

Offene Referenzplattform für Echtzeit Linux im Maschinen und Anlagenbau (RTLOpen)

| | | | |
|----------------------|-------------------|-----------------|--------------------|
| R. Kalmar | S. Meininger | K. Viebig | H. Wußmann |
| Fraunhofer IESE | Hofmann Maschinen | VisionTools | BERGHOF Automation |
| Fraunhofer-Platz 1 | Altrheinstr. 11 | Goethestr. 63 | Harretstr. 1 |
| 67663 Kaiserslautern | 67550 Worms | 68751 Waghäusel | 72800 Eningen |

Kurzfassung

In dem Verbundprojekt RTLOpen wird auf Basis des Open Source Betriebssystems Linux eine flexible, performante, zukunftssichere und kostengünstige Referenzplattform für den mittelständischen Maschinen- und Anlagenbau erstellt. Dabei werden die besonderen Anforderungen der Anwendungsdomäne wie Echtzeitanforderungen, komplexe Berechnungen, Interoperabilität, Sicherheit, Qualität, Langlebigkeit und Ausfallsicherheit berücksichtigt. Ergebnisse sind u.a. eine frei verfügbare Referenzarchitektur basierend auf dem RTAI Realtime-Linux Kern, eine Realtime-Entwicklungsmethode, eine Sammlung von verfügbaren Open Source Werkzeugen, sowie Erfahrungsberichte.

1. Echtzeitsysteme im Maschinenbau

1.1 Anforderungen für Echtzeit-Systeme im Maschinenbau

Der deutsche Maschinen- und Anlagenbau hält einen internationalen Wettbewerbsvorsprung insbesondere aufgrund der herausragenden Qualität seiner mechatronischen Lösungen. Die hierfür eingesetzte Technik ist für Steuer- und Regelungsaufgaben optimiert und in diesem Punkt sehr leistungsfähig. Dabei hat der Einzug der Software-Technik zu leistungsfähigeren und wettbewerbsfähigeren Produkten geführt und die Position des deutschen Anlagen- und Maschinenbaus gefestigt. Bereits heute wird in einigen Anwendungsgebieten der größte Teil der Wertschöpfung durch den Einsatz von Informationstechnik erzielt.

Im Vergleich zur PC-Technik bietet die im Anlagen- und Maschinenbau etablierte programmierbare Steuer- und Regelungstechnik jedoch in zunehmendem Maße Nachteile bei der Interoperabilität (z.B. der Intranet-Anbindung der Maschinen), bei modernen Bedienoberflächen und bei der Implementierung von den immer wichtiger werdenden komplexen Auswertungen von Messdaten. Dazu kommt, dass die spezialisierte Hardware bis zum Faktor zehn teurer ist als Lösungen auf Basis von Standard-Industrie-PCs. Vor dem Hintergrund des hohen Wertschöpfungsanteils der Informationstechnik können hier durch den Einsatz von PC-Technologie erhebliche Wettbewerbsvorteile auf der Kostenseite wie auch bei der technischen Leistungsfähigkeit erzielt werden.

Kostengünstigen PC-Lösungen auf der Basis von Desktop-Betriebssystemen, wie sie im Maschinenbau bereits für Bedienoberflächen und zur Datenauswertung verwendet werden, fehlt jedoch die Echtzeitfähigkeit, welche für sicherheits- und zeitkritische Steuer- und Regelungsaufgaben unbedingt gegeben sein muss. Im Bereich der PC-Echtzeitbetriebssysteme sind zur Zeit Linux- und Windows-NT-basierte Lösungen kommerziell verfügbar. Technische wie auch lizenzpolitische Gründe führen jedoch beim Einsatz kommerzieller Lösungen, wie z.B. von Windows-NT im Maschinen- und Anlagenbau zu Problemen und hohen Kosten. Beispielsweise machen bei größeren Stückzahlen und insbesondere kleineren eingebetteten Produkten die Lizenzkosten einen nicht unerheblichen Teil der Gesamtkosten aus. So kann der Einsatz von freier Software auch bei einem angenommenen höheren Entwicklungsaufwand schnell den eigenen Wertschöpfungsanteil steigern und zu steigenden Gewinnen führen.

