



Software Plattform Embedded Systems 2020

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

**Beschreibung der exemplarischen Umsetzung eines Modellierungsansatzes für
embedded Grafiksysteme von XiSys Software GmbH-E4You:**

- Datenzentrisches Modell -
- Visualisierung -
- Prototypenhafte Umsetzung mit XiBase9 -
- Fallstudie: „High Lift System“-

Version: 2.1

Projektbezeichnung	SPES 2020 AT	
Verantwortlich	Klaus Gerstendörfer, XiSys Software GmbH - Embedded4You e.V.	
QS-Verantwortlich	Robert Schachner, RST Industrie Automation GmbH, Embedded4You e.V.	
Erstellt am	Dez. 2011	
Freigabestatus	<input type="checkbox"/>	Vertraulich für Partner: entfällt
	<input checked="" type="checkbox"/>	Projektöffentlich
	<input type="checkbox"/>	Öffentlich
Erzeugung	Klaus Gerstendörfer, XiSys Software GmbH, Embedded4You e.V.	
Mitwirkend	Peter Schuller, MicroSys Electronics GmbH, Embedded4You e.V.	

Inhalt

1	Einordnung und Beschreibung	3
1.1	Übersicht der Fallstudie von Embedded4You	3
1.2	embedded Grafik mit Modellansatz innerhalb der Domäne	3
2	Forschungsgegenstand: Embedded Grafik und Datenmodelle	5
2.1	Übergeordnete Zielsetzung.....	5
2.2	Die Ausgangsbasis XiBase9, ein konzeptioneller Überblick	6
2.3	Übersicht: Modellierungsansatz für eine embedded Grafik	6
2.4	Ein datenzentrisches Beschreibungsmodell als zentrales Bindeglied.....	7
3	Beschreibung der Forschungsarbeiten.....	9
3.1	Methodische und funktionale Integration in das datenzentrische Beschreibungsmodell.....	9
3.2	Umsetzung der Ergebnisse aus 3.1 im Grafiksystem	10
3.3	Exemplarische Umsetzung der Modellierung am Beispiel eines Avionik-Geräts	12
4	Zusammenfassung der Ergebnisse	16
5	Einsehbare Projektdokumente	17

1 Einordnung und Beschreibung

1.1 Übersicht der Fallstudie von Embedded4You

Die von E4You vorgeschlagene Fallstudie **Teststand für High Lift Flügelstrukturen**“ betrachtet heutige Anforderungen moderner Test- und Erprobungsumgebungen für Flügelstrukturen und untersucht methodische Ansätze für Domänen übergreifende Modellierungen künftiger Lösungen dieser Anwendungskategorie – Domäne -.Der zu untersuchende modellbasierte Forschungsansatz dieser komplexen Aufgabe fasst die Einzelbeiträge der E4You-Mitglieder Elma Trenew GmbH, FTI Engineering Network GmbH, IMACS GmbH, N.A.T. GmbH, MicroSys Electronics GmbH, RST Industrie Automatisierung GMBH und XiSys Software GmbH zusammen. Er steht exemplarisch für die sich ergänzenden einzelnen Kompetenzbereiche der einzelnen beteiligten E4You-Partner. Sie übernehmen dabei gemeinsam eine Aufgabe, die zu einer integrierten komplexen Lösung führen soll, die für eine einzelne Firma in dieser Form nicht anzustreben ist. Die E4You-Fallstudie ist dem Bereich SPES-AT1 AP zugeordnet, denn in ihren Teilaspekten stellt sie ein typisches Beispiel eines modellbasierten Lösungsansatzes für Probleme der heutigen Anforderung auch in der Automatisierungstechnik dar.

Die Basis des E4You Forschungsansatzes ist hierbei, ein Abstraktionsmodell der unterliegenden heterogenen, computergestützten Systemstrukturen für die Steuerung der Anlage und die Datenerfassung über die Sensorik zu untersuchen. Dieses Modell ist die weitere Grundlage für eine „Hardware in the Loop Simulations- und Testumgebung.“ Exemplarisch soll ein Prototyp/Demonstrator entstehen, auf dem die Nachweise für eine durchgängige Entwicklungs- und Testumgebung geführt werden können. Gleichzeitig sollen die Arbeiten in Bezug zum SPES-2020-Metamodell gesetzt werden.

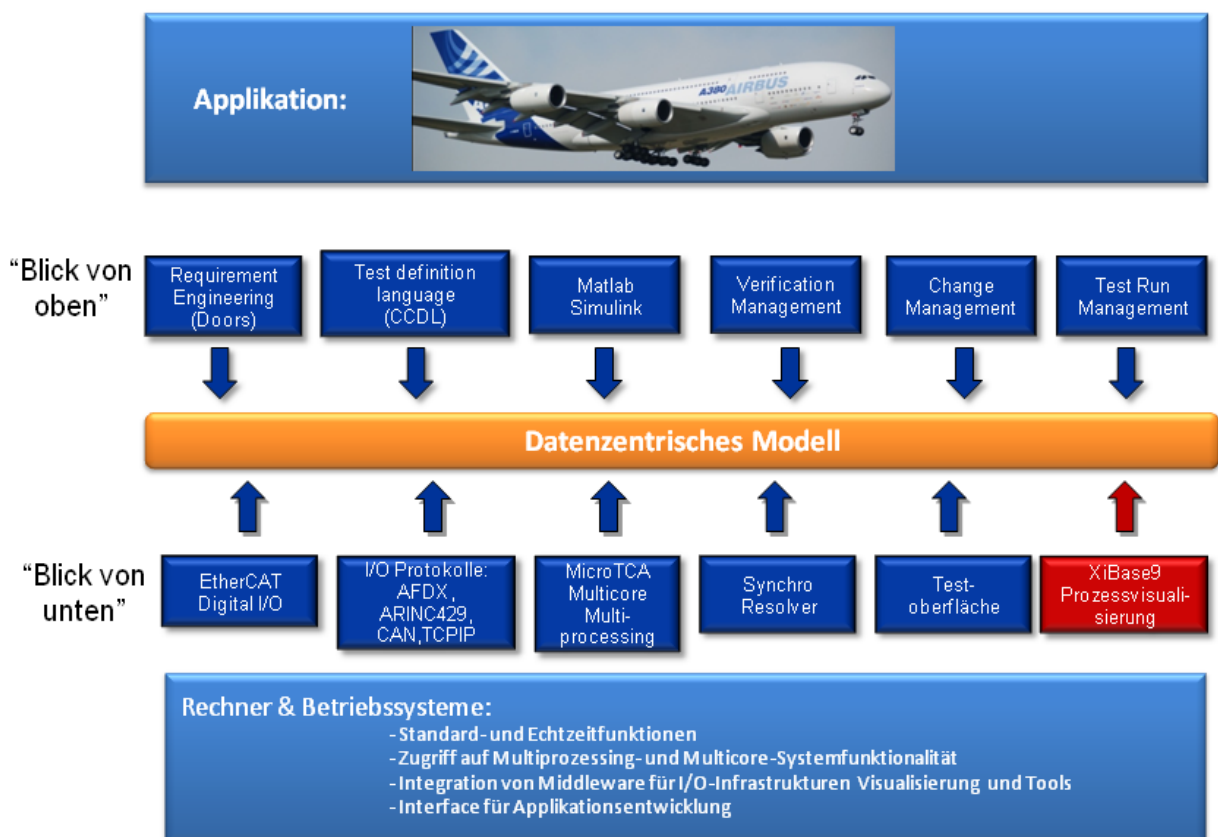
In Abb. 1 sind die Teilaspekte der Domäne grafisch dargestellt. Zwei Betrachtungen sind für das Gesamtkonzept wesentlich: Der „Blick von oben“ konzentriert sich auf die funktionalen Aspekte der Domäne hinsichtlich der notwendigen Lösungsansätze, um die gesamte Aufgabenstellung aus Anwendungssicht zu definieren. Der „Blick von unten“ fasst die realen embedded Hard- und Softwarekomponenten zusammen, die für die funktionale Ausführung der Aufgabe notwendig sind. Das Datenmodell, als zentrale Verbindungsschicht, abstrahiert und integriert gleichzeitig die Komponenten der gesamten Domäne. Das Ziel soll sein, von der Projektdefinition über die System- und Geräteauslegung bis zur Integration mit Test und Verifikation mit dem modellorientiert Ansatz den gesamten Engineering-Prozess zu vereinfachen.

1.2 Embedded Grafik mit Modellansatz innerhalb der Domäne

XiSys spezialisiert sich in diesem Umfeld unter anderem auf die Erforschung, Umsetzung und Bereitstellung geeigneter Modellierungsansätze für anspruchsvolle embedded Grafiksysteme. Mit der Grafikplattform XiBase9 stand eine Ausgangsbasis zur Verfügung, anhand der die Modellierungsansätze untersucht und exemplarisch der Domäne für die Prozessvisualisierung zur Verfügung

gestellt wurden (in Rot hervorgehoben in Abb. 1). Die Visualisierung steht natürlich nicht im zentralen Fokus des Gesamtmodells. Sie repräsentiert allerdings die wesentliche Funktion der grafischen Interaktion zwischen dem Bedienen und der Prozessvisualisierung auf der oberen Ebene und dem Zusammenspiel mit den durch die Hardware und hardwarenahe Software vorgegebenen Randbedingungen der „Sicht von unten“.

Abb. 1: Struktur der Domäne: Fallstudie Teststand für High Lift Flügelstrukturen“ und Einordnung der Prozessvisualisierung



Folgende Fragestellungen mussten in diesem Umfeld deshalb untersucht werden:

- Struktur und Abbildung eines geeigneten Modells gekoppelt mit dem datenzentrischen Ansatz der Domäne
- Modellierung der internen Funktionen Objektgestaltung, bzw. –design und Objektanimation
- Abbildung physikalisch-mathematischer Funktionen im Modell

Ein wesentlicher Aspekt der alle drei Teilbereiche der Forschungsarbeiten betraf, war die Untersuchung und Umsetzung von Methoden zur Vereinfachung und Beschleunigung des Entwicklungs-

prozesses komplexer embedded Grafikanwendungen. Die Fragen, die hier zu betrachten waren, sind:

- Reduktion aufwendiger und wiederkehrender Programmierung von Objekten und Bildschirmmasken
- Vereinfachung der Integration in komplexe heterogene Systemumgebungen: Hardware, Betriebssystem, Anwendung, Vernetzung, Umlenkung von grafischen Ein- und Ausgaben auf verschiedene Plattformen.
- Ein geringer Systemressourcenbedarf für embedded Anwendungen und Unterstützung geeigneter Hardwareplattformen, die in diesem Umfeld typischerweise mit PowerPC-, x86- oder kompatiblen, ARM-, MIPS- oder SH4 – CPUs ausgestattet sind.

Hinsichtlich Ergonomie und Bedienbarkeit wurden folgende Aspekte miteinbezogen:

- Funktionsgerechte Darstellung von Systemprozessen für intuitive Interaktionen mit dem Anwender über animierte Objekte. und/oder „Touch“-Bedienung.
- Grafische Darstellung vieler Systemabläufe und- Funktionen in überlagerten Fenstern ohne zu verwirren.
- Internationale Sprachunterstützung ohne explizite Programmierung mit einfachem Bedienkonzept.
- Moderne Bedienkonzepte mit Touch-Unterstützung, transparenten Fenstern, Antialiasing, Schatten usw.

2 Forschungsgegenstand: Embedded Grafik und Datenmodelle

2.1 Übergeordnete Zielsetzung

Die Verbindung von Visualisierung mit Datenmodellen für embedded grafische Benutzerumgebungen, ist ein innovativer Ansatz, mit dem das Erstellen von komplexen grafischen Funktionen drastisch reduziert werden kann. Diese Kombination verspricht eine völlige Entkoppelung der Programme, die auf der einen Seite die Maschine steuern und auf der anderen Seite zur Darstellung der Maschinendaten nötig sind. Es war unter anderem zu untersuchen, wie grafische Oberflächen ohne, bzw. mit möglichst wenig traditioneller Programmierung von Grafikdesignern gestaltet werden können. Animierte Objekte sollen für die Darstellung und Manipulation von Prozess- und Systemdaten in dem Grafiksystem durch Rückgriff auf das unterliegende Datenmodell nur noch konfiguriert werden. Aufwendige explizite Programmieraufgaben sollen dadurch größtenteils entfallen. Damit einhergehend sind, vereinfachte und beschleunigte Entwicklungsprozesse der maßgebliche Nutzen für die spätere kommerzielle Verwertung der Forschungsergebnisse. Exemplarisch soll an

einem Versuchsträger (beispielhafte Implementierung) der Modellansatz innerhalb der E4You-Domäne umgesetzt und überprüft werden. Zusätzlich sollen die neuen Methoden hinsichtlich ihrer Integrationsfähigkeit in das SPES2020-Metamodell überprüft und verifiziert werden.

2.2 Die Ausgangsbasis XiBase9, ein konzeptioneller Überblick

XiBase9 ist ein betriebssystemunabhängiges und objektorientiertes Grafiksystem. Grafikobjekte werden mit einem komfortablen Vektoreditor erstellt oder durch Modifikationen bereits vorhandener Objekte entsprechend angepasst. Ein Grafikserver kommuniziert über eine fest definierte Schnittstelle mit den Systemanwendungen und ist für die Darstellung der Objekte mit ihren Funktionen verantwortlich. Die grafischen Funktionen für die Objektdarstellung sind völlig entkoppelt von dem Anwendungsprogramm. Änderungen an der grafischen Oberfläche können somit einfach umgesetzt werden, ohne den Programmcode der Anwendungen modifizieren zu müssen. Dies führt zu übersichtlichen Codestrukturen und besserer Wartbarkeit der Software im produktiven Einsatz. XiBase9 wird bereits seit Langem in embedded Systemen mit hohen Anforderungen an die Zuverlässigkeit und Echtzeitfähigkeit bei gleichzeitig minimalem Ressourcenbedarf eingesetzt. Individuelle oder gewohnte, „windows-ähnliche“ grafische Oberflächen sind mit verfügbaren Entwicklungswerkzeugen einfach zu erstellen und mit seiner Internationalität praktisch für alle wichtigen westlichen, asiatischen und arabischen Anwendungen einsetzbar.

2.3 Übersicht: Modellierungsansatz für eine embedded Grafik

Mit der Einführung einer Modellierung war ein Paradigmenwechsel im Softwaredesign von embedded Grafiksystemen zu untersuchen. Was ist damit gemeint?

Zwei Punkte sind ausschlaggebend dafür, diesen Paradigmenwechsel zu beschreiben. Zum einen ist es die zunehmende Komplexität heutiger und künftiger embedded Systemlösungen, die strukturiert und übersichtlich beherrscht werden soll. Das vermeidet die bekannten Datengräber und führt beim Kunden zur Reduktion auf die Kernkompetenz seiner Entwicklungsarbeit. Zum anderen soll und musste die einfache Wiederverwendbarkeit Forschungsgegenstand sein. Vorhandene und künftige Systembausteine sollen sozusagen in Komponentenform, baukastenähnlich einsetzbar und wiederverwertbar sein. Es ist zwingend notwendig, aufwendige Programmierung einfacher zu gestalten bzw. gänzlich zu vermeiden. Hierdurch wird die Komplexität der Aufgaben leichter beherrschbar. Außerdem reduzieren sich die Entwicklungszeiten bei gleichzeitig verringertem Wartungsaufwand für die Software. Mit der Modellierung wurde untersucht, wie idealerweise aus einem formalen Modell möglichst automatisch Grafikobjekte und -funktionen zu erzeugen sind, d.h., es mussten dafür neue Methoden und Werkzeuge definiert und exemplarisch erstellt werden.

Darüber hinaus muss es möglich sein das Werkzeug mit den Technologien des Demonstrators und den Anforderungen in der Automatisierungstechnik die durch die Firma IMACS (E4You-SPES2020 Projektpartner) repräsentiert werden zu kombinieren.

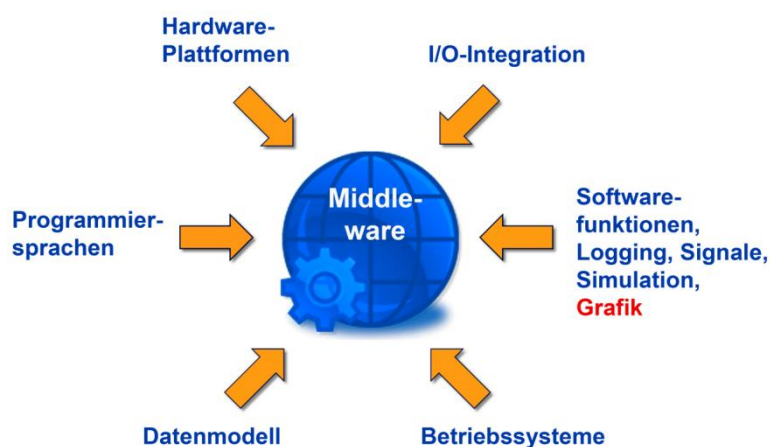
2.4 Ein datenzentrisches Beschreibungsmodell als zentrales Bindeglied

Die Firma RST Industrie Automation konzentriert sich im Rahmen der Forschungsarbeiten auf die Untersuchung eines geeigneten Domänen übergreifenden, datenzentrischen Beschreibungsmodells. Es steht als zentrale Abstraktionsschicht der Domäne zur Verfügung und wird auf der Basis der Middleware „GAMMA“ exemplarisch umgesetzt.

Das Modell abstrahiert, konfiguriert und integriert (siehe Abb.2):

- Gesamte Hardwareplattformen, inkl. CPUs, BUS-, Speicher- und lokales I/O-System
- Betriebssysteme
- Kommunikations- und Netzwerkinfrastrukturen: TCPI/IP, Feldbusse wie CAN, EtherCAT, ProfiNET, etc
- Programmiersprachen
- **Embedded Grafiksysteme**
- Grafische Modellierungsumgebungen, wie z. B. MatLab-Simulink
- Heterogene Rechnerarchitekturen, auch in verschiedenen Netzwerken (meist TCP/IP) integriert
- SPS-Systeme
- Software-Know-how in Form von Komponenten, das kann domänenspezifisch aber auch übergreifend sein. In diesem Aspekt kommt speziell der Schwerpunkt Code-Wiederverwendbarkeit zum Tragen.
- Über OPC (OLE for process control) -Schnittstellen weitere Techniken, wie z. B. Prozessvisualisierungen

Abb. 2: „Middleware -datenzentrisches Beschreibungsmodell-“



Das Ziel der Modellierung ist es, den Entwicklungsprozess im speziellen hier auch die Visualisierung von aufwendigen Betriebssystemadaptionen, Treiberanpassungen für Peripherie, I/O-Funktionen und der Integration von Netzwerkfunktionen auf der Basis zeitraubender „C“-Programmierung zu entheben. Ein wesentlicher Bestandteil der Middleware ist dabei eine datenzentrische Organisation der Systemvariablen durch ihre Abbildung in einem Datenmodell.

Der Forschungsansatz bezüglich der embedded Grafikumgebung bezog sich vor diesem Hintergrund im wesentlichen auf zwei große Teilaspekte:

- Methodische und funktionale Integration in das datenzentrische Beschreibungsmodell
- Untersuchung geeigneter formaler Methoden den in 2.3 beschriebenen Modellierungsansatz zu erfüllen.

2.5 Visualisierung: Modellierung der dynamischen Prozesse

Neben der Integration des Grafiksystems in das Datenmodell der Domäne bestand eine zweite zentrale Aufgabe der Arbeiten darin, durch einen neuen Modellierungsansatz das Erzeugen von dynamischen Prozessabläufen in und mit Grafikobjekten wesentlich zu vereinfachen. Hier musste Neuland betreten werden. Werkzeuge für das Design grafischer Oberflächen und Objekte sind zahlreich am Markt vertreten, somit wurde auf in diesen Bereich kein weiterer Fokus gelegt. Der Zugriff, Austausch und die Interaktionen der grafischen Objekte mit den Prozessdaten für die Darstellung dynamischer Prozesse jedoch, erwies sich als der zentrale Punkt einen neuen Modellierungsansatz zu untersuchen. An dieser Stelle erfolgt in der Regel der größte Aufwand an individueller und wiederkehrender Programmier- und Entwicklungszeit für animierte grafische Objekte. Deshalb konzentrierten sich die Untersuchungen in einem ersten Schritt auf die Analyse und Definition generischer Prozesse und Programmabläufe für eine Modellierung. In einem zweiten Schritt musste ein geeignetes Werkzeug für Interaktion mit dem Modell für die praktische Arbeit zur Verfügung gestellt werden. Folgende Aufgaben wurden durchgeführt:

- a. Variable und Zustandsgrößen aus der Domäne werden über das datenzentrische Beschreibungsmodell abstrahiert und müssen vom Modell zu verarbeiten sein. Eine geeignete Abstraktion war zu erarbeiten.
- b. Analyse generischer Prozess- und Programmabläufe für animiert Objekte
- c. Definition und Abstraktion eines entsprechenden Regelwerks, um die Ergebnisse aus b) zu beschreiben.
- d. Definition und exemplarische Umsetzung des Werkzeugs

Im Ergebnis war der Nachweis zu führen, dass mit dem beschriebenen Arbeitspaket und der Bereitstellung des Modells, wesentliche und zeitraubende Programmieraktivitäten im Entwicklungsprozess animierter Grafikobjekte durch Konfigurieren mithilfe eines geeigneten Werkzeugs ersetzt werden können.

3 Beschreibung der Forschungsarbeiten

3.1 Methodische und funktionale Integration in das datenzentrische Beschreibungsmodell

In dem vorliegenden datenzentrischen Modell wird die Applikation um ein Softwaremodell gestaltet, bei dem die Daten im zentralen Fokus und vor allem außerhalb der Applikation als eigenes Konstrukt stehen. Die Zugriffe auf diese Daten sind so konzipiert, dass sie lokal, über Netzwerke, über verschiedene Rechner mit verschiedenen Betriebssystemen und Programmiersprachen funktionieren. Die Funktionen einer gesamten Steuerung werden in einem Designprozess konzipiert, die Umsetzung in ausführbare Programme auf den verschiedenen Systemkomponenten wird über die Middleware realisiert. Voraussetzung dafür ist natürlich die vorherige Integration der Komponenten (Hard- als auch Software) in das „Middleware-Framework“. Auf diese Weise sind generell Domänen realisierbar, in denen heterogene Rechnerumgebungen, Datenlogger, I/O-Systeme, Sensorik und Aktuatoren transparent miteinander kommunizieren. Eine weitere Eigenschaft dieses Modells sind Variablenstrukturen, die mit Intelligenz ausgestattet sind. Eine zu verarbeitende Messgröße teilt z. B. innerhalb des Datenmodells automatisch Änderungen über ihren Zustand anderen Prozessen mit. Diese Prozesse müssen nicht umständlich „pollend“ Änderungen von Daten abfragen, sondern werden von der Messgröße automatisch informiert und aktiviert. Eine Variable besitzt also neben der reinen Dateninformation die Eigenschaft, Funktionen direkt auslösen zu können. Damit werden die Programmstrukturen wesentlich vereinfacht und übersichtlicher. Die meisten Systemfunktionen sind über einfaches Konfigurieren anstatt durch aufwendiges Programmieren umsetzbar. Mit der Möglichkeit der Simulation der Prozessvariablen und der Aufzeichnung aller Änderungen lassen sich einfach Testprozeduren außerhalb der eigentlichen Steuerungssoftware aufsetzen. Dadurch werden die Entwicklung und der Test entscheidend vereinfacht. Test und Entwicklung greifen so nahtlos ineinander, ohne dass die Steuerungssoftware entsprechend erweitert werden muss.

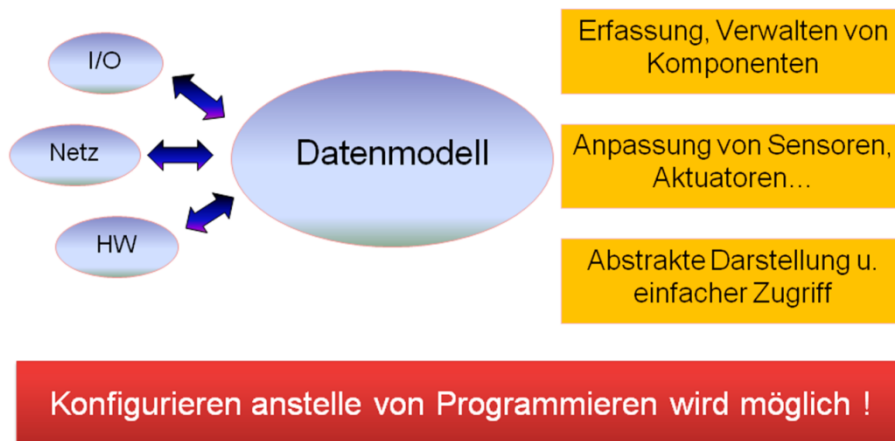
Aus diesem Zusammenhang heraus war es notwendig, ein detailliertes Verständnis über die Abstraktion der Daten im Modell in Form der Variablenstrukturen zu erwerben, um die möglichen und geeigneten Konzepte für die Umsetzung in der embedded Grafik zu erstellen.

In der Abbildung 3 sind die drei zentralen Einheiten dargestellt, deren Variableneigenschaften als zentrale Komponenten für die Modellierung im Grafiksystem zur Verfügung stehen müssen. Das sind:

- Das Erfassen und Verwalten von Komponenten
- Das Anpassen an Informationen, die von Sensoren, Aktuatoren, Reglern, Algorithmen, etc. bereitgestellt oder verarbeitet werden.
- Die Abstraktion für die Darstellung und möglichst einfache Zugriffsmethode

Aus diesen Eigenschaften wird eine Modellierung des Gesamtsystems (Domäne oder Subdomänen) möglich, die es erlaubt, die Anforderungen, wie in 2.1, 2.3 und 2.4 beschrieben zu erfüllen.

Abb. 3: Das Datenmodell, für die Beschreibung der Systemkomponenten



3.2 Umsetzung der Ergebnisse aus 3.1 im Grafiksystem und Einführung eines Objekteditors (siehe 2.5)

Das Datenmodell beschreibt die durch Variablen repräsentierten Prozesszustände einer Domäne.

Diese Variablen sollen möglichst ohne explizite Programmierung mit Hilfe einer geeigneten Grafikplattform visualisiert und manipuliert werden. Die zu lösende Kernproblematik ist das Bereitstellen bzw. Erstellen von beliebigen animierbaren grafischen Anzeigeobjekten, ohne die Erfordernis expliziten Programmierens.

Das Zeichnen von Objekten könnte zurzeit sicherlich mit unterschiedlichen Werkzeugen und der Unterstützung der entsprechenden Grafikformate seitens der Grafikplattform zufriedenstellend gelöst werden. Solche Objekte sind jedoch statisch. D.h. es können keine Regeln vergeben werden, die es ohne Programmieren erlauben, das Aussehen so zu animieren, dass entsprechende Messwerte automatisch angezeigt werden. Sobald anspruchsvolle Oberflächen gefordert werden ist es deshalb in der Praxis häufig notwendig einen Grafikdesigner für die ansprechende optische Gestaltung und einen Programmierer für die notwendige Animation zu beauftragen.

Zielsetzung ist es nun die Aufgabe ausschließlich dem Grafikdesigner zu übertragen. Hierfür müssen jedoch alle Programmieraufgaben so ersetzt werden, dass der Designer in der Lage ist durch Konfigurieren von Regeln dasselbe Ergebnis zu erzielen.

Um diese Problematik zu lösen war es notwendig ein neues Grafikformat zu entwickeln welches neben der rein darstellenden Funktionalität auch die Änderung aller Eigenschaften in Abhängigkeit

von variablen Steuergrößen erlaubt. Dieses neue Grafikformat kann auch als eine „Interpretersprache“ für Vektorgrafikobjekte verstanden werden.

Ein Anzeigeobjekt welches in diesem Format erstellt wird enthält also nicht nur die optische Beschreibung des Objektes sondern auch eine Beschreibung des kompletten dynamischen Verhaltens in Abhängigkeit von variablen Eingangsgrößen. Eingangsgrößen können zu visualisierende Werte aus dem Datenmodell sein. Es ist jedoch auch möglich Ereignisse wie Mausbewegungen, Toucheingaben usw. als Eingangsgrößen zu verwenden. Dies ermöglicht es beliebige Objekte so zu konfigurieren dass sie auch Ereignisse auslösen können.

Für eine komfortable Handhabung des neuen Formats ist es erforderlich ein entsprechendes Zeichen- und Konfigurationswerkzeug anzubieten. In Anlehnung an gängige Zeichenwerkzeuge wurde exemplarisch ein Werkzeug entwickelt, welches es erlaubt Vektorobjekte zu entwerfen und zu animieren. Die Vektorobjekte werden als selbständige Einheiten im Dateisystem hinterlegt. Sie können dort zur Weiterverwendung in Bibliotheken geordnet und mit Hilfe eines GUI-Builders in Bildschirmmasken der entsprechenden Visualisierung eingebunden werden.

Für die Visualisierung der Daten des Datenmodells und die Interaktion mit Bedienern muss das Visualisierungssystem nun folgende Eigenschaften aufweisen:

- Einlesen der Konfigurationsdaten (Bildschirmmasken, Verknüpfungen, ...)
- Andocken an das Datenmodell
- Auslesen der Variableneigenschaften der anzuzeigenden Variablen
- Auslesen der Eigenschaften der Anzeigeobjekte
- Abgleichen der Skalierungen und Aktionen die in den Konfigurationsdaten festgelegt sind
- Abarbeiten von Visualisierungsanforderungen
- Überprüfen und gegebenenfalls Weiterleiten von Benutzerinteraktionen an das Datenmodell
- Veröffentlichen (anzeigen) der zu visualisierenden Daten auf einem
 - Lokalen Grafikdisplay
 - auf einem Remotedisplay
 - auf einem Webbrowser

Für die Umsetzung wurden folgende Module neu erstellt bzw. erweitert:

- Zur Erstellung und Animation der Objekte war eine zentrale Aufgabe einen Objekteditor zu (siehe Abschnitt 2.5) zu entwerfen. Das neue Grafikformat wurde exemplarisch implementiert.
- Der GUI-Builder von XiSys Software wurde erweitert.
- Das Visualisierungsprogramm XiVisu welches zur Laufzeit die oben beschriebenen Belange abarbeitet wurde neu geschrieben. Ein Minimalset an Möglichkeiten wurde implementiert.

- Modul zur Interpretation des neuen Grafikformats wurde in den XiBase9-Grafikserver implementiert.

3.3 Exemplarische Umsetzung der Modellierung am Beispiel eines Avionik-Geräts

1. Schritt: Entwurf eines Grafikobjekts im Designprozess

Die grafischen Objekte, hier ein Fluglageanzeiger (Abb.4), werden mit dem „Objekteditor“ entworfen. Der Objekteditor erzeugt eine Liste aus Befehlen, die das Aussehen des Objektes sowie dessen dynamisches Verhalten beschreiben. Diese Befehle werden vom Grafikserver zur Laufzeit interpretiert.

Das hier beschriebene Objekt soll folgende dynamische Eigenschaften unterstützen:

- Horizontdarstellung mit Steigen, Sinken, Schräglage
- Darstellung und Einblendung von Grenzwerten wie Überziehwinkel, Schräglagen, etc.
- Einblendung von Anflugwinkeln, programmierten Wegemarken, Warnungen.

Das Objekt benötigt hierfür 2 Animationseingänge. Die Animationseingänge werden in einem 2. Schritt via GUI-Builder mit entsprechenden Variablen des Datenmodells verknüpft. Weiterer Aufwand ist nicht notwendig.

Mit dem Objekteditor wird die Vektorliste im neu definierten Format erstellt. [Wie aus Abb. 4 ersichtlich ist](#), besteht das gezeigte Objekt aus Hunderten von Befehlen. Die Befehlsliste wird jedoch vollautomatisch aus den Konfigurationseingaben abgeleitet. Sie dient nur zur Kontrolle und muss nicht direkt manipuliert werden.

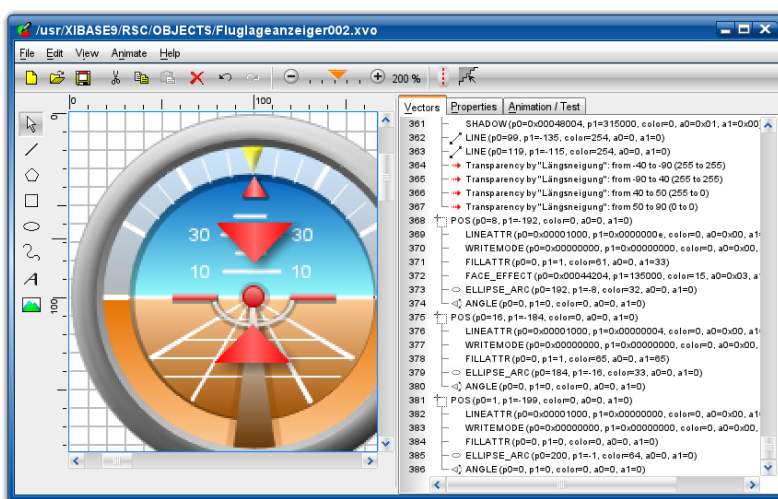


Abb. 4 Objekteditor mit Vektorliste, am Beispiel Fluglageanzeiger

Eine Herausforderung der Arbeiten war es, die Bedienung des Objekteditors [ist intuitiv erlernbar zu gestalten](#). Das Bedienkonzept sollte sich an gängigen Zeichenwerkzeugen orientieren. Für alle

Objekteigenschaften wurden deshalb **einfach zu handhabende Konfigurationsmöglichkeiten vorgesehen**. Darüber hinaus soll und muss jede Zustandsänderung unmittelbar angezeigt werden können.

Im Entwicklungsprozess wird zuerst **wird** das Objekt statisch, d.h. ohne Berücksichtigung seiner dynamischen Eigenschaften entworfen. Im zweiten Schritt werden die Animationseingänge, über welche das Objekt von außen manipuliert werden kann, definiert. Im nächsten Schritt können nun einzelne Eigenschaften von Vektoren mit Animationseingängen verknüpft werden. Der immer verfügbare WYSIWYG-Modus zeigt sofort, ob das gewünschte Verhalten erreicht wird.

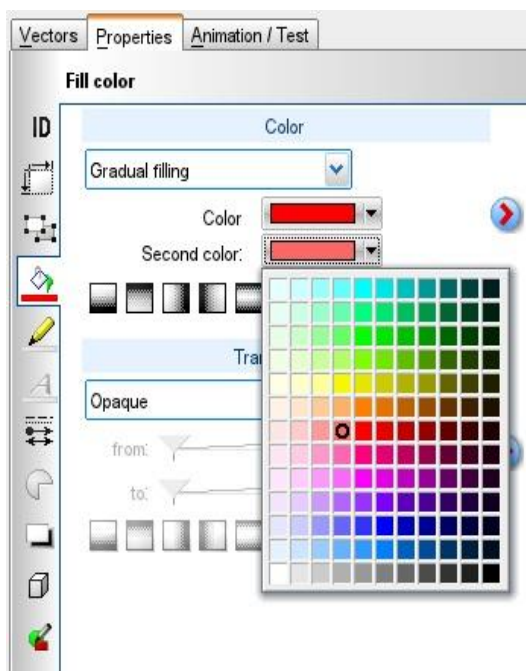


Abb. 5 Statische Eigenschaften einstellen, hier: Farbverlauf

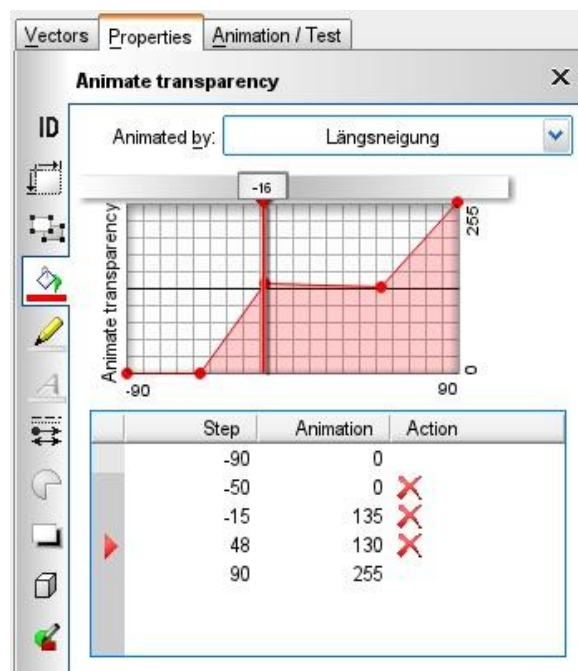


Abb. 6 Dynamische Eigenschaften konfigurieren, hier: Animation der Transparenz

Das Festlegen des dynamischen Verhaltens erfolgt analog zu den statischen Eigenschaften. Es wird ein Animationseingang mit seinem Minimalwert, Maximalwert und seinem aktuellen Wert angelegt. Je nach Eigenschaften des Animationseingangs (softwaregesteuert, ereignisgesteuert, timergesteuert, ...) kann der aktuelle Wert zur Laufzeit mit einem Softwarekommando oder z.B. über eine Toucheingabe geändert werden.

Alle Vektoren die im neuen Grafikformat definiert sind können sich auf Animationseingänge beziehen und in definierten Grenzen ihre Eigenschaften mit Hilfe des Animationswertes ändern.

In unserem Beispiel wird der Vektor der die Neigung des Horizonts darstellt über den Animationseingang „Querneigung“ gesteuert.

Animationen konfigurieren, anstatt zu programmieren

Das Aussehen sowie das dynamische Verhalten des Objektes kann bereits während des Erstellens ausführlich getestet werden. Schon während der Festlegung der einzelnen Animationsschritte wird sofort die Auswirkung auf das dynamische Verhalten des Objektes im Editor angezeigt. Es kann auch in einen generellen Testmodus umgeschaltet werden mit dem ein Objekt mit allen seinen dynamischen Regeln getestet werden kann.

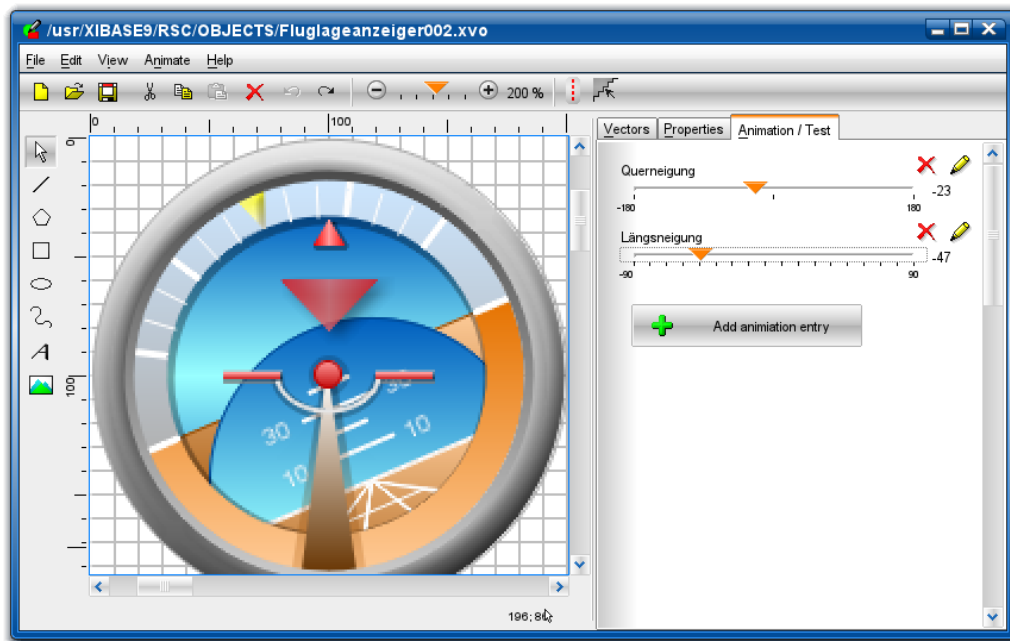


Abb. 7 Test des dynamischen Verhaltens

2. Schritt: Prozessvariable konfigurieren, anstatt zu programmieren

Der erweiterte GUI-Builder wurde mit einem zusätzlichen Reiter auf seiner Eigenschaftenseite ausgestattet. Über diesen sind die neuen Interfaces zu dem Datenmodell konfigurierbar. Der GUI-Bilder wird zum Konfigurationswerkzeug, das die Variablen aus dem Modell mit Objekten aus den Bildschirmmasken verbindet. Mit ihm wird das Regelwerk dem die Visualisierung genügen soll, festgelegt. Eigenschaften wie Datentyp, Grenzwerte, Passwortschutz werden, falls sie im Datenmodell hinterlegt sind, übernommen und mit den Eigenschaften der Anzeigeobjekte automatisch abgeglichen. In diesem Regelwerk sind nicht nur die Verknüpfungen der Variablen beschrieben. Es wird auch das generelle Verhalten der Bildschirmmasken (Fenster öffnen, Fenster schließen, passwortgeschützte Aktionen, Sprachumschaltung, Grenzwertüberprüfungen, Fehlermeldungen, ...) definiert.

Ähnlich wie das Erstellen von Anzeigeobjekten werden Bildschirmmasken generiert. Zuerst werden mit ausgewählten Objekten Bildschirmmasken erstellt. Das Positionieren erfolgt via „Drag & Drop“. Nach dem Festlegen bzw. Ändern der Objekteigenschaften wird mit Hilfe eines Modellbrowsers die Zuordnung der Modellvariablen zu den Anzeigeobjekten getroffen. In einem weiteren Schritt werden die spezifischen Eigenschaften der Variablen mit den gewünschten Anzeigefunktionen abge-

stimmt. Das komplette Regelwerk wird in einem Konfigurationsfile im XML-Format abgelegt. Diese Konfiguration wird später vom Visualisierungsprogramm als Regelwerk zur Laufzeit verwendet. Um einem SPES2020-Metamodell gerecht zu werden, muss in einer zukünftigen Ausbaustufe das XML-Konfigurationsfile so geändert werden, dass es den Anforderungen des neuen Meta-Datenformats entspricht.

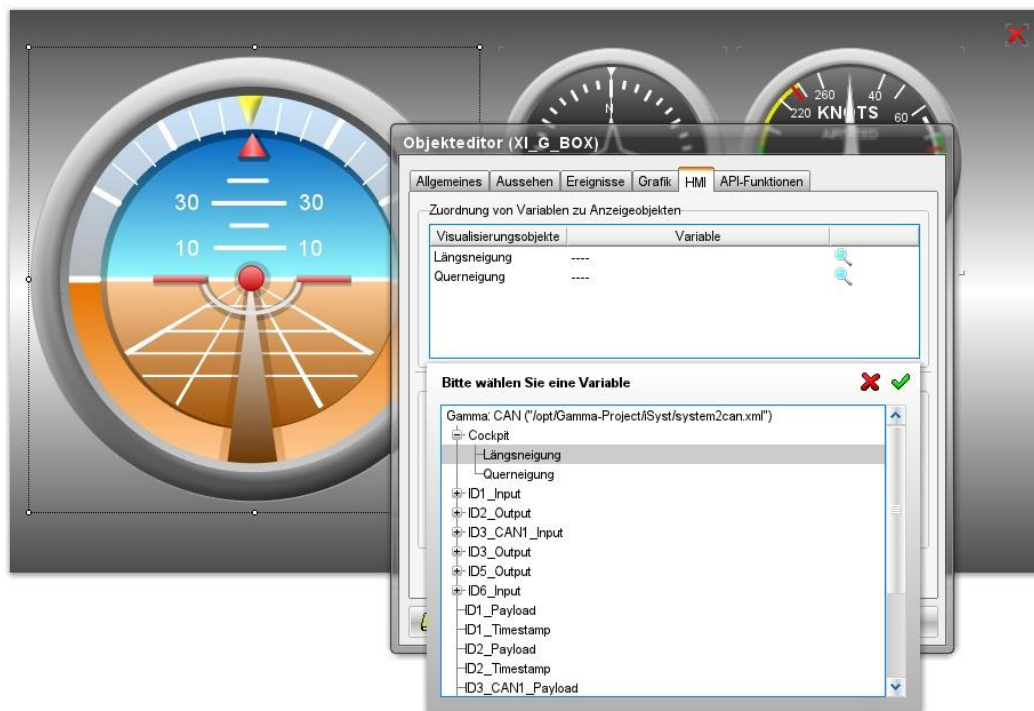


Abb. 8: Verknüpfen des Grafikobjekts Fluglageanzeige mit dem Datenmodell der Domäne im GUI-Builder

3. Schritt: Verknüpfung des konfigurierten Regelwerks mit den Laufzeiteigenschaften der Objekte

Die Grafikobjekte haben nun ein Aussehen, wie Form, Farbe, Größe, Transparenz, Beschriftung und die Beschreibung ihrer Bewegungseigenschaften erhalten. Zur Laufzeit müssen die Objekte nun noch zum Leben erweckt werden. Das Regelwerk hierfür ist in der Konfigurationsdatei die mit dem GUI-Builder erzeugt wurde abgebildet. Die Visualisierung von Daten erfolgt häufig nach einem ähnlichen Schema (immer wiederkehrend).

- Laden und Auswerten der Konfigurationsdatei
- Initialisieren der Grafikumgebung
- Laden und Organisieren der Bildschirmmasken
- Andocken an das Datenmodell
- Initialisieren des Variablenzugriffs
- Abgleich der Eigenschaften (Skalieren, ...)

- Erfassen von Variablenänderungen, z. B. Reaktionen auf Signale oder Polling-Verfahren.
- Organisieren der Änderungen in Prioritätslisten
- Verteilen der Änderungen an betroffene Objekte
- Loggen von Wertepaaren zur Darstellung von Trends
- Manipulieren von Variablen durch GUI – Events, z. B. Reaktionen auf Eingaben von Außen, wie Änderungen von Wegepunkten, Trimmwerten, etc.
- Überprüfen von Grenzwerten
- Passwortschutz
- Verteilte Anzeige, d.h. Ein- und Ausgabefunktionen auf verschiedene Displays.

Diese Funktionalität kann in einem geeigneten Modell abgebildet werden. Es stellt eine weitere Abstraktionsebene zur Verfügung, über die die obigen Abläufe, mit individuellem Regelwerk ausgestattet, automatisch abgearbeitet werden.

In unserem Fall wurde die Funktionalität exemplarisch im Visualisierungsprogramm „XiVisu“ abgebildet. Dieses Programm benötigt nur die entsprechende Konfigurationsdatei. Es kann für alle Arten von Visualisierungen die dem obigen Modell genügen verwendet werden.

4 Zusammenfassung der Ergebnisse

Folgende Ergebnisse wurden im Projekt erreicht:

- Die Integration und Adaption am Beispiel der XiBase9-embedded Grafik an das datenzentrische Beschreibungsmodell wurde exemplarisch umgesetzt, dadurch steht die Funktion dieser Subdomäne der übergeordneten Domäne „High Lift Prüfstand“ zur Verfügung.
- Mit der Modellierung kann wiederkehrende Programmierarbeit bei der Entwicklung komplexer grafischer Oberflächen drastisch reduziert und vereinfacht werden.
- Gestalten von grafischen Objekten ohne Programmierung ist möglich.
- Der GUI-Builder von XiSys Software, als Maskengenerator, wurde um die Möglichkeit des Konfigurierens und Einbeziehens von Variablen aus einem Datenmodell erweitert. Daten werden durch Abfragen übernommen, in Listen verwaltet und mit Beschreibungen ihrer Eigenschaften versehen. Dadurch sind sie mit Anzeigefunktionen und -eigenschaften, Skalierungen und Laufzeitregeln verknüpfbar.
- Anpassungen im zentralen Datenmodell, z. B. an das SPES2020-Metamodell können durch die Abstraktion einfach im Grafikmodell umgesetzt werden.

Diese neue Methode, Grafikobjekte zu gestalten, zu konfigurieren und zu verknüpfen, anstatt aufwendig zu programmieren, hat das Potenzial, 80-90% des Entwicklungsaufwandes für traditionelle embedded Grafiksysteme einzusparen. Damit bestehen bereits zu Projektende die Prognosen für eine erfolgreiche wirtschaftliche Verwertung sehr günstig.

5 Einsehbare Projektdokumente

Die Konzepte, Nachweise und exemplarische Umsetzungen der Forschungsarbeiten wurden hauptsächlich in Form von Softwareprogrammen erstellt, die im Rahmen dieser Zusammenfassung natürlich nicht veröffentlicht werden. Einsicht in die Programme und deren Dokumentationen können sehr gerne in Rücksprache mit der XiSys Software GmbH, RST Industrie Automation GmbH und IMACS GmbH erfolgen.