliability herab, wenn Node keinen Sinn macht

Inkonsistente Nodes werden dann ‚vergessen‘

Prüft nur bestimmte Nodes (z.B. Objekte): Weniger Daten

XCF: XML-Schnittstellen-Modellierung

XML-Struktur sollte Informationen reflektieren, nicht Daten

Vordefinierte XML-Vokabularien für Vision-Systems,
z.B. XML-Struktur für Objekterkennungsergebnisse

Validierung durch XML-Schema möglich

Information	Daten (XML-RPC)
<pre><OBJECT> <CLASS>CUP</CLASS> <CENTER x="32" y="44"/> <REGION> <RECT x="13" y="27" w="80" h="80"/> </REGION> </OBJECT></pre>	<pre><member> <name>CENTER</name> <value> <struct> <member> <name>y</name> <value><int>44</int></value> </member> <member> <name>x</name> <value><int>32</int></value> </member> </struct> </value> </member></pre>

Integration: Ziele erreicht?

XML ist flexibel und erweiterbar

XML-Schnittstellen leicht lesbar und verständlich
(Introspektion, Einstiegsschwelle, Planung, Diskussion)

Überall einheitliches Datenformat, einheitlicher Datenzugriff
(Einstiegsschwelle, Fusion)

Events können an XPath gebunden werden
(Koordination)

Sehr unkomplizierte XCF-Schnittstellen
(Usability, Einstiegsschwelle)

Aber: XML-Schnittstelle muss definiert werden

VAMPIRE: Action!!!

Szenario: Cocktails mixen

User A bringt VAMPIRE Cocktailrezept bei (Zutaten, Ablauf)

VAMPIRE unterstützt **User B** bei Zubereitung des Cocktails

Film

Danke für Eure Aufmerksamkeit!
Noch Fragen?

Diskussion

Ist die Vereinigung von Daten- und Kontrollfluss gut?

Kann zuviel Flexibilität der Integration auch schaden?

Begünstigen XML-Schnittstellen chaotische Verhältnisse?