Desktop-PC-Betriebssysteme unterliegen einem teuren, hochfrequenten Update-Zyklus und werden nach ca. drei bis fünf Jahren nicht mehr vom Hersteller unterstützt. Eine häufig wechselnde Lizenzpolitik der kommerziellen Hersteller macht eine längerfristige betriebswirtschaftliche Kalkulation deshalb unmöglich. Diese Situation verbietet den Einsatz kommerzieller Desktop-PC-Betriebssysteme in langlebigen Maschinen und Anlagen. Die häufigen und nicht beeinflussbaren Änderungen kommerzieller Betriebssystemanbieter oder Abkündigungen führen zu hohen Wartungskosten der Software. Open Source Software und eine Linux basierte Plattform versprechen hier mehr Sicherheit und geringe Kosten. Allerdings fehlt bei Open Source eine regulierende Instanz, die Interoperabilität und Stabilität garantiert. Hier versucht die Industrie derzeit mit dem Open Source Automation Development Lab eine unabhängige Zertifizierungsinstanz zu etablieren [OSADL].

1.2 Stand der Praxis von Open Source für Echtzeitsysteme im Maschinenbau

Das Standard-Linux Betriebssystem ist nicht echtzeitfähig. Verschiedene kommerzielle und nichtkommerzielle Erweiterungen ermöglichen es jedoch auch Echtzeitprozesse unter Linux laufen zu lassen. Sie basieren meist auf einer der folgenden beiden Möglichkeiten Echtzeitfähigkeit in das originär nicht echtzeitfähige Linux einzubringen: Verringerung der Latenzzeiten durch Minimierung der unterbrechungsfreien Funktionen im Betriebssystemkern (geringere Reaktionszeiten z.B. auf Interrupts, jedoch keine Zusicherung harter Echtzeit) oder Dual-Kernel-Architekturen mit neuen Programmierschnittstellen (Reaktionszeit im Mikrosekundenbereich, garantierte Zeitschranken, „harte“ Echtzeit). Im Open Source-Bereich konkurrieren derzeit die sogenannten „preemption-Patches“ (z.B. [Molnar]) mit Echtzeitkernen (z.B. [RTAI]). RTLOpen setzt hier auf die Erweiterung RTAI, die harte Echtzeit ermöglicht.

Neben der reinen Verfügbarkeit einer Betriebssystemplattform ist das Vorhandensein entsprechender Software-Werkzeuge, sowie einer Reihe von standardisierten und frei verfügbaren Treibern für industrieübliche Sensorik, Aktorik sowie für Kommunikationsaufgaben unabdingbar für einen breiten industriellen Einsatz. Gerade hier bestehen noch erhebliche Defizite im Open Source Bereich, die im Rahmen des Projekts RTLOpen untersucht werden sollen.

Insgesamt ist um Open Source im Bereich Echtzeitanwendungen ein rasch wachsender Markt entstanden, welcher Linux in kurzer Zeit zu einer wichtigen Plattform hat werden lassen. Für Echtzeitlösungen gilt dies allerdings noch nicht in dem Maße.

1.3 Projektziele

Vor dem Hintergrund der in der Einleitung dargestellten Zusammenhänge erscheint eine auf Linux basierte Open Source-Initiative strategisch sinnvoll und wirtschaftlich notwendig, um die Position der deutschen Maschinen- und Anlagenbauer zu untermauern und langfristig zu sichern. Im Rahmen des Projekts RTLOpen wird deshalb die professionelle Verwendung einer Open Source-Lösung für mechatronische Aufgaben evaluiert und pilotiert.

Die wissenschaftlichen Ziele des Projekts RTLOpen lassen sich wie folgt zusammenfassen:

- Identifikation, Bereitstellung, Evaluierung, Dokumentation und Verfügbarmachen einer industriell nutzbaren Open Source Softwareplattform
- Definition und Evaluierung von Techniken und Methoden zur Software-Entwicklung von Echtzeit-Applikationen

Die Herausforderungen des Projekts lagen und liegen in der Kombination und Ergänzung existierender freier Open Source Linux-Erweiterungen und Software-Werkzeuge, sowie der Ergänzung von Entwicklungsmethoden um echtzeitspezifische Aspekte die geeignet sind für die Anwendung im Maschinenbau. Des Weiteren fehlten empirische Aussagen und Erfahrungsberichte zum Einsatz von Open Source-Lösungen in der Branche.

