

Universelle Integration und Konfiguration von Geräten mit ASAM GDI

von Dr.-Ing. Robert Patzke

*Dies ist das Manuskript einer Veröffentlichung in der **Elektronik automotive** (November 2006, WEKA-Verlag, www.elektroniknet.de), ergänzt um Anmerkungen zu möglichen Alleinstellungsmerkmalen des ASAM GDI.*

Warum sind viele Softwareprojekte unerwartet teuer, warum erscheint es oft unmöglich, eine Applikation zu erstellen, die allen Erwartungen gerecht wird oder mit wenig Aufwand an geänderte Anforderungen angepasst werden kann?

Der Autor sieht eine Ursache in der mangelhaften Modularisierung von Softwareprojekten und der ständigen Neuerfindung des Rades. Hierbei ist die Integration von Gerätschaften ein Endlosthema. Immer wieder werden neue Treiber zur Einbindung von Geräten geschrieben, sei es durch den Wechsel des Betriebssystems oder Änderungen der Geräteschnittstellen bedingt. Die Kosten für die Geräteintegration sind immens. Der Ruf nach einer universell verwendbaren Standardlösung liegt auf der Hand und der ASAM (Association for Standardisation of Automation and Measuring systems, www.asam.net) hat solche Standards geschaffen. Dafür wurden viele Mannjahre an Diskussionen und Spezifikationen aufgewendet und die Erwartungshaltungen sind groß, die gewünschten Vorteile nutzen zu können.

Aber Systemlieferanten, die diese Aufgaben aus ihrer Sicht bereits zufriedenstellend gelöst haben, werden sich gegen Vorgaben durch einen Standard sperren. Schließlich wird ihnen genau damit die Neuerfindung des Rades abverlangt. Noch schlimmer, der den Standard einfordernde Kunde erwartet natürlich, dass die Geräteintegration nach diesem Standard nicht zu seinen Lasten geht. Er brummt dem Zulieferer zusätzliche Lasten auf, ohne dafür zu bezahlen.

Wie können wir diesen gordischen Knoten zerschlagen, wie erreichen wir eine Win-Win-Situation für Hersteller und Anwender gleichermaßen?

Das funktioniert nur, wenn mit der Einführung des Standards auch ein zuverlässiges Geschäftsmodell für die Zulieferer entsteht. Hier liegt der Anspruch beim Anwender, nicht heute so und morgen anders zu argumentieren, sondern seine Forderung nach ASAM-Standards so deutlich zu artikulieren, dass die Zulieferer darauf ihre Planungen abstützen können.

Für den Standard ASAM GDI haben sich diesbezüglich General Motors und Volkswagen in den letzten Monaten besonders engagiert (Siehe Präsentationen zum ASAM Users Day 2006 auf www.asam.net).

Das Integrationskonzept des ASAM GDI

GDI steht für Generic Device Interface und erhebt damit den Anspruch, als Grundlage für die Integration aller Arten von Geräten zu dienen. Wobei beim GDI unter Geräteintegration nicht nur der Gerätezugang über eine Kommunikationsschnittstelle des Computers verstanden wird, sondern alle Aufgaben, die erforderlich sind, um ein Gerät aus der Sicht des Anwendungsprogrammes betriebsbereit zu machen. Dabei spielt die Konfiguration, also das Aktivieren und Parametrieren der Geräte für eine Applikation, eine herausragende Rolle. Es gibt Anwendungen, z.B. der Einsatz von Datenloggern bei Langzeitmessungen oder im Crashtest, bei denen der Konfigurationsvorgang im Vergleich zur eigentlichen Messwerterfassung und -übertragung eine überragende Rolle spielt.

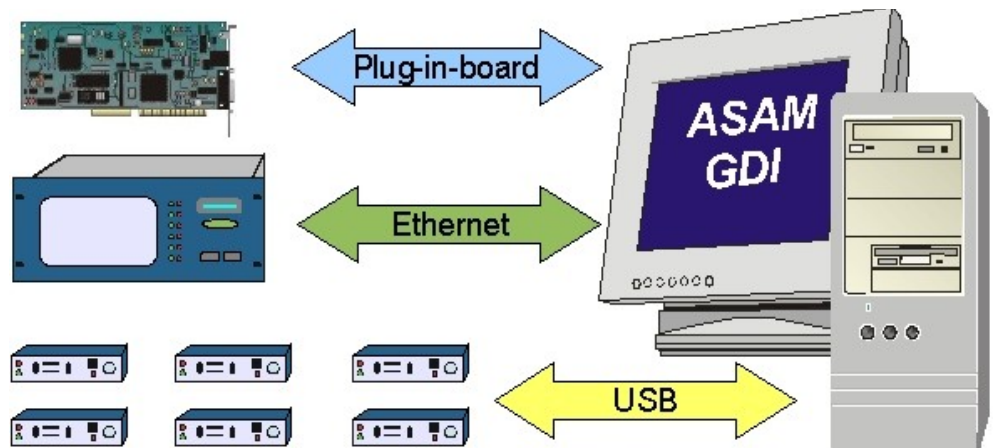


Bild 1. Geräteintegration mit ASAM GDI

Beim ASAM GDI hat man sich bez. der universellen Einsetzbarkeit sehr hohe Ziele gesteckt. Dieser Standard soll

- unabhängig machen vom Betriebssystem des Computers,
- unabhängig machen von der eingesetzten Kommunikationstechnologie,
- unabhängig machen vom Gerätlieferanten,
- die Möglichkeit bieten, Gerätetreiber und angeschlossene Geräte und ihr Verhalten für eine gegebene Computer-Plattform zu zertifizieren und
- unabhängig machen von der technologischen Weiterentwicklung der Hard- und Software in der Zukunft.

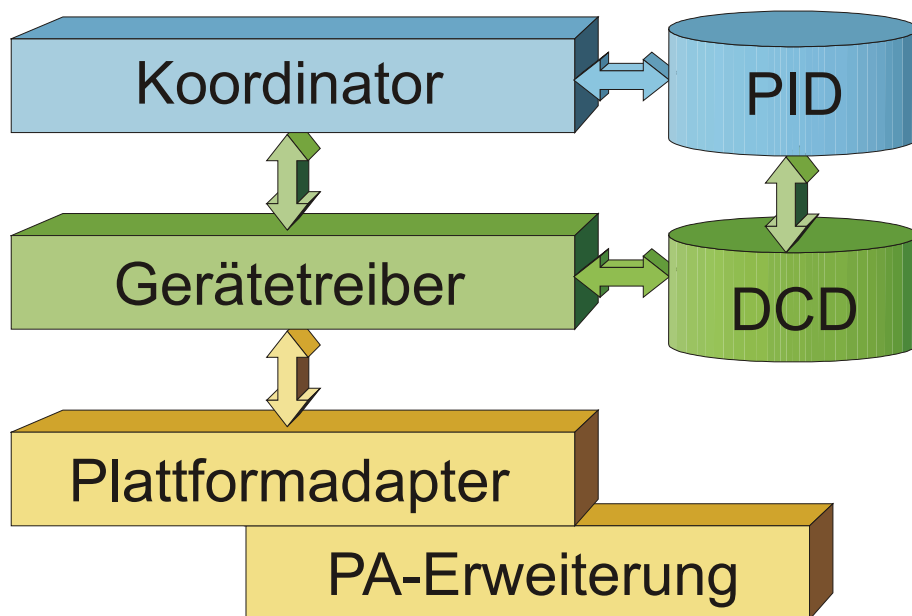


Bild 2: Komponenten des ASAM GDI Standards

Dabei wird auch noch die Migration von Gerätebeständen hochgehalten, denn der GDI

- kapselt ein Gerät und seine Kommunikationsschnittstelle, um es zu anderen Geräten dieser Art für eine Anwendung kompatibel zu machen, und
- erfordert somit keine Geräteneuentwicklung oder die Anwendung spezieller Kommunikationsschnittstellen.

Um all diesen Ansprüchen genügen zu können, definiert der GDI-Standard mehrere Komponenten, die aufeinander aufbauen, aber für sich abgeschlossen spezifiziert und einsetzbar sind. So entstehen entkoppelte Module, die von unterschiedlichen Spezialisten des jeweiligen Anwendungsbereiches bearbeitet werden können (Bild 2).

Plattformadapter und PA-Erweiterung

Der Plattformadapter und die Plattformadapter-Erweiterung sind die Basismodule, um vom Betriebssystem und den damit verwalteten Kommunikationsschnittstellen unabhängig zu werden. Der Plattformadapter ist für jedes unterschiedliche Betriebssystem und die Erweiterung für bestimmte Typen von Kommunikationsschnittstellen zu entwickeln. Zur Zeit stehen am Markt Plattformadapter für LINUX und Windows zur Verfügung und PA-Erweiterungen für RS232, Ethernet-TCP/IP, USB und CAN.

Hier wird der Vorteil der Spezialisierung sofort deutlich, denn es ist naheliegend, einen Plattformadapter von einem Experten für das betreffende Betriebssystem ausführen zu lassen.

Betrachten wir dazu einmal die möglicherweise anstehende Anpassung an die neue Windows-Version Vista und denken daran, dass damit auch wieder neue Treiberkonzepte zu berücksichtigen wären. Die Portierung einer ASAM GDI Umgebung von einer älteren Windows-Version auf die neue würde sich hier auf die Anpassung des Plattformadapters bzw. der PA-Extension beschränken und könnte vom Vista-Spezialisten vorgenommen werden. Alle anderen Komponenten des GDI Standards bleiben davon unberührt. Ähnliches gilt natürlich auch, wenn ein ganz anderes Betriebssystem, z.B. für Echtzeitanwendungen in abgesetzten Systemen, zum Einsatz kommt. Auch hier wird man den Spezialisten für dieses Betriebssystem mit der Anpassung des Plattformadapters betrauen und erreicht damit eine hohe Qualität in der Softwareentwicklung.

Deutlich wird hieran auch die Kostenersparnis durch die Modularität des GDI Standards, denn die Umstellung von einem Betriebssystem auf das andere beschränkt sich entwicklungsstechnisch auf die Anpassung des Plattformadapters.

Plattformadaptoeren sind universell einsetzbare Software-Produkte und für ein Betriebssystem ausgelegt, die Anzahl verschiedener Arten ist überschaubar. Nicht jedoch die Anzahl unterschiedlicher Lieferanten, denn die Festlegung der Schnittstelle zum Gerätetreiber ermöglicht es jedem Betriebssystemspezialisten, ein solches Produkt zu entwickeln und anzubieten. Die PA-Erweiterungen sind für ein bestimmtes Betriebs- und Kommunikationssystem ausgelegt. Sie werden bei Bedarf vom Plattformadapter geladen. In der Regel enthält der Gerätetreiber die entsprechenden Informationen für sein Gerät und stösst das Laden der Erweiterung bei der Initialisierung an. Man beachte, dass auch das zum Konfigurationsvorgang gehört, der weiter unten noch ausführlich erläutert wird.

Anmrk. Dieser einfache Weg, sich vom Betriebssystem und den Kommunikationsschnittstellen unabhängig zu machen, ist ein Alleinstellungsmerkmal des GDI verglichen mit anderen Standards in diesem Bereich.

Gerätetreiber und Gerätefähigkeitsbeschreibung (DCD)

Der Gerätetreiber und die Gerätefähigkeitsbeschreibung (Device Capability Description, DCD) werden vom Gerätehersteller erstellt. Auch hier sorgt die Spezialisierung wieder für entsprechende Qualität, denn das größte Know-how über Eigenschaften und Fähigkeiten eines Gerätes ist beim Hersteller zu erwarten. Der Programmierer des Gerätetreibers muss lediglich die nach oben zum Koordinator vorgegebene Schnittstelle einhalten und kann für die Kommunikation mit seinen Geräten auf die Dienste eines Plattformadapters und der passenden PA-Erweiterung zugreifen.

Wie das Gerät dann aus Sicht der Anwendung zu bedienen ist, welche Fähigkeiten es anbietet und wie diese genutzt werden, wird in einer DCD nach ebenfalls standardisiertem Format hinterlegt. Hier werden sog. virtuelle Geräte (Virtual Device, VD) verwaltet.

Es ist allerdings zu beachten, dass in der Anwendung dieses Konzeptes zwei verschiedene Sichten auf Geräte existieren können. Einmal die Sicht aus der Applikation auf ein optimal geeignetes, d.h. zugeschnittenes, Gerät und die Sicht auf das Gerät selbst, das der Gerätehersteller möglicherweise mit einer anderen Vorstellung der Applikation oder ggf. sogar für einen ganz anderen Anwendungsfall konzipiert hat. Der GDI Standard soll hier auch die Anpassung ermöglichen, also über die Konfiguration nur eine bestimmte Sicht auf ein Gerät zu aktivieren oder diese durch die Kombination verschiedener Geräte darzustellen. Die Zuordnung von virtuellen Geräten zu den physikalischen Geräten ist deshalb von weiteren Randbedingungen abhängig (Bild 3).

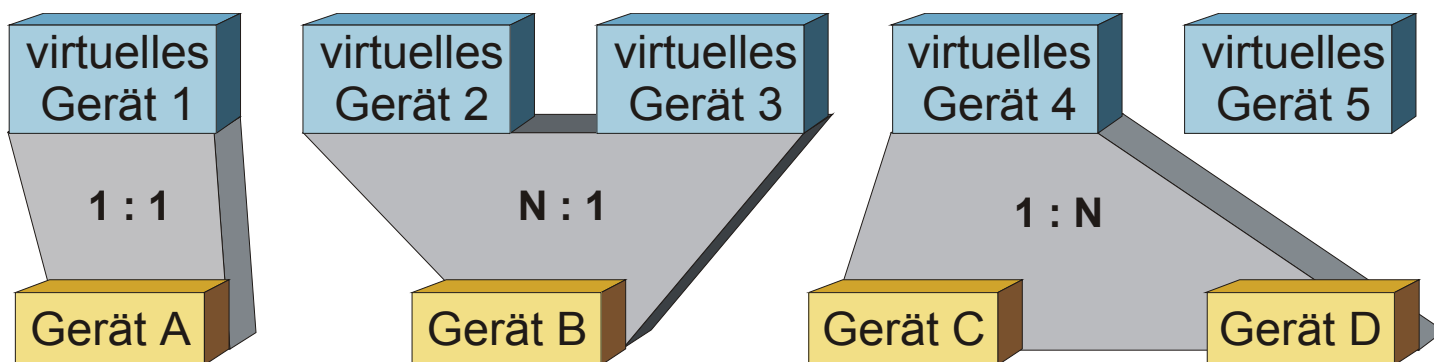


Bild 3: Abbildung von virtuellen und realen Geräten

Der einfachste Fall ist die 1:1-Zuordnung von virtuellem zu physikalischem Gerät. Hierbei ist anzumerken, dass die Gerätebeschreibungen in der DCD Klassenbeschreibungen sind, das heißt, es wird immer nur ein Gerätetyp dargestellt, unabhängig davon, wie viele Geräte dieser Art tatsächlich angeschlossen sind.

Mit der N:1-Zuordnung bietet sich die Möglichkeit, ein Gerät mit mehreren Facetten anzubieten oder eine ganze Gerätegruppe über eine Kommunikationsschnittstelle zu verwalten. Dieser Fall tritt auch dann auf, wenn ein Gerät mehrere entkoppelte Funktionsgruppen aufweist, die unabhängig voneinander verschiedene Zustände annehmen können.

Die 1:N-Zuordnung schließlich erlaubt es, anspruchsvolle Geräteanforderungen durch eine Gruppe sich ergänzender Geräte zu erfüllen. Hierdurch ergibt sich für den Geräteanbieter eine sehr hohe Flexibilität, um die Anforderungen einer Anwendung zu erfüllen. Betrachten wir dazu als einfaches Beispiel eine Leistungsmessung. Während Hersteller A hier seine Leistungsmessgeräte vorteilhaft ins Spiel bringt, kann Hersteller B dieselbe Aufgabe auch durch getrennte Strom- und Spannungsmessgeräte mit Software-Multiplikation im Gerätetreiber erfüllen. Das Anwendungsprogramm stellt hier keinen Unterschied fest.

Ganz ohne Zuordnung zu einem physikalischen Gerät kann über den GDI-Treiber ein sog. Intelligenzausgleich vorstatten gehen, indem Softwarefunktionalitäten, die das Gerät eines Herstellers hat, von einem anderen Hersteller im Gerätetreiber realisiert sind.

Anmrk. Diese verschiedenen Relationsmöglichkeiten bei der Abbildung von logischen zu physikalischen Geräten sind ein weiteres Alleinstellungsmerkmal des GDI Standards verglichen mit anderen Standards für diesen Einsatzbereich. Damit sind hochflexible Gerätetreiber konstruierbar.

GDI Companion Standards

Companion Standards sind beim ASAM GDI vordefinierte virtuelle Geräte aus der Sicht einer bestimmten Anwendung. Hier wird eine komplette DCD definiert und damit liegt ein Pflichtenheft für eine Standard-Gerätesicht vor. Hersteller können dafür Gerätetreiber schreiben. Die Gerätefähigkeitsbeschreibung kann vom ASAM-Server heruntergeladen werden, sobald ein Companion Standard vom Technical Advisory Board (TAB) des ASAM freigegeben wurde. Verfügbar sind dort zur Zeit Companion Standards für

- Crash-Versuche,
- Rollenprüfstände und
- Vielfach-Messtellengeräte.

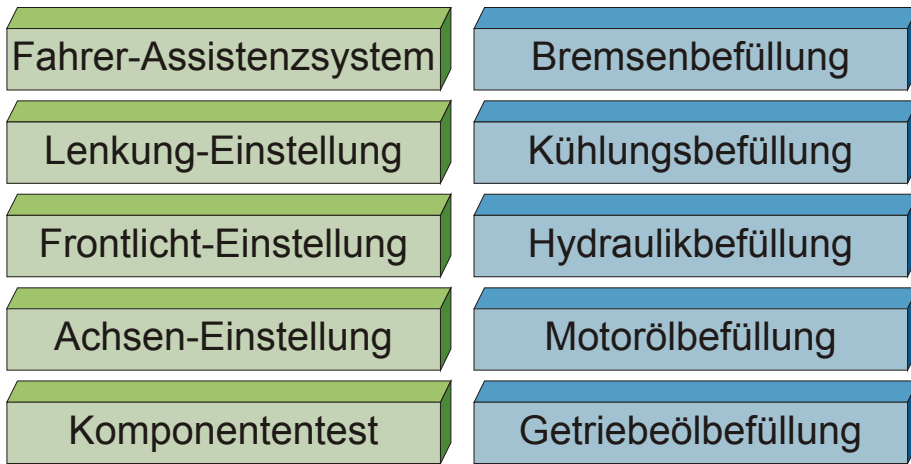


Bild 4: Von VW entwickelte und geplante Companion Standards

Daneben werden aber auch von Anwendern Companion Standards definiert, die nicht oder noch nicht über den ASAM laufen und schnellstmöglich einen konkreten Standardisierungsbedarf decken. Ein Beispiel dafür sind die umfangreichen Aktivitäten bei Volkswagen, wo der ASAM GDI inzwischen auch in der Produktion eingesetzt wird (Bild 4).

In Arbeit ist aktuell eine universelle Ankopplung an die im ASAM AE standardisierte Schnittstelle MCD3 zur Anbindung von KFZ-Steuergeräten (ECU). An dieser Stelle werden Synergieeffekte der ASAM Standards deutlich, denn eine inhaltliche Festlegung zu den virtuellen Geräten ist beim GDI nicht mehr erforderlich. Das Objektmodell des MCD3 wird so, wie es im ASAM AE spezifiziert wurde, in ein virtuelles Gerät vom GDI gekapselt. Dieser Companion Standard beschreibt also lediglich eine bestimmte Methode, die sich neutral zu weiteren möglichen zukünftigen Entwicklungen beim MCD3-Standard verhält.

Daneben werden aber auch von Anwendern Companion Standards definiert, die nicht oder noch nicht über den ASAM laufen und schnellstmöglich einen konkreten Standardisierungsbedarf decken. Ein Beispiel dafür sind die umfangreichen Aktivitäten bei Volkswagen, wo der ASAM GDI inzwischen auch in der Produktion eingesetzt wird (Bild 4).

Anmrk. Die ausgewiesene Möglichkeit, virtuelle Geräte aus Anwendersicht zu definieren und die Treiber daran anzupassen, ist ein weiteres Alleinstellungsmerkmal des GDI Standards verglichen mit anderen Standards für diesen Bereich. Damit kann der Auftrag zur Geräteintegration an den Gerätehersteller gehen, der auch entsprechend hohes Know-how dafür hat und damit zur Verbesserung der Anlagenqualität beiträgt.

Die konkrete Implementierung des ASAM GDI an der Fertigungslinie hat Besonderheiten aufgezeigt, die sicher auch an anderer Stelle der Automobilproduktion noch in Erscheinung treten werden. Es wird deutlich, dass die fortschreitende Entwicklung der Steuergeräte (ECU) und die Aufrüstung der Sensorik in den Kraftfahrzeugen zu einer Rückkopplung zum Produktionsprozess führt und damit höhere Anforderungen an die Flexibilität der Produktionsautomatisierung stellt. Das wurde bei VW in einer GDI-Pilotimplementierung für Befüllprozesse besonders deutlich. Nach Meinung des Autors wird diese Neugestaltung von Prozessen mit Rückkopplung über die OBD-Schnittstelle (On-Board-Diagnose) zu flexibleren Automatisierungssystemen und einer Reduktion der Produktionszeiten führen. Wenn sich jetzt noch weitere Spareffekte durch modularisierte Gerätetreiber ergeben, hat sich für VW der Einsatz des Standards schnell bezahlt gemacht.

Koordinator und Konfigurationsdatei (PID)

Der Koordinator stellt die Schnittstelle zur eigentlichen Anwendung dar und soll diese von systematischen und allgemein formulierbaren Aufgaben entlasten. Dazu gehört die Entkopplung von der Gerätesicht und die Konfiguration der Anlage.

Man kann die aktive Funktion des Koordinators zusammen mit der Konfigurationsdatei (Parameter Instance Description, PID) in etwa mit dem Hochfahr-Vorgang eines modernen Computer-Betriebssystems vergleichen. Die benötigten Treiber werden geladen und die angeschlossenen Geräte werden im Sinne der geplanten Anwendung initialisiert. Die eigentliche Anwendung wird erst dann aktiv, wenn alle von ihr benötigten Ressourcen bereitgestellt und parametrisiert sind.

Mit diesem Vorgehen gelingt ein weiterer entscheidender Schritt der Modularisierung, denn bislang haben Anwendungsprogramme immer beide Aufgaben zu erfüllen gehabt, nämlich das Inbetriebnehmen der Gerätschaften und die eigentliche Nutzung. Naheliegender, dass auf dem Weg über den Konfigurator auch noch Diagnose und Wartung der Ressourcen in eigene Anwendungsprogramme ausgelagert werden können.

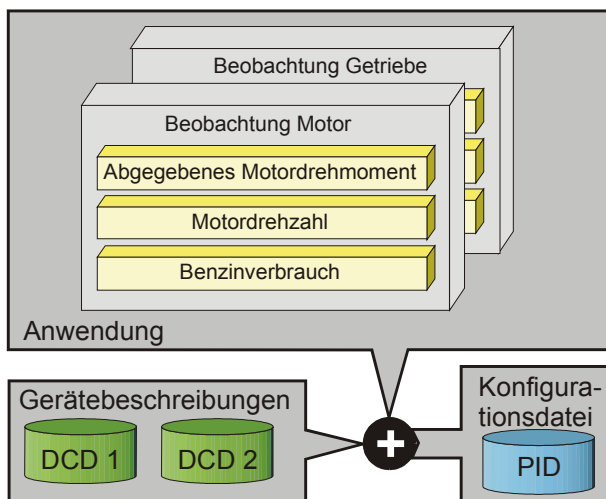


Bild 5: Konfiguration über Gerätebeschreibungen und Anwendung

Anmrk. Das Konzept, die Konfiguration einer Anlage an den Koordinator zu übergeben und die eigentliche Anwendung von solchen Aufgaben zu entlasten, ist ein weiteres Alleinstellungsmerkmal des GDI Standards verglichen mit anderen Standards für diesen Bereich. Damit ist ein entscheidender weiterer Modularisierungsvorgang ermöglicht.

Wie erfährt aber der Koordinator, welche Ressourcen eine Anwendung benötigt und wie diese zu parametrieren sind? Hier wird die Konfigurationsdatei (PID) wirksam, denn zu ihrer Erstellung werden neben den DCDs auch noch Eingaben durch den Projektierer der Anwendung herangezogen (Bild 5).

Offline-Konfiguration

Das Konzept des GDI, für die Konfiguration ein Dateiformat zu definieren und den Konfigurator als Interpreter dieser Datei den Hochlauf einer Anlage zu ermöglichen, legt die Offline-Konfiguration nahe. Also eine eigenständige Applikation, mit der eine Anwendung projiziert und alle dafür notwendigen Ressourcen beschrieben werden.

Um hier den Anschluss an die immer mehr in den Vordergrund tretenden WEB-Technologien zu erreichen, wurde für die Konfigurationsdatei XML als Format gewählt. Als Vorlage für die Konfigurationsdatei dienen aufbereitete Gerätefähigkeitsbeschreibungen im Format XML Schema. Dieses Format wurde gewählt, um ähnliche Beschreibungsmöglichkeiten zu bekommen, wie sie das ursprüngliche DCD-Format des ASAM GDI mit einer an CORBA-IDL angelehnten Syntax hat. Zwar wäre das grundsätzlich auch mit einem eigenen XML-Dialekt machbar gewesen, dennoch verspricht der Einsatz von XML-Schema den Vorteil, Standard-XML-Werkzeuge für die Erstellung bzw. Bedienung der Konfigurationsdatei heranzuziehen (siehe Kasten). Für den Einsatz von XML Schema zur Gerätebeschreibung spricht auch noch das grundsätzliche Klassenmodell des GDI, bei dem in der DCD die Klassen virtueller Geräte beschrieben werden und deren Instanziierung über die Kommunikation mit dem Gerätetreiber erfolgt. Die Konfigurationsdatei enthält entsprechend alle diese Instanzen. Ein ähnliches Bild wird auch allgemein beim Einsatz von XML und XML Schema gezeichnet. XML Schema wird als Klassenbeschreibung geführt und die damit erstellten XML Dateien sind die Instanzen.

Gerätebeschreibung: Warum XML Schema und nicht einfach XML?

Die Anwendung von UML zur grafischen Darstellung programmtechnischer Zusammenhänge und von XML zur Beschreibung inhaltlich gefüllter Strukturen kann man heute schon als Stand der Technik betrachten.

Aus der Anwendung von XML entstehen allerdings häufig auch diverse "Dialekte", indem die Namen der XML-Elemente zur Interpretation der Dateninhalte herangezogen werden (z.B. Kennzeichnung von Fettschrift in Texten). Dies hat den Nachteil, dass es spezieller Parser/Interpreter bedarf, um den Inhalt dieser Elemente zu analysieren oder zu modifizieren.

XML Schema bringt dagegen von Haus aus mächtige Anweisungen zur Festlegung (Interpretation) von Dateninhalten mit sich. Es existieren normative Angaben zur Definition von Datentypen, Wertebereichen, Defaultvorgaben, Festwertvorgaben und vieles mehr. Entsprechend ausführlich können die Eigenschaften von Geräten dargestellt werden, sofern sie sich über numerische Werte beschreiben lassen. Und gerade das ist bei technischen Gerätschaften der Fall. Dazu folgendes einfache Beispiel.

```
<element name="Input" maxOccurs="8">
  <complexType>
    <sequence>
      <element name="Bandwidth">
        <simpleType>
          <restriction base="integer">
            <enumeration value="1"/>
            <enumeration value="10"/>
            <enumeration value="100"/>
            <enumeration value="1000"/>
          </restriction>
        </simpleType>
      </element>
    </sequence>
    <attribute name="id" type="ID"/>
  </complexType>
</element>
```

Es soll ein Messgerät zum Einsatz kommen, das mehrere Signaleingänge mit unterschiedlicher, einstellbarer Bandbreite besitzt. Für die Anwendung werden zwei Kanäle genutzt und für verschiedene Grenzfrequenzen konfiguriert. Die Parametrierung erfolgt "Offline" anhand der Gerätebeschreibungen und es wird eine Konfigurationsdatei erzeugt, die beim Start der Anlage ausgewertet wird.

Zunächst sind die Eigenschaften des Gerätes in XML Schema zu beschreiben. Dazu wählen wir für einen Messeingang ein Element mit dem Namen <Input> und darin ein weiteres mit dem Namen <Bandwidth> zur Angabe der Einstellungen für die Grenzfrequenz (weisser Kasten). Mit der Angabe maxOccurs="8" erfährt der Anwender, dass dieses Gerät maximal 8 Kanäle dieser Art bietet. In der Typspezifikation von <Bandwidth> werden für diesen Kanaltyp die Einstellungsmöglichkeiten 1, 10, 100 und 1000 vorgegeben.

Bilden wir daraus nun die konkrete Konfigurationsdatei in XML, die die Instanzen der benötigten Kanäle mit ihren zugewiesenen Eigenschaften darstellt (gelber Kasten, rechts).

Man kann sehr leicht erkennen, dass diese Instanzen mit Standard-XML-Werkzeugen erzeugt und bearbeitet werden können, weil dazu keine Auswertung der Elementnamen erforderlich ist. Die relevanten Informationen sind in der standardisierten "Sprache" XML-Schema formuliert.

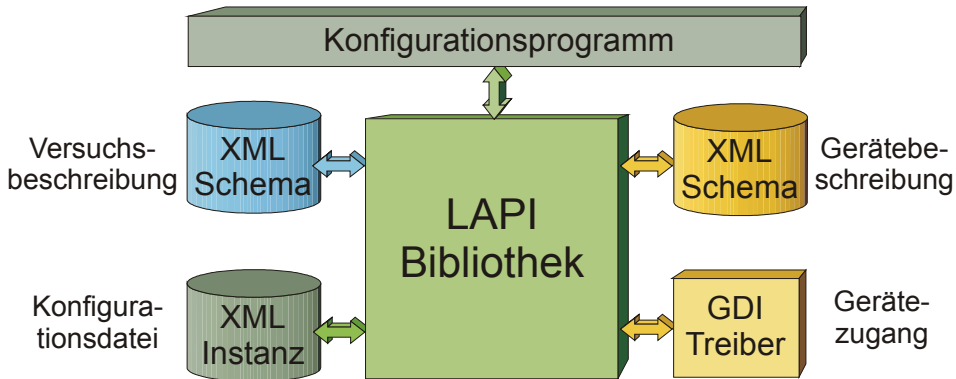
```
<Input id="Eingang01">
  <Bandwidth>10</Bandwidth>
</Input>
<Input id="Eingang02">
  <Bandwidth>100</Bandwidth>
</Input>
```

Damit diese XML-Instanz auch eindeutig von den Gerätetreibern verstanden werden kann, bedarf es einer vordefinierten generischen Gerätestruktur. Und genau das leistet der GDI mit seiner Zuordnung von Gerätefähigkeitsbeschreibung und Treiberschnittstelle.

Anmrk. Der Einsatz von XML Schema anstelle von XML zur Beschreibung der Gerätefähigkeiten für die Konfiguration ist ein weiteres Alleinstellungsmerkmal des GDI Standards verglichen mit anderen Standards für diesen Bereich. Damit wird der Einsatz von Standard XML-Werkzeugen zur Konstruktion und Bedienung von Gerätebeschreibungen und Konfigurationsdateien begünstigt.

Projekt Datenlogger-Konfiguration

In einem gemeinschaftlich von BMW, DaimlerChrysler und Porsche geleiteten Projekt zur herstellerunabhängigen Konfiguration von Datenloggern für den Bereich Gesamtfahrzeug-Engineering und -Erprobung wird von dem oben beschriebenen Konzept Gebrauch gemacht. Unter dem Begriff LAPI (Logger-API) wird eine Software-Bibliothek entwickelt, die Zugriffe auf die Gerätebeschreibungsdateien (Datenlogger und Messmodule in XML Schema formuliert) und auf die Konfigurationsdateien (XML Instanz) ermöglicht und damit als Basis für anspruchsvolle Konfigurationsapplikationen dient.



Die Gerätebeschreibungsdateien und die zugehörigen GDI-Treiber werden von den Geräteherstellern entwickelt. Die Software-Bibliothek und ein XML Schema zur Darstellung des gesamten Versuchs werden von der Fa. **MFP** entwickelt. In diesem Projekt wird kein Koordinator eingesetzt, da es hier ausschließlich um die Konfiguration von Datenloggern und Messmodulen geht und das Parsen/Interpretieren der Konfigurationsdatei mit entsprechender Bedienung der Gerätetreiber von der LAPI-Bibliothek unterstützt wird.

Bild 6: Komponenten des Logger-API-Projektes

In diesem Projekt sind noch weitere Besonderheiten realisiert, die zur Zeit (noch) nicht Gegenstand einer Standardisierung in der GDI-Arbeitsgruppe sind:

- Suchen nach kompatibler Hardware in einem Überangebot verschiedener Hersteller,
- Rückwirkung willkürlich ausgewählter Hardware auf die Möglichkeiten zur Konfiguration,
- Zuschalten von Herstellerapplikationen zur Detailbearbeitung und
- Interaktion mit Gerätetreibern zur Kapselung von Daten in besonderen Formaten (z.B. A2L-Dateien).

Die LAPI-Arbeitsgruppe hatte zudem entschieden, Messmodule mit XCP-Protokoll einzufordern. Hierfür wurde von **MFP** auch eine PA-Extension erstellt, die das XCP-Protokoll über den CAN-Bus abwickelt.

ASAM GDI in der ISO

Schon vor einigen Jahren wurde ein NWIP (New Work Item Proposal) zur Normung des ASAM GDI in der ISO gestellt. Um ein solches Vorhaben zu starten, müssen sich mindestens fünf Länder bereit erklären, daran mitzuarbeiten. Beim GDI haben sich sogar sieben Länder dazu bereit erklärt, was das zuständige technische Komitee der ISO dazu bewegen hat, für dieses Vorhaben eine eigene Arbeitsgruppe einzurichten (Bild 7, nächste Seite).

Die entsprechende ISO 20242 ist auf 6 Teile ausgelegt:

1. Overview,
2. RMSI, Resource Management Service Interface (Plattformadapter),
3. VDSI, Virtual Device Service Interface (Gerätetreiber),
4. DCPT, Device Capability Profile Template (DCD und PID),
5. APSI, Application Program Service Interface (Koordinator), und
6. Conformance test methods, criteria, and reports.

Companion Standards oder spezifische PA-Extensionen sind nicht Gegenstand der ISO-Normung.

ISO 20242-1 wurde bereits 2005 veröffentlicht. In 2007 wird mit Teil 2, 3 und 4 gerechnet. Möglicherweise sind Teil 5 und Teil 6 dann erst 2008 soweit, obwohl auch für 2007 auf dem Plan.

Anmrk. Die vollständige Festlegung eines Konzeptes zur Geräteintegration und -konfiguration in einer ISO Norm mit mehreren Teilen ist ein weiteres Alleinstellungsmerkmal des GDI Standards verglichen mit anderen Standards für diesen Bereich. Damit ist ASAM GDI eine "runde Sache" und stellt sich den Anforderungen an Stabilität und Zuverlässigkeit für eine strategisch wichtige Basis.

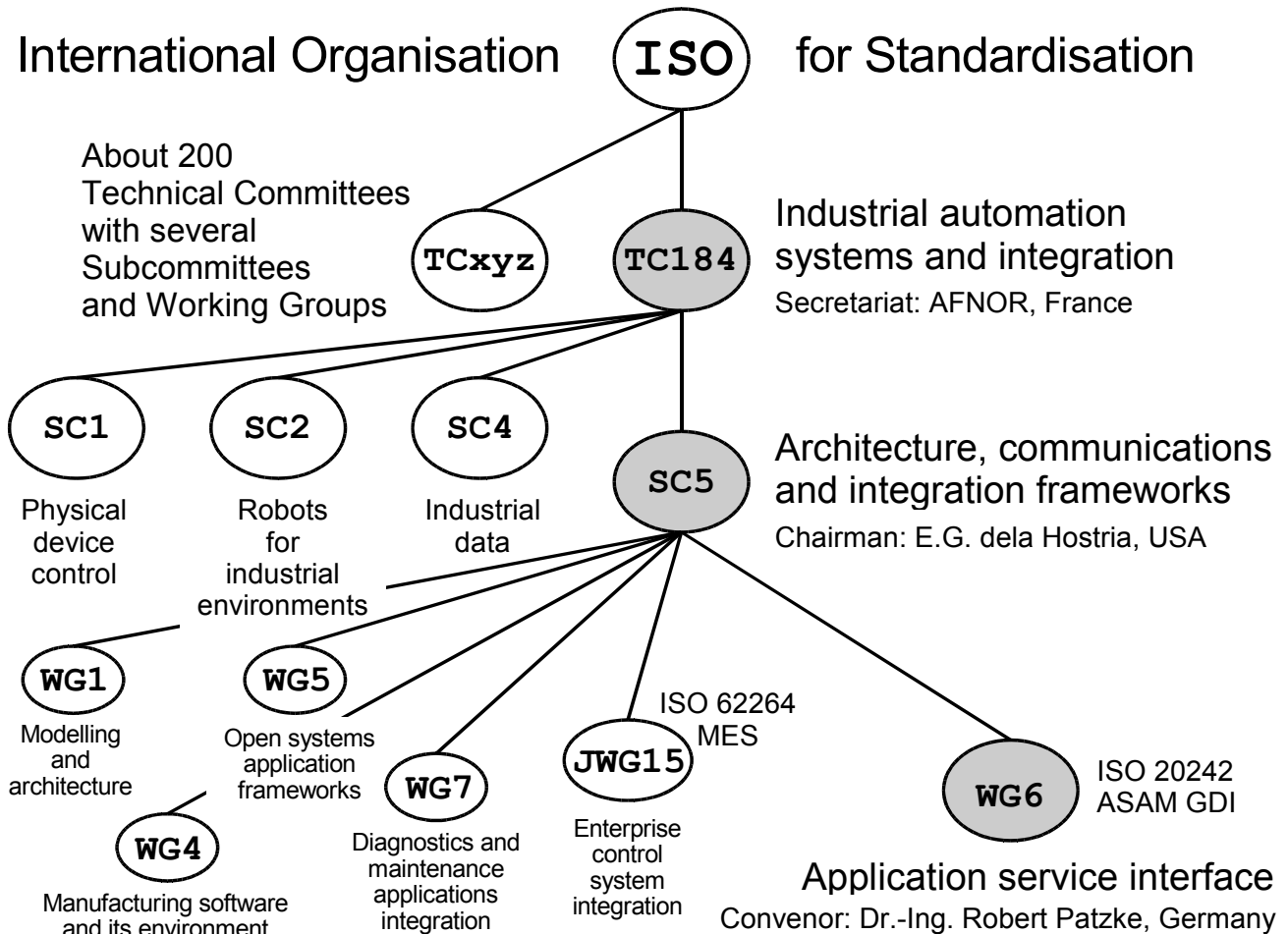


Bild 7: Organigramm zur Position des GDI in der ISO

Der generische Ansatz des GDI für die Geräteintegration hat Japan bewogen, zum Teil 4 (DCD und PID) einen Anhang beizusteuern, der die Anwendung der ISO 20242 für allgemeine Applikationen in der Produktion beschreibt. Unter dem Stichwort MICX (Manufacturing Informationen Collaboration with XML) wird hier eine Schnittstelle zwischen dem sog. Level 3 (MES, Manufacturing Execution System) und Level 2 (Automation Systems) für die diskrete Fertigung avisiert. Hier wird noch nach Wegen gesucht, um dort bereits etablierte Industriestandards, wie OPC, zu berücksichtigen.