2. Die RTLOpen Plattform

Die RTLOpen Plattform bezeichnet die Summe der Projektergebnisse und umfasst alle konzeptionellen Aspekte, die für eine Entwicklung von Echtzeitsystemen im Maschinenbau notwendig sind: Entwicklungsmethodik, Betriebssystem und Entwicklungswerkzeuge.

2.1 Entwicklungsmethodik

Eines der wesentlichen Ziele des Projekts RTLOpen ist die systematische Anwendung software-technischer Methoden zur kontrollierten Entwicklung echtzeitfähiger Software Systeme im Maschinenbau. Hierbei betrachten wir eine Symbiose aus bereits bewährten Praktiken aus Theorie und Praxis der Domänen Software, Hardware und Elektronik. Bewährte Konzepte zur Beherrschung komplexer Systeme basieren unter anderem auf Prinzipien wie Generalisierung, Teile-und-Herrsche sowie Abstraktion. Unter Berücksichtigung dieser Konzepte entsteht im RTLOpen Projekt eine Methode, die bislang bekannte komponenten-basierte Ansätze [KobrA] zur systematischen Erstellung großer Softwaresysteme erweitert und auf die Bedürfnisse im Maschinenbau angepasst ist. Die Herausforderung im Kontext des RTLOpen Projekts besteht nun darin, geeignete Konzepte zur Modellierung von Echtzeitkonzepten zu identifizieren und unter gleichzeitiger Berücksichtigung der spezifischen Rahmenbedingungen in eine Methodik zu integrieren, welche an der Systementwicklung im Maschinenbau ausgerichtet ist. Ausgangspunkt für die Modellierung der Software unter Echtzeitanforderungen ist die UML (Unified Modeling Language, www.uml.org) die sich in der Softwareindustrie bereits

seit Jahren als inoffizieller Standard durchgesetzt hat. Da klassische Entwicklungsmethoden im Maschinenbau meist keinen Bezug zu systematischer Integration von Software nehmen, ist auch hier die Herausforderung, die UML als Modellierungssprache auf die Konzepte zu reduzieren die im Maschinenbau sinnvoll angewendet werden können.

Eines der Ergebnisse des RTLOpen Projekts ist z.B. die integrierte Modellierung und der Test von Echtzeitanforderungen im Einklang mit der UML. Die Erweiterung des Spezifikationsmodelle um Timingdiagramme wie sie in der UML 2.0 definiert sind sowie die Nutzung der UML eigenen Erweiterungsmechanismen erlauben es, jedes Komponentenartefakt mit Echtzeitanforderungen zu modellieren bzw. zu annotieren. Die Konzepte von Spezifikation und Realisierung unterstützen zum einen die parallele Entwicklung der benötigten Software Bausteine, lassen sich jedoch ebenso auf Systemebene übertragen. Dies bedeutet vor allem, dass die parallele Entwicklung von Software, Hardware und Elektronik bezüglich der zu Beginn gestellten Echtzeitanforderungen synchronisiert und effizient durchgeführt werden kann (s. Abbildung 1).

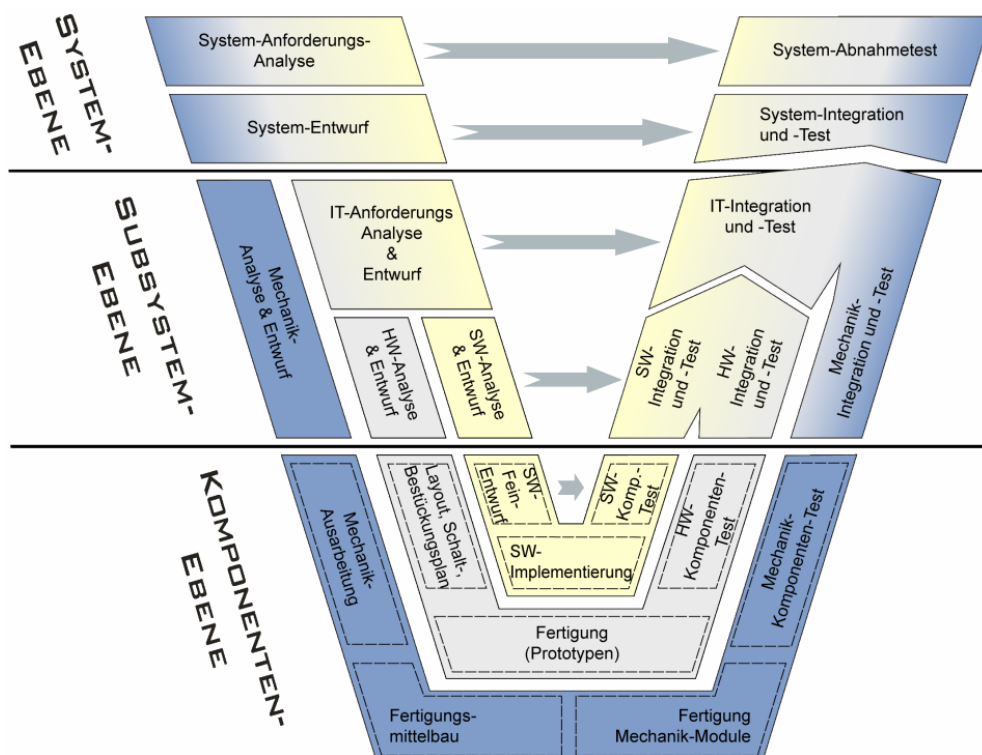


Abbildung 1: Das 3-Ebenen Vorgehensmodell; parallele Entwicklungsstränge in mechatronischen Systemen [EQUAL]

Der komponenten-basierte Ansatz liefert nicht nur Vorteile auf der Konstruktionsseite von Systemen, er unterstützt zugleich Qualitätssicherungsmaßnahmen sowie effiziente Wiederverwendung von bereits entwickelten Komponenten oder gar kompletten Systemen.

Eine solch systematische Vorgehensweise in der Software-Entwicklung ist für viele kleinere Maschinenbauunternehmen neu. Im Rahmen des RTLOpen-Projektes wurde z.B. die Modellierung mit UML von der Fa. Hofmann Maschinen evaluiert.

Software Qualitätssicherung

Testen gehört zu den fundamentalen Prinzipien im Entwicklungsprozess und wird von fast allen Firmen angewandt, wenn auch mit unterschiedlichem Grad der Systematisierung von bekannten Verfahren. Im Bereich des Testens von Echtzeitanforderungen kann auch forschungsseitig ein erhebliches Defizit festgestellt werden, welches sich zusätzlich in einer geringen Werkzeugunterstützung widerspiegelt. Ein Verfahren, angefangen bei der Spezifikation bis hin zum konkreten Testen von Echtzeitanforderungen steht im Mittelpunkt von Testansätzen, die im RTLOpen Projekt entwickelt werden.

Ausgangspunkte für die im Projekt angewendete Testmethodik ist eine Spezifikation des Systems aus welcher Testfälle abgeleitet wurden. Dabei gab es die zuvor beschriebenen funktionalen und zeitlichen Anforderungen.

Das folgende Beispiel greift die verfolgte Methodik im Modultest auf und erklärt wie diese am Projekt-Demonstrator angewendet wurde. Eine zuvor durchgeführte Werkzeugrecherche ergab, dass kein Open Source Werkzeug die Anforderungen für das Testen von Echtzeitsystemen abdeckte. Somit wurde zunächst das traditionelle Modultest-Werkzeug cppUnit eingesetzt, mit dem Echtzeitanforderungen der Demonstratorsoftware überprüft wurden. Ziel ist zu klären, ob und wie exakt Überprüfungen von Echtzeitanforderungen mittels eines solchen Standard-Werkzeugs möglich sind.

Hauptbestandteile des RTL Open Demonstrators sind Aktuatoren, mit dem ein Benutzer das System steuern kann, ein Sensor, der Positionsveränderungen des Spielbrettes durch konkrete Werte vorgibt und ein Motor, der das Spielbrett entsprechend der Werte des Sensors bewegt. Abbildung 2 zeigt schematisch die Echtzeit-Anforderung. Ein Benutzer steuert das System mit Hilfe der Aktuatoren. Dies wird von einem Sensor registriert und führt zu Korrekturbefehlen an den Motor. Diese Information wird mit einer gewissen Verzögerung durch den Motor gesteuert. Diese Verzögerung darf einen definierten Wert nicht übersteigen.

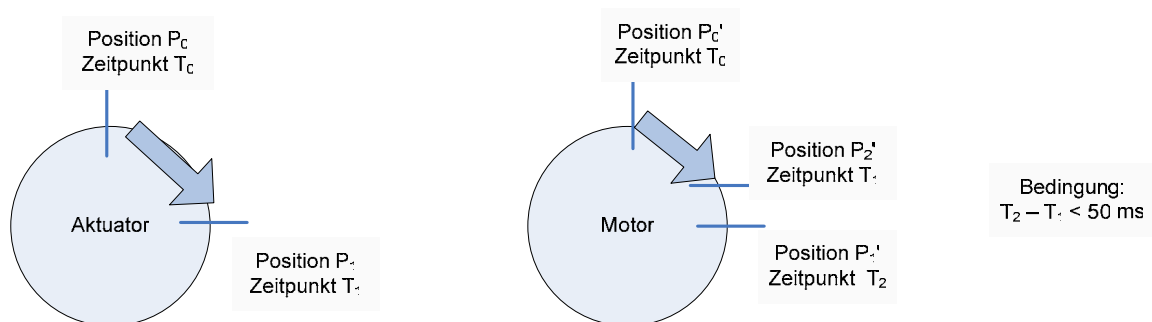


Abbildung 2: Echtzeitanforderung des Projekt-Demonstrators

Diese Anforderung wird nun auf zwei unterschiedliche Arten überprüft: Werkzeug unterstützt (semi-automatisch) und manuell.

1. CppUnit wird in Verbindung mit einem Zeitmesser, der durch das System zur Verfügung gestellt wird, ausgeführt. Wie genau diese Messung ist, wird dann mit dem zweiten Ansatz verglichen.
2. Es wird ein Oszilloskop verwendet, das die einzelnen Positionen und die benötigte Zeit sehr genau messen kann.

Es wird erwartet, dass der zweite Ansatz genauere Daten liefert als das Testwerkzeug, wobei offen ist, wie groß die Unterschiede sind und ob die Ergebnisse des Testwerkzeuges für den aktuellen Kontext ausreichend sind.

Zusammenfassend liefern diese Ergebnisse eine Antwort, wie Testen mit Echtzeitanforderungen möglich ist und an welchen Stellen die Grenzen traditioneller Open Source Testwerkzeuge liegen.

2.2 Betriebssystem-Plattform

Die RTAI-Linux Echtzeit-Plattform [RTAI] mit den verwendeten Patches ist hinreichend stabil und leistungsfähig, um kommerziellen Anforderungen zu genügen. Allerdings erfordert die Kombination der verschiedenen Module Geduld und Ausdauer, da keine Instanz über die Synchronisation aller Teile achtet – es gibt derzeit noch keinen Distributor für RTAI-Linux, ähnlich SuSE oder Redhat. Im Rahmen des Projektes wurden verfügbare Teile genutzt und integriert und keine Teile selbst entwickelt (außer speziellen Treibern). Eine RTLOpen-Zusammenstellung der Pakete ist über die Projekt-Webseite als CD-ROM Abbild verfügbar.

2.3 Open Source Software-Werkzeuge

Bei der Software-Werkzeugunterstützung ergibt sich die größte Lücke zwischen freien und kommerziellen Anbietern. Deshalb ist hier die Ergebnisfindung auch noch nicht abgeschlossen. Gegebenenfalls ist es effektiver und effizienter, für bestimmte Entwicklungstätigkeiten auf kostengünstige und ausgereifte Lösungen aus der Windows-Welt zurückzugreifen und Linux lediglich als Zielsystem zu verwenden. Eine Empfehlung seitens des Projekts für Entwicklungswerkzeuge ist in einem eigenen Handbuch dokumentiert.

3. Erfahrungen und Bewertungen

3.1 Modellierung von Realzeit-Systemen

Ähnlich einer Konstruktionszeichnung im Maschinenbau können Software-Systeme geplant und entworfen werden. Neben den funktionalen Anforderungen kommen im Falle von Echtzeitsystemen Reaktionszeiten und Ausführungszeiten als nichtfunktionale Anforderungen hinzu. Im Rahmen des RTLOpen-Projekts wurde bei der Firma Hofmann Maschinen aus Worms Methoden der komponentenorientierten Softwareentwicklung evaluiert. Zum Einsatz kam in diesem Zusammenhang die grafische Beschreibungssprache UML in der Version 2.0. Die System- und Softwarestruktur des fertig entwickelten und ohne UML2 dokumentierten Echtzeit-Messkerns einer Reifengleichförmigkeitsmaschine (RGM) wurde mit UML2 neu dokumentiert.

Diese neue Dokumentation des Messkerns hatte zum Ziel, die Struktur eines Systems mit Echtzeit-Anforderungen so zu dokumentieren, dass auch nicht an der Entwicklung beteiligte Personen an Hand der Dokumentation zu einem Verständnis des Systemaufbaus gelangen.

Auf diese Weise wurde mit wenigen Elementen von UML2 (Verteilungs- und Komponentendiagramm, Sequenzdiagramm und Aktivitätsdiagramm) ein komplexes Echtzeit-Messsystem hinsichtlich

1. der statischen Sicht der einzelnen Komponenten des Systems, und
2. der zeitlichen Abläufe und Abhängigkeiten der Prozesse in dem System

UML - konform dokumentiert.

Als Werkzeug wurde das Open Source-Tool Poseidon [Poseidon] verwendet. Es reicht für das Sammeln von ersten Erfahrungen aus und bietet eine gute Unterstützung der verschiedenen Diagrammart.

Es ist noch schwierig zu sagen, ob die Erwartungen erfüllt wurden, da die Ingenieure die UML-Notationen kaum kennen. Die Möglichkeit, aus einer Modellspezifikation für zukünftige Szenarien Programmcode zu generieren wird als vielversprechend angesehen.

3.2 Werkzeugkette für eingebettete Systeme unter Linux

Im Rahmen der Entscheidung für eine Hard-/Software-Plattform muss man auch die zugehörige Toolkette hinsichtlich Effizienz, Kosten und Akzeptanz untersuchen und in die Bewertung für die Gesamtlösung einbeziehen. Aus Sicht eines Steuerungsherstellers, der im wesentlichen vom Absatz seiner Hardwarekomponenten lebt, ist es in der Akquisitionsphase immer schwierig, teure Software-Entwicklungs-Tools mitverkaufen zu müssen, obwohl hierin keine eigene Wertschöpfung liegt.

Deshalb ist der Ansatz, eine Toolkette aus dem Open-Source-Bereich zusammen zu stellen, aus vertrieblicher Sicht zu begrüßen. Gepaart mit dem Wegfall der Kosten für Betriebssystem-Lizenzen kann sich je nach Stückzahl eine Einsparung von 100.000 EUR und mehr pro Jahr ergeben.

Somit bleibt die Frage nach der Effizienz und der Akzeptanz. Geht man als Neuling an die Thematik Linux und Open-Source-Tools heran, steht man nach einer ersten Sichtungphase vor einer schier unüberschaubaren Vielfalt von Informationen, Tools und Möglichkeiten. Hier ist es nicht einfach heraus zu finden, welche Tools in welcher Kombination die eigenen Anforderungen optimal erfüllen. Diese Frage lässt sich auch nicht allgemeingültig beantworten, weil je nach Aufgabenstellung andere Tools benötigt werden. Anforderungen eines reinen Anwendungsprogrammierers sind andere als die eines hardwarenahen Systementwicklers.

Deshalb soll im Rahmen des RTLOpen-Projekts eine beispielhafte Toolkette gezeigt werden, die in Kombination gut funktioniert. BERGHOF hat bei der Auswahl der Toolkette den Fokus auf eingebettete Systeme gerichtet. Dabei ist es wichtig, die Anwendungsszenarien im Auge zu behalten. Die Toolkette muss einerseits in der Büroumgebung einer Entwicklungsabteilung in Kombination mit der Windows-dominierten Officewelt funktionieren, andererseits aber auch vor Ort an Maschinen und Anlagen im Feld einsetzbar sein. Deshalb ist neben den reinen Tools auch die Hard- und Software-Infrastruktur mit zu berücksichtigen.

Entlang der im Entwicklungsprozess notwendigen Aufgaben wurden Tools gesichtet, erprobt und zu einer funktionierenden Kette zusammengefügt. Diese Zusammenstellung darf nicht als verbindliche, sondern nur als eine mögliche betrachtet werden. Sie soll dem Neueinsteiger helfen, schnell zu einer funktionierenden Umgebung zu kommen, aus der er selbst seine individuelle Werkzeugkette zusammenstellen und weiter optimieren kann. Der ansonsten nötige Einarbeitungsaufwand mit Sichtung, Erprobung und Auswahl soll entfallen.

Für die einzelnen Phasen des Entwicklungsprozesses wurden folgende Tools ausgewählt:

- **Spezifikation / Design:**
Kommerzielle Produkte: Microsoft Word, Microsoft Visio, Mindmanager
Open-Source-Produkte: Open Office, Freemind

Anmerkung: Im Rahmen der Spezifikation wird eng mit dem Kunden kommuniziert, der oft nur über die Microsoft-Office-Produkte verfügt. Deshalb bietet sich hier noch an, ohnehin vorhandene kommerzielle Produkte zu verwenden.

- **Entwicklung:**
Open-Source-Produkte:
Integrierte Entwicklungsumgebung IDE Kdevelop mit Editor
Compiler GCC für PowerPC
- **Fehlersuche:**
Debugger GDB für PowerPC
HW-Debugger Lauterbach (HW-bezogen)
- **Versionsverwaltung:**
Subversion
- **Dokumentation:**
Doxygen
- **Verbesserung / Fehlertracking:**
Mantis

Zusammenfassend lässt sich sagen, dass der durch Linux getriebene Open-Source-Gedanke Produkte hervorgebracht hat, die sich im industriellen Umfeld einsetzen lassen. Es gibt Hürden bei der Einführung, die zu bewältigen sind. Neben der Auswahl der richtigen Tools aus der großen verfügbaren Menge macht ihre Erstinstallation mit Zuteilung der richtigen Nutzerrechte oft Probleme. Danach laufen die meisten etablierten Tools stabil und treffen auf große Akzeptanz bei den Entwicklern. Durch die kostenfreie Verfügbarkeit ist sogar die Bereitschaft höher, auch für ungeliebtere Tätigkeiten wie z.B. das Dokumentieren und Versionieren Tools einzusetzen und dann auch konsequent zu benutzen. Eine Charakterisierung und Bewertung aller im Projekt evaluierten Werkzeuge wird in einem Handbuch veröffentlicht.

3.3 Von Windows zu Linux: Migration und Lernkurve

Bei VisionTools wurde bisher sämtliche Software unter Windows ohne Echtzeitanforderungen entwickelt. Für neue leistungsfähige Produkte und eine stabilere Plattform soll langfristig Echtzeit-Linux etabliert werden. Im Zuge der Evaluierung zur Migration von Windows zu Echtzeit-Linux wurden viele Erfahrungen gesammelt, die nachfolgend kurz umrissen werden. Die Integration existierender Werkzeuge und Patches hat sich für Vision Tools als komplexe Problematik herausgestellt, insbesondere die Auswahl des geeigneten Kernels: Aufgrund der verschiedenen Distributionen, sowie Konzepten bei echtzeitfähigen Betriebssystemen wird es zu zusätzlichem Aufwand zur Anpassung von Hard- und Software-Komponenten kommen. Die Verfügbarkeit von Treibern für spezielle und insbesondere moderne Hardware ist nach Erfahrung von Vision Tools nur für Kernel 2.6.x gegeben wie beispielsweise FireWire Treiber.

Freie Software-Werkzeuge, die die besonderen Anforderungen zur Entwicklung eingebetteter Software unter Realzeit-Bedingungen unterstützt, sind vorhanden, und sollen mit Unterstützung durch andere Projektpartner zur Verfügung gestellt werden.

Literatur oder Schulungen sind wenig verfügbar, so dass ein – gegenüber verbreiteten kommerziellen Lösungen – erhöhter Einarbeitungs- und Trainingsaufwand entsteht.

Für die Zukunft wäre es vorteilhaft, einen Distributor und Ansprechpartner für ein Echtzeit-Linux zu haben, so dass die Software-Plattform nicht selbst gepflegt werden muss. Erste Kontakte mit Herrn Knopper (Knoppix) haben Interesse zur Übernahme einer Live-Distribution erkennen lassen.

4. Die zukünftige Rolle von Echtzeit-Linux im Maschinenbau

Aufgrund der Reife der verfügbaren Lösungen und einer stetig wachsenden Anwenderschar von Open Source Software, gerade bei kleineren Unternehmen, wird die Verwendung von Echtzeit-Linux in den nächsten Jahren voraussichtlich deutlich steigen. Ein Indikator dafür sind die Interessensgruppen, die sich in den letzten Jahren gebildet haben ([OSADL, ASQF]), sowie die Sichtbarkeit des Themas auf Messen und Veranstaltungen. Allerdings stellt die Entwicklung von Echtzeit-Applikationen mit Ihren besonderen Anforderungen eine Nische dar, so dass der Markt relativ klein ist.

Die Resonanz auf das Thema „Einsatz von Linux“ und „Open Source“ beim Verband des deutschen Maschinen und Anlagenbaus (VDMA) ist ebenfalls sehr gut. Eine gemeinsame Veranstaltung ist bereits geplant.

Die Anwendungspartner haben grundsätzlich positive Erfahrungen bei der Verwendung von Echtzeit Linux gemacht und Kunden äußerten sich sehr positiv gegenüber ersten Produktprototypen. Sie werden in Zukunft die RTLOpen Plattform auch bei der Entwicklung neuer Produkte einsetzen und vertreiben.

Für Anwender bedeutet die Verwendung der RTLOpen-Plattform, eine unter Industriebedingungen getestete Referenzlösung zu bekommen, die unterschiedlichste Facetten des Einsatzes von Linux-basierter Rechnertechnik samt Software und zugehöriger Werkzeuge in Automatisierungssystemen abdeckt. Die Projektergebnisse sind auf der Webseite www.rtlopen.de frei verfügbar. Aus diesen Gründen erwarten die Konsortialpartner eine gute Resonanz bei den Maschinen- und Anlagenbauern in Deutschland, die durch den Einsatz der RTLOpen-Plattform Stückkosten senken, den Innovationsgrad ihrer Produkte erhöhen und somit letztlich konkurrenzfähiger am Markt agieren können.

5. Referenzen

- [ASQF] Arbeitskreis Software-Qualität und -Fortbildung e. V. mit Fachgruppen Automatisierung und Open Source / Linux.
- [EQUAL] BMBF-Projekt EQUAL, *Methoden zur Unterstützung der entwicklungsbegleitenden Qualitätssicherung von eingebetteter Software*, Förderkennzeichen 02PP1060, <http://www.embedded-quality.de>
- [KobrA] Atkinson, et. al., *Component-Based Product Line Engineering with UML*, Addison-Wesley, 2001.
- [Molnar] Realtime Preemption Patch: <http://people.redhat.com/mingo/realtime-preempt/>
- [OSADL] Open Source Automation Development Lab, <http://www.osadl.org>
- [Poseidon] Poseidon for UML 4.2, <http://www.gentleware.com>
- [RTAI] Realtime Linux Application Interface, RTAI Community Projekt-Homepage: <http://www.aero.polimi.it/~rtai